

**Multi-Rotor Drone to Fly Autonomously along a River and
3D Map Modeling of an Environment around a River**

Akhmad Taufik

June 2016

Summary

In many developing countries, most people living in remote locations such as mountainous areas cannot take advantages of the electricity due to the limited access to the available system. Then, a small hydro electric power plant that utilizes the potential energy of water in a river has been developing in order to solve this problem. Nowadays, photo images and 3D coordinate data along a river are obtained by humans walking along the river by using a camera and a GPS. However, this method is very risky for the humans due to a swift water flow of the river and sharp stones on the bottom of the river. In addition, this method takes much time, cost and energy, to complete the work. In order to avoid the risk for the humans and to reduce the working time, cost and energy, a method to use a multi-rotor drone with a camera flying autonomously along a river is proposed to perform the work in this research. The first structure of the small hydro electric power plant comprises a simple weir located at the upstream of the river. Nowadays, photo images of an environment around the location of a weir are captured using a camera operated by a human. However, it is difficult for the human to capture the photo images due to that it is hard to reach the location of the weir. In addition, the photo images captured by the human are only two-dimensional images. In order to provide detailed views of the environment, a method to create a 3D textured map and a 3D map model of an environment around a river is also proposed in this research.

This research is composed of two works, namely a study on a multi-rotor drone to fly autonomously along a river and a study on 3D map modeling of an environment around a river. The purposes of the first study are to develop an algorithm of image processing to determine flying directions of a multi-rotor drone using photo images of river scenes and to make a multi-rotor drone perform autonomous flights along a river using a single-lens camera and the image processing. The purposes of the second study are to create a 3D textured map by using image processing of static photo images and video images of an environment around a river and a 3D map model of the environment by using a 3D printer.

In the first study, an algorithm of image processing was firstly developed. In the algorithm, a photo image of a river scene captured by a camera is converted to an HSV image. Then, the HSV image is converted to a binary image. Then, the lower part of the binary image is used to determine the river area. Then, the top side of the river area is used to determine the flying direction. After that, the developed algorithm was installed into a personal computer to perform the image processing. Furthermore, flying experiments where a multi-rotor drone (AR. Drone 2.0) performed manual flights along a river were carried out. The experiments were carried out under off-line and with real-time image processing. Finally, flying experiments where the multi-rotor drone performed autonomous flights along a river were carried out. The experimental result where the multi-rotor drone could autonomously fly along the river using the single-lens camera and the image processing for the distance of 83 [m] was obtained.

In the second study, a flying experiment where a multi-rotor drone captured images of an environment around a river was firstly carried out. In the experiment, a fisheye-lens camera mounted on the multi-rotor drone was used to capture overlapping static photo images and video images of the environment. Then, the image processing consisting of five consecutive steps, namely ① the loading and ② the aligning images, ③ the building the dense point clouds, ④ the polygonal meshes, and ⑤ the 3D textured maps were performed by using the Agisoft PhotoScan Professional (Ver. 1.1). After that, a thickened 3D map was built by using a selected area in the 3D textured map of the static photo images. Finally, a 3D map model of the environment around the river was created by using the thickened 3D map and the IRIS 3D printer.

Contents

Summary	ii
Contents	iii
1 Introduction	1
1.1 Background.....	1
1.2 Literature Reviews.....	2
1.3 Purposes of Research.....	6
1.4 Structure of Dissertation.....	6
2 Image Processing to Determine Flying Directions Using Photo Images of River Scenes	8
2.1 Introduction	8
2.2 Segmentation of River Scenes.....	9
2.2.1 Original Photo Images.....	9
2.2.2 Binary Images	10
2.2.3 River Area Determined Using the Lower Part of the Binary Image	12
2.2.4 Flying Direction Determined on the Upriver side of River Area.....	13
2.3 Algorithm of Image Processing.....	14
2.4 Experiments to Capture Photo Images Using a Few Cameras	17
2.4.1 Method	17
2.4.2 Results	21
2.4.2.1 Photo Images of River Scenes Captured Using Single-Lens Camera	21

2.4.2.2	Photo Images of River Scenes Captured Using Raspberry Pi Camera	24
2.4.2.3	Photo Images of River Scenes Captured Using Thermographic Camera	27
2.5	Experiments to Capture Videos Using a Single-Lens Camera	34
2.5.1	Method.....	34
2.5.2	Results	35
2.6	Conclusions	42
3	Multi Rotor Drone with a Single-Lens Camera to Fly Autonomously along a River	43
3.1	Introduction	43
3.2	Flying Experiments Using a Multi-Rotor Drone Flight-Controlled Manually	44
3.2.1	Method	44
3.2.2	Results	49
3.2.2.1	Off-Line Image Processing.....	49
3.2.2.2	Real-Time Image Processing.....	51
3.3	Algorithm for Autonomous Flight Along a River	53
3.3.1	Flying to the River	53
3.3.2	Flying Along the River.....	54
3.3.3	Flying to the Riverside	55
3.4	Flying Experiments Using a Multi-Rotor Drone Flight-Controlled Autonomously .	56
3.4.1	Method	56
3.4.2	Results	66
3.4.2.1	Flying Experiment on a Straight River.....	66
3.4.2.2	Flying Experiment on Curved Rivers	69
3.5	Conclusions	85

4	3D Map Modeling of Environment around a River	86
4.1	Introduction	86
4.2	Flying Experiment Using a Multi-Rotor Drone with a Fisheye-Lens Camera.....	87
4.2.1	Method	87
4.2.2	Result.....	90
4.2.2.1	Static Photo Images of an Environment around a River.....	90
4.2.2.2	Video Images of an Environment around a River	92
4.3	3D Mapping Using Image Processing of Static Photo Images and Video Images	94
4.3.1	Loading Images	94
4.3.2	Aligning Images	96
4.3.3	Building Dense Point Clouds	103
4.3.4	Building Polygonal Meshes	105
4.3.5	Building 3D Textured Maps.....	107
4.4	3D Map Modeling Using 3D Textured Map of Static Photo Images.....	109
4.4.1	Thickening 3D Textured Map	110
4.4.2	Preparing Material and Color 3D Printer	112
4.4.3	Printing 3D Map Model	114
4.4.4	Result.....	115
4.5	Conclusions	118
5	Summary of Conclusions	119
	References	120
	List of Papers and Awards.....	130
	Acknowledgments	132
	Curriculum Vitae	133

Chapter 1

Introduction

1.1 Background

In many developing countries, most people living in remote locations such as mountainous areas cannot take advantages of the electricity due to the limited access to the available system. Then, a small hydro electric power plant that utilizes the potential energy of water in a river has been developing in order to solve this problem. Nowadays, photo images and 3D coordinate data along a river are obtained by humans walking along the river by using a camera and a GPS. However, this method is very risky for the humans due to a swift water flow of the river and sharp stones on the bottom of the river. In addition, this method takes much time, cost and energy, to complete the work. In order to avoid the risk for the humans and to reduce the working time, cost and energy, a method to use a multi-rotor drone with a camera flying autonomously along a river is proposed to perform the work in this research.

The first structure of the small hydro electric power plant comprises a simple weir located at the upstream of the river. Nowadays, photo images of an environment around the location of a weir are captured using a camera operated by a human. However, it is difficult for the human to capture the photo images due to that it is hard to reach the location of the weir. In addition, the photo images captured by the human are only two-dimensional images. In order to provide detailed views of the environment, a method to create a 3D textured map and a 3D map model of an environment around a river is also proposed in this research.

Based on the background, this research is composed of two works. The first work is a study on a multi-rotor drone to fly autonomously along a river and the second one is a study on 3D map modeling of an environment around a river.

1.2 Literature Reviews

A review of published papers dealing with the studies on multi-rotor drone to perform an autonomous flight is presented in this section. The papers were reviewed based on the purposes of the first work of this research. The summary of the literature review of the first work is shown in Fig. 1.1.

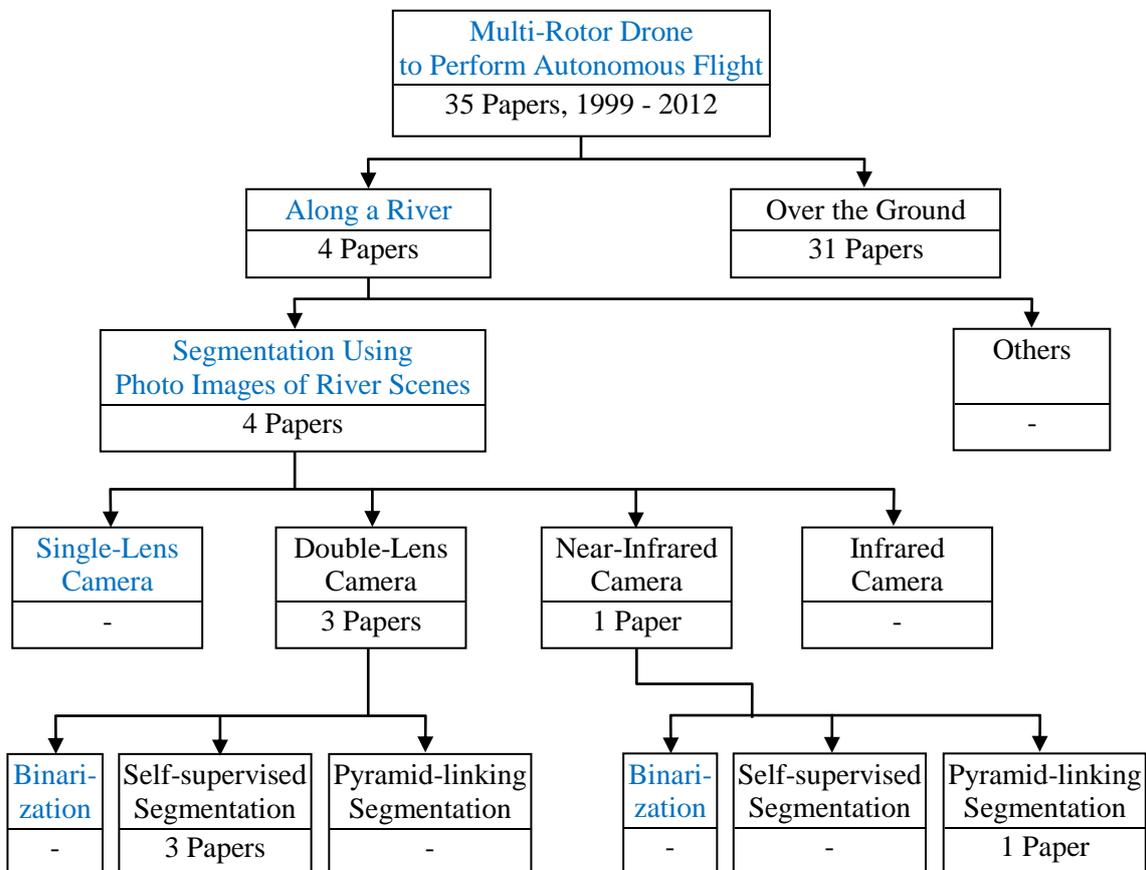


Fig. 1.1 Summary of the literature review of the study on multi-rotor drone to fly autonomously along a river

The 35 papers [1 – 35] were reported on multi-rotor drones to perform autonomous flights over environments reached by the authors. As far as the authors reached, only four papers [1 – 4] were reported on multi-rotor drones to perform autonomous flights along a river. In all of them reported, the segmentation of river scenes method was used to detect the river. Scherer et al [1], Chambers et al [2], and Achar et al [3] were reported on the use of a double-lens (stereo) camera mounted on a multi-rotor drone flying autonomously along a river at low altitude to capture photo images of the river scenes. In the three papers, the self-supervised segmentation method was used in the image processing to make their multi-rotor drones could fly autonomously along a river. Rathinam et al [4] was reported on the use of a near-infrared camera mounted on a multi-rotor drone flying along a river at high altitude to capture near-infrared images of river scenes. In his paper, the pyramid-linking image segmentation method was used in the image processing to make his multi-rotor drone could fly autonomously along a river. Then, the 31 papers [5 – 35] were reported on multi-rotor drones to perform autonomous flights over ground environments.

As far as the authors reached, there is no report in which a single-lens camera mounted on a multi-rotor drone was used to capture photo images of river scenes as presented in this research. In addition, there is no report in which the binarization method was used in the image processing of photo images to make a multi-rotor drone could fly autonomously along a river as presented in this research.

Then, a review of published papers dealing with the studies on 3D map modeling of an environment is also presented in this section. The papers were reviewed based on the purposes of the second work of this research. The summary of the literature review of the second work is shown in Fig. 1.2.

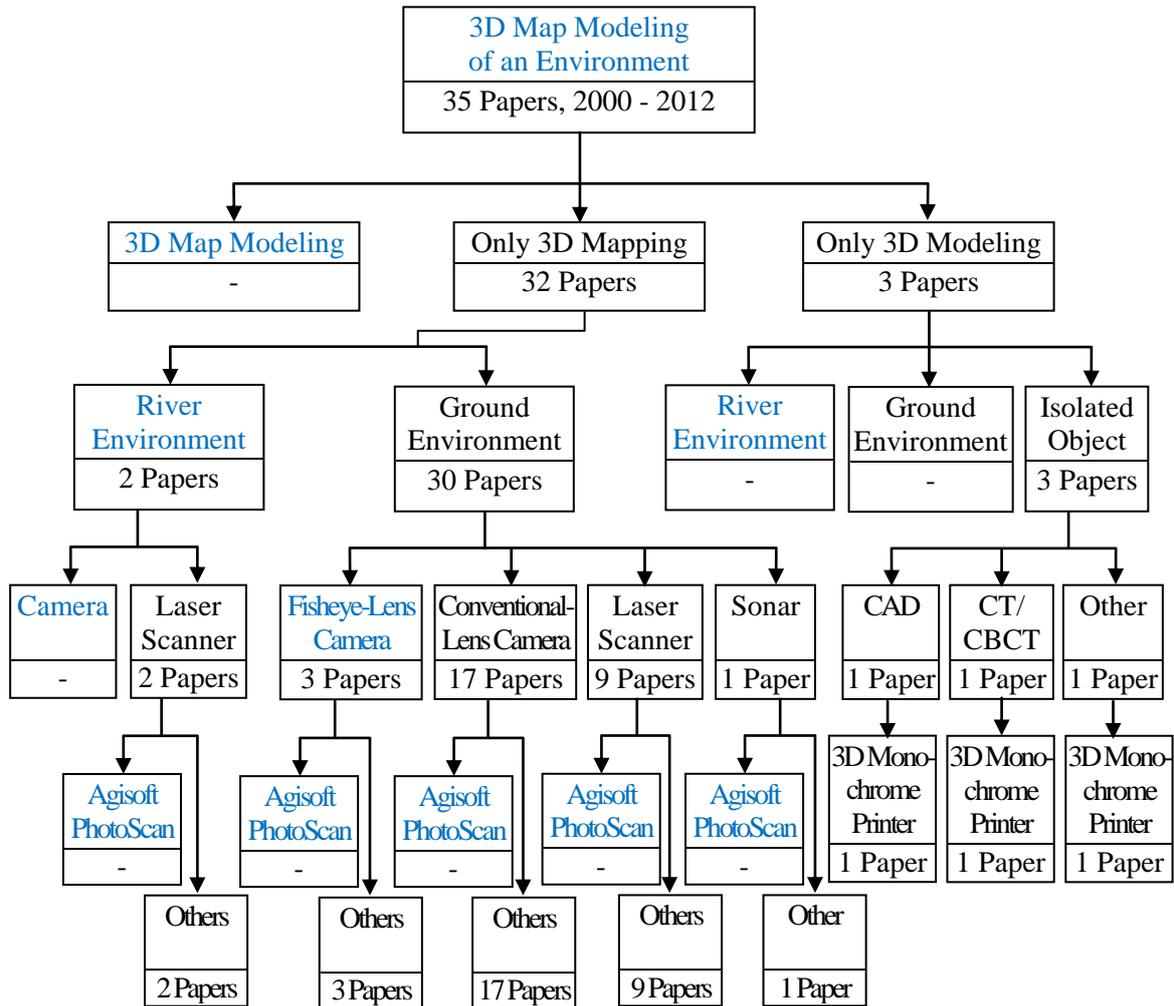


Fig. 1.2 Summary of the literature review of the study on 3D map modeling of an environment around a river

The 35 papers [36 – 70] were reported on 3D mapping and 3D modeling of an environment reached by the authors. As far as the authors reached, there is no paper in which a 3D map modeling of an environment around a river was reported as presented in this research.

The 32 papers [36 – 67] were reported on only 3D mapping of environments. Then, only two of them [36] and [37] were reported on 3D mapping of river environments. In the two papers, a 3D laser scanner was used to obtain the 3D point data of vegetation along the river. Then, the 30 papers [38 – 67] were reported on 3D mapping of ground environments. To obtain the data of the ground environments, three papers [38 – 40] were reported on the use of fisheye-lens cameras, 17 papers [41 – 57] were reported on the use of conventional-lens cameras, nine papers [58 – 66] were reported on the use of laser scanners, and one paper [67] was reported on the use of a sonar. Then, all of them use a software to create the 3D maps of the ground environment. As far as the authors reached, there is no report in which the Agisoft PhotoScan was used to create a 3D textured map of an environment around a river as presented in this research.

Then, the three papers [68 – 70] were reported on only 3D modeling of isolated objects for applications in medical field. Olmo et al [68] was reported on the use of CAD Software to create the 3D data of a bone. Metzger et al [69] was reported on the use of CT/CBCT to obtain the 3D data of a skull. Lam et al [70] was reported on the use of another method to obtain the 3D data of scaffolds. In the three papers, 3D monochrome printers were used to produce the 3D models of isolated objects. As far as the authors reached, there is no report in which a 3D color printer was used to create a 3D map model of an environment around a river as presented in this research.

1.3 Purposes of Research

This research is composed of two works, namely a study on a multi-rotor drone to fly autonomously along a river and a study on 3D map modeling of an environment around a river. The purposes of the study on a multi-rotor drone to fly autonomously along a river are listed as follows.

1. To develop an algorithm of image processing to determine flying directions of a multi-rotor drone using photo images of river scenes.
2. To make a multi-rotor drone perform autonomous flights along a river using a single-lens camera and the image processing.

Then, the purposes of the study on 3D map modeling of an environment around a river are listed as follows.

1. To create a 3D textured map by using image processing of static photo images and video images of an environment around a river.
2. To create a 3D map model of the environment by using a 3D color printer.

1.4 Structure of Dissertation

The dissertation is structured in the following manner. Chapter 1 presents the background of the research, literature reviews, purposes of the research and structure of the dissertation. The literature review is addressed to the study on a multi-rotor drone performing an autonomous flight by using a camera and image processing and study on 3D map modeling of an environment.

Chapter 2 presents a study on image processing to determine flying directions of a multi-rotor drone using photo images of river scenes. The method of segmentation of

river scenes is firstly presented in this chapter. Then, an algorithm of image processing to perform the segmentation of river scenes is described. Then, the experimental method and results of capturing photo images of river scenes by using a few cameras are presented. After that, the experimental method and results of capturing videos of river scenes by using a single-lens camera are presented at the end of this chapter.

Chapter 3 presents an experimental study on autonomous flights performed by a multi-rotor drone using a single-lens camera and the image processing. The experimental method and results where the multi-rotor drone flying along a river flight-controlled manually are firstly presented. Then, an algorithm to perform an autonomous flight along a river is described. After that, the experimental method and results where the multi-rotor drone flying autonomously along a river by using the single-lens camera and the image processing are presented at the end of this chapter.

Chapter 4 presents a study on 3D map modeling of an environment around a river. The experimental method and result in which a multi-rotor drone with a fisheye-lens camera captured static photo images and video images of a targeted environment around a river are firstly presented. Then, the 3D mapping to create 3D textured maps of the captured static photo images and video images is presented. After that, the 3D map modeling to create a 3D map model of the environment around the river by using a 3D printer is presented at the end of this chapter.

Finally, chapter 5 presents summary of conclusions of the research. In addition, lists of references of reviewed literature and publications as the core of dissertation are given at the end of this dissertation.

Chapter 2

Image Processing to Determine Flying Direction

Using Photo Images of River Scenes

2.1 Introduction

Recently, image processing to segment a photo image into multiple parts are used in many fields such as, robotic, photographic, medical diagnosis, and so on. The development of image processing is supported by the advance of technologies to acquire and process digital data such as, digital cameras and image processing softwares. In this chapter, a study on image processing to determine flying direction using photo images of river scenes captured by a few cameras is presented.

The purposes of the study presented in this chapter are to develop an algorithm of image processing to determine flying direction using photo images of river scenes and to perform image processing of photo images of river scenes captured by a few cameras.

In this study, an algorithm of image processing to determine flying directions of a multi-rotor drone was firstly developed. Then, a program computer was written by using an open source programming language based on the developed algorithm. After that, some experiments to capture photo images of river scenes were carried out by using three different cameras. Finally, image processing of the photo images of river scenes captured by the three cameras were performed.

2.2 Segmentation of River Scenes

There are three main steps in order to perform the segmentation of river scenes. In the first step, an original photo image of a river scene was converted to an HSV image. Then, the HSV image was converted to a binary image by adjusting the threshold values of HSV parameters of the HSV image. In the second step, the river area was determined using the lower part of the binary image. In the third step, the flying direction was determined by finding the center point on top side of the river area. Figure 2.1 shows the three main steps to perform the segmentation of river scenes.

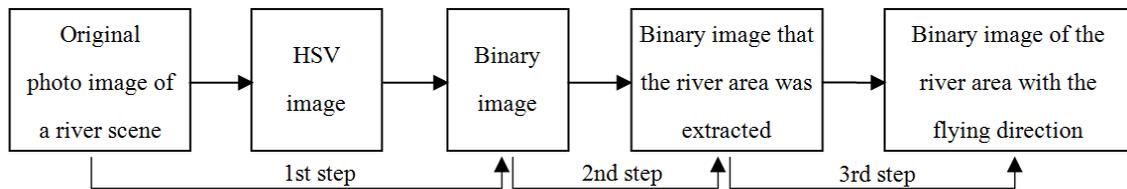


Fig. 2.1 Three main steps to perform the segmentation of river scenes

2.2.1 Original Photo Images

Figure 2.2 shows an original photo image of a river scene captured by a camera which consists of the sky, some regions such as foliage, foliage reflections and sky reflections. The original photo image is an RGB image that is a digital image having three channels in each pixel, namely red, green, and blue. Each channel consists of 8-bit data of colors. The maximum number of colors that can be displayed in the photo image is 256. The original photo image is formed by the reflection of visible light on surface of the river and other objects such as, foliage, mountain and sky. In order to segment the river area from the river scene, the original photo image was converted to an HSV image, then the HSV image was converted to a binary image representing the river area and other areas.



Fig. 2.2 Original photo image of a river scene

2.2.2 Binary Images

Figure 2.3 shows an HSV image converted from the original photo image. The original photo image of a river scene captured by the camera was firstly converted to an HSV image that is a digital image having three channels in each pixel, namely “hue” (colors), “saturation” (grayness), and “value” (brightness). Each channel contains 8-bit data of a color, grayness, or brightness.

In order to detect the river area, the HSV image was converted to a binary image by adjusting the minimum and maximum threshold values of the three parameters of the HSV image, namely “hue”, “saturation”, and “value”. The value of each parameter was stated in a numeral range from 0 to 256. The minimum and maximum threshold values of the HSV parameters used to convert the HSV image to the binary image were 0 and 256 for “hue” parameter, 0 and 47 for “saturation” parameter, and 97 and 241 for “value” parameter as shown in Fig. 2.4. The minimum and maximum threshold values of “hue” parameter was not adjusted due to that the water of the river almost has no color. The most important parameter in this study is the of “value” parameter.



Fig. 2.3 HSV image converted from the original photo image

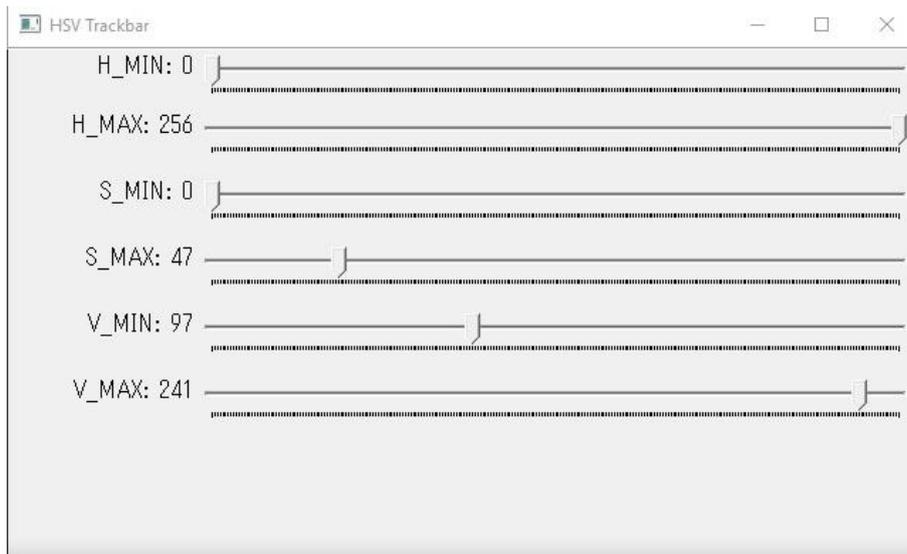


Fig. 2.4 Minimum and maximum threshold values of HSV parameters when converting the HSV image to a binary image

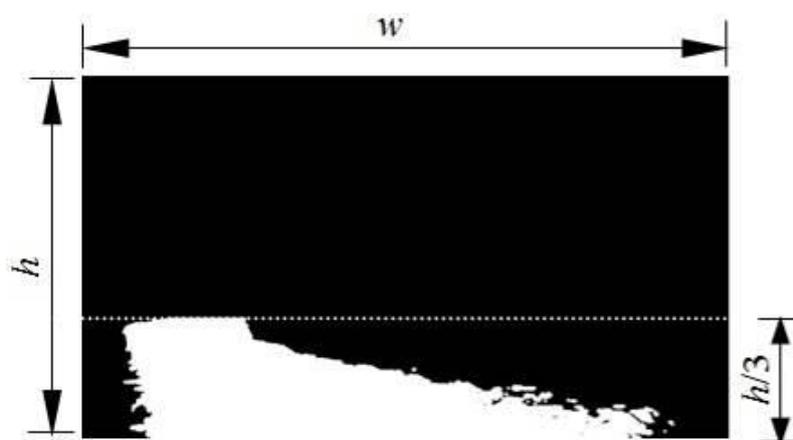
The river area was shown in the lower side of the binary image. However, several areas that are outside the river area may appear in the upper side of the binary image as shown in Fig. 2.5. To overcome this problem, the superfluous areas were removed in order to find a clear river area.



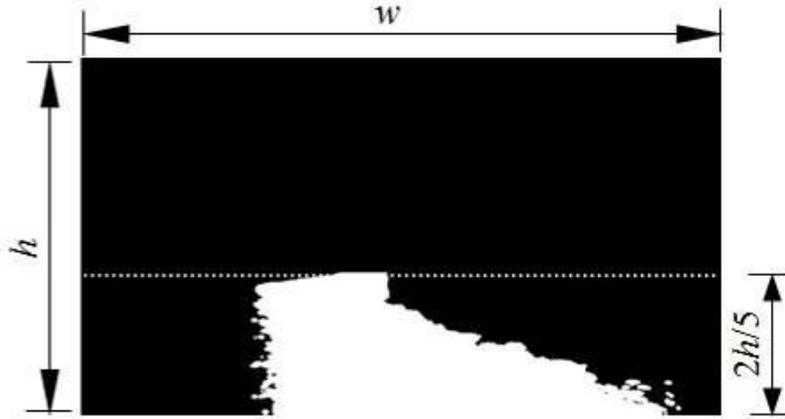
Fig. 2.5 Binary image converted from the HSV image

2.2.3 River Area Determined Using the Lower Part of the Binary Image

In the binary image, the river area was shown in the bottom half side. The height of the river area was depended on the position of the camera when capturing the original photo image. Then, the river area was determined as the bottom part of one-third and two-fifths of the whole binary image as shown in Fig. 2.6. The white area was considered as the river area and the black area was considered as other areas.



(1) River area determined at the bottom part of one-third of the whole binary image



(2) River area determined at the bottom part of two-fifths of the whole binary image

w : width of the image [pixel]
 h : height of the image [pixel]

Fig. 2.6 River area determined using the lower part of the whole binary image

2.2.4 Flying Direction Determined on the Upriver side of River Area

The flying direction was determined by finding the center point of top side of the river area, namely upriver side. Firstly, the center point of the bottom side of the whole binary image was determined. The coordinate of the center point of the bottom side was denoted by $P_1(x_1, y_1)$. Then, the left and right end of the upriver side denoted by P_a and P_b were determined as shown in Fig. 2.7. Then, the center point of top side denoted by P_2 was determined by finding the center point between P_a and P_b . The coordinate of center point of top side of the river area was denoted by $P_2(x_2, y_2)$. Furthermore, the flying direction was then determined using a line drawn from the center point of bottom side of the whole binary image to the center point of top side of the river area as shown in Fig. 2.8. The O-xy is the coordinate frame of the binary image. $P_1(x_1, y_1)$ represents the current position of the camera and $P_2(x_2, y_2)$ represents the end point of the flying direction.

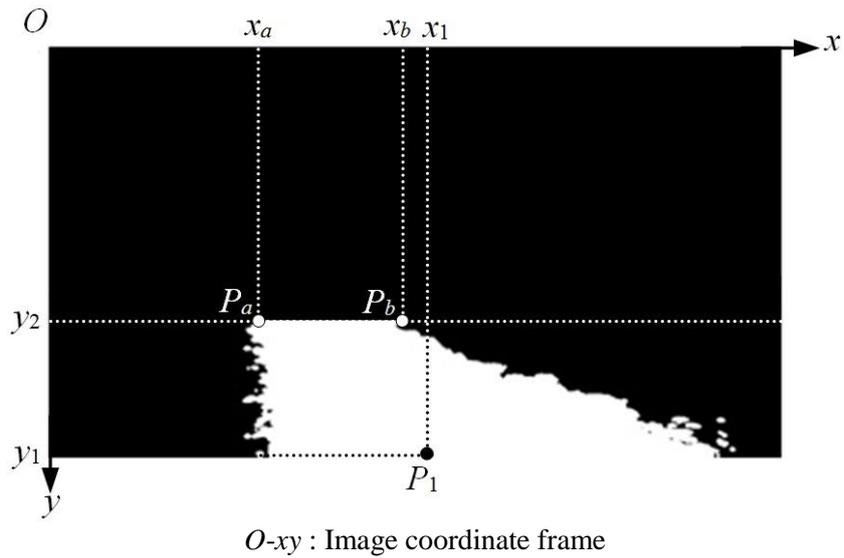


Fig. 2.7 The left and right end of top side of the river area found to determine the center point of the river area

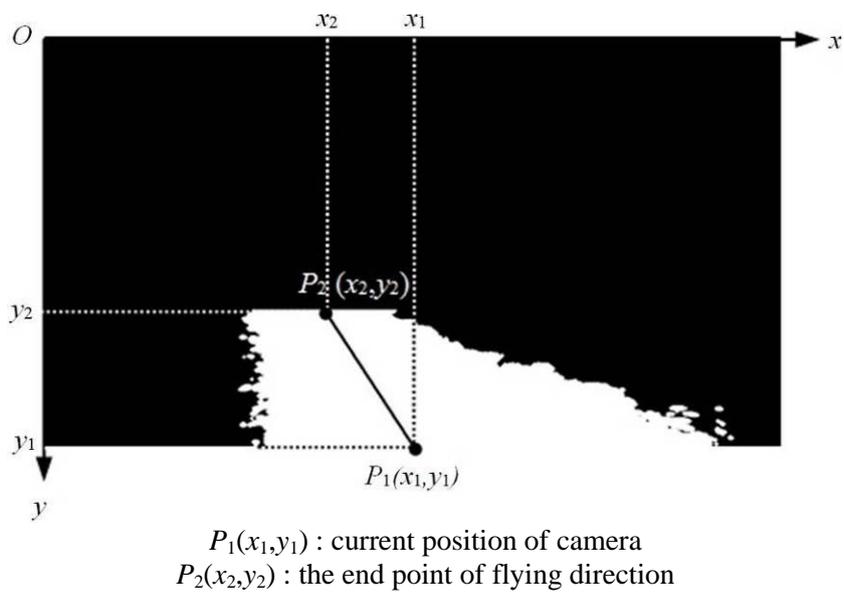


Fig. 2.8 Flying direction visualized by a line drawn from P_1 to P_2 in the river area

2.3 Algorithm of Image Processing

Figure 2.9 shows the algorithm of image processing to perform the segmentation of river scenes. The algorithm was used to segment the river area from the river scene and to determine the flying direction on the river area. Firstly, the initial values of HSV

parameters were defined. Then, the data of an original photo image of a river scene was read. After that, a while loop was created in order to convert the original image to binary image with flying direction. The original image was firstly converted to an HSV image. Then, the HSV image was converted to a binary image by selecting threshold values of the HSV parameters of the HSV image. After that, the superfluous areas in the upper side of the binary image were removed. The river area was then determined in the lower side of the binary image. Furthermore, the flying direction was determined on the river area. Finally, the binary image with flying direction on the river area was shown as the result of the image processing.

The algorithm of image processing to perform segmentation of river scene was then written as a computer program by using Microsoft Visual C++ 2010 Express integrated with OpenCV (Ver. 2.4.9) that is an open source library of digital image processing. The computer program was used to segment the river area from the river scene and to determine the flying direction on the river area by using the image processing of photo images of river scenes captured by a camera.

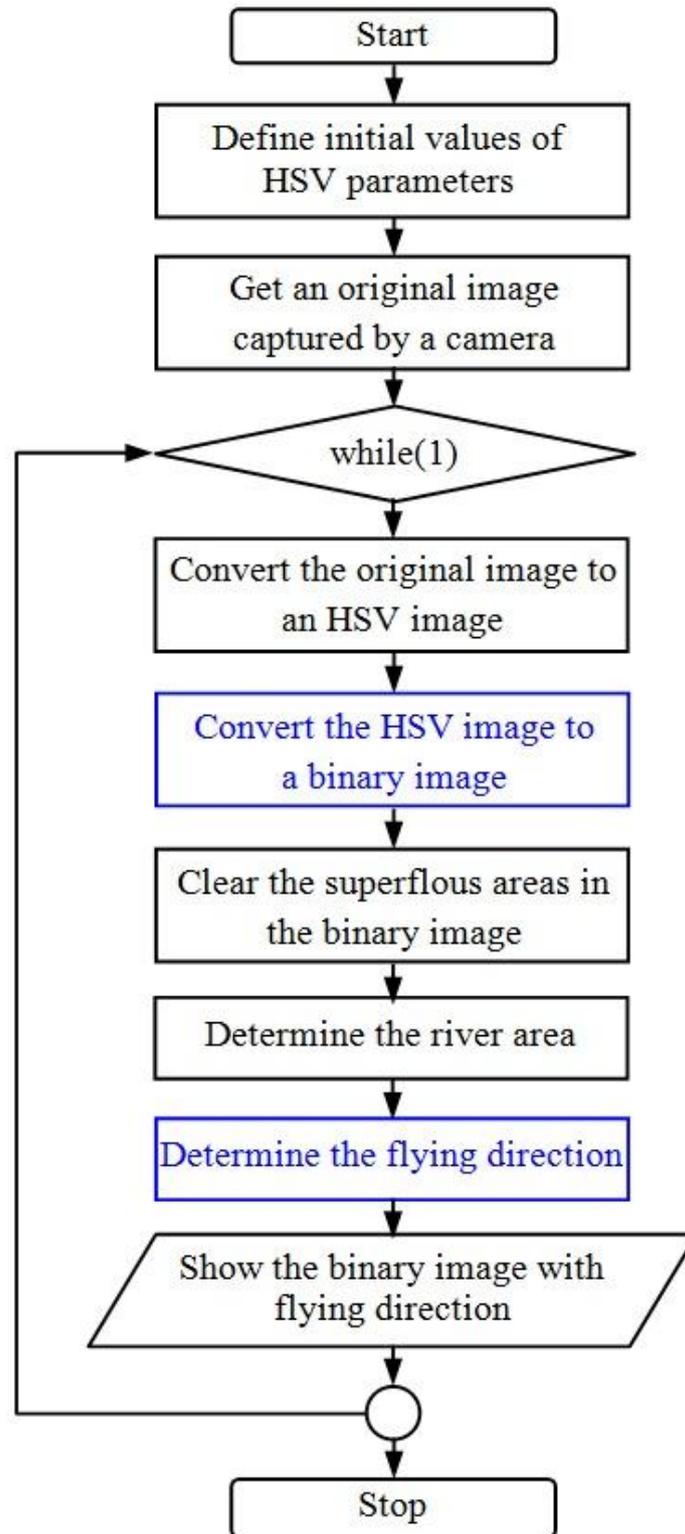


Fig. 2.9 Algorithm of image processing to perform segmentation of river scenes

2.4 Experiments to Capture Photo Images Using a Few Cameras

2.4.1 Method

The experiments to capture photo images of river scenes were carried out by using three different cameras, namely a single-lens camera, Raspberry Pi Camera, and thermographic camera as shown in Fig. 2.10. The photo images of river scenes captured by the cameras were then processed by using the developed algorithm presented in Section 2.3. After that, the results of the image processing were compared in order to determine the best camera for using in this study.



Fig. 2.10 Three cameras used in experiments to capture photo images of river scenes

In the first experiment, a single-lens camera having a wide angle of $92 [^\circ]$ was used to capture photo images of river scenes. Firstly, the single-lens camera was connected to the mainboard of AR. Drone 2.0. The mainboard and a tablet computer installing a software to operate the single-lens camera were connected each other through a wireless connection (Wi-Fi) as shown in Fig. 2.11. Then, capturing commands were sent from the tablet computer to the mainboard through the Wi-Fi. After that, photo images of river scenes were captured by the single-lens camera and sent to the tablet computer through the Wi-Fi. The photo images of river scenes were saved in the memory of the tablet computer in JPEG format files.

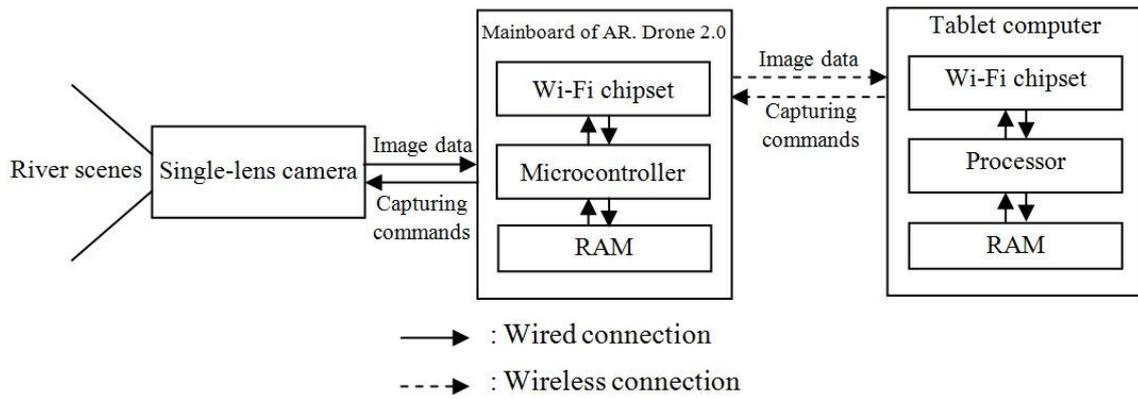


Fig. 2.11 Signal flow of system used in the experiment in which a single-lens camera captures photo images of river scenes



Fig. 2.12 Single-lens camera connected to the main board of AR.Drone 2.0 attached to a stick held by a person in the center of a river in an experiment to capture photo images of river scenes

Figure 2.12 shows the single-lens camera connected to the mainboard of AR. Drone 2.0 attached to a stick held by a person standing in the center of a river in an experiment to capture photo images of river scenes. The position of the single-lens camera was 2 [m] above the surface of the river. The tablet computer used to operate the single-lens camera and to save the photo images captured by the single-lens camera was holded by a person standing at the riverside.

In the second experiment, a Raspberry Pi Camera that is a near-infrared camera was used to capture photo images of river scenes. The Raspberry Pi Camera was connected to the Raspberry Pi mainboard by using a Camera Serial Interface (CSI-2) connector. Then, a memory card installing a software to operate the Raspberry Pi Camera was installed in the Raspberry Pi mainboard. The software runs on Raspbian as the operating system. The Raspberry Pi mainboard was then connected to a personal computer by using a network cable as shown in Fig. 2.13.

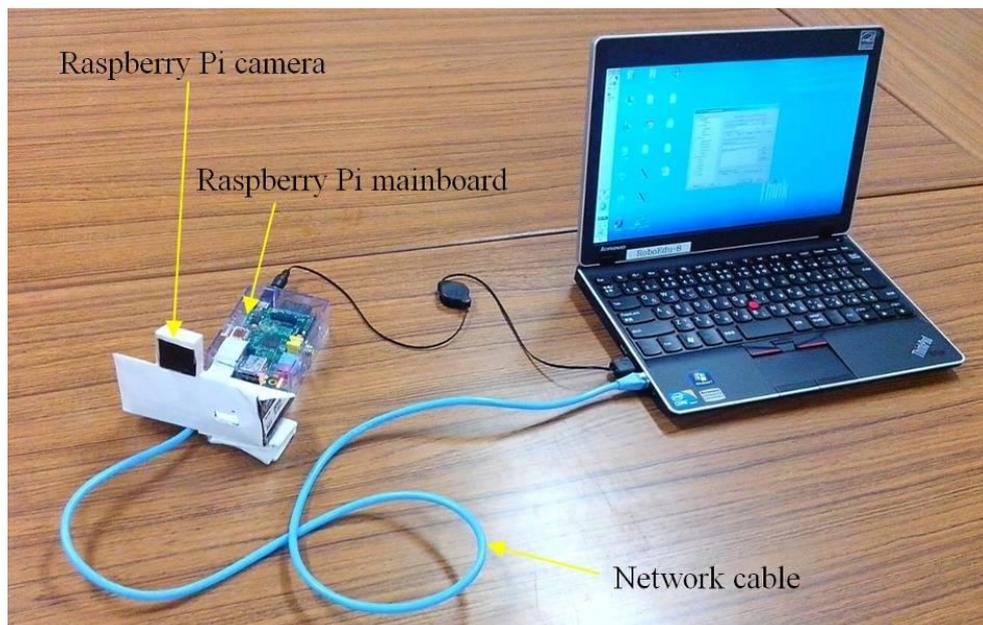


Fig. 2.13 Raspberry Pi Camera attached to Raspberry Pi mainboard connected to a personal computer used in experiment to capture photo images of river scenes

In the networking system, the Raspberry Pi mainboard was used as the client and the personal computer was used as the server. Two softwares were used to establish the networking system: Xming X Server 6.9.0.31 and Putty. The specification of processor of the personal computer is Intel Core i3 U380 @ 1.33 GHz and the capacity of RAM is 2.00 GB.

In the third experiment, the Thermo Shot Camera F30W that is a thermographic or infrared camera was used to capture photo images of river scenes. The thermographic camera was equipped with two lenses, namely a standard-size-lens and a small-lens as shown in Fig. 2.14. Infrared light were used by the standard-size-lens and visible light were used by the small-lens to form a photo image. The range of temperature where the standard-size-lens captures an infrared image is from -20 to 350 [°C] with accuracy of 0.2 [°C]. Since the temperature of environment in the morning and afternoon were different, the experiments to capture RGB and infrared images of river scenes by using the thermographic camera were carried out in those two different time. After finishing the three experiments, the image processing of the photo images captured by the three cameras were performed to determine the best camera for using in further experiments.

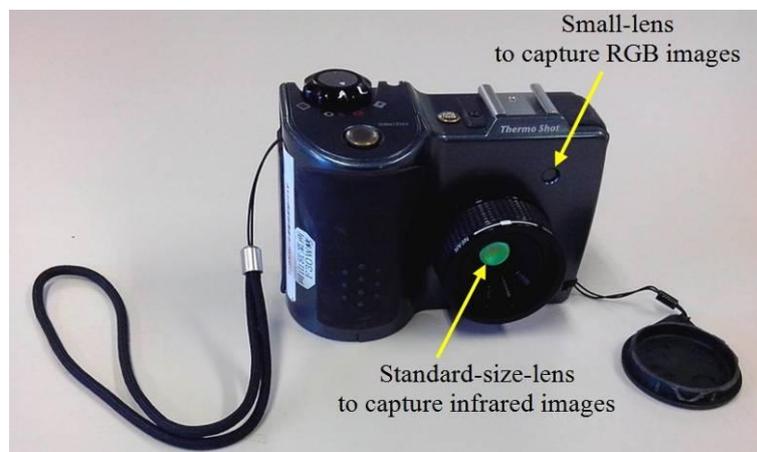


Fig. 2.14 Thermographic camera with two lenses used in the experiment to capture photo images of river scenes

Figure 2.15 shows the location of experiments to capture photo images of river scenes by using the three different cameras. The experiments were carried out on Ishite River in Matsuyama city, Ehime Prefecture, Japan.



Fig. 2.15 Location of experiments to capture photo images of river scenes by using three different cameras

2.4.2 Results

2.4.2.1 Photo Images of River Scenes Captured Using Single-Lens Camera

Figure 2.16 shows the examples of original photo images of river scenes captured by using the single-lens camera. Visible light were used by the single-lens camera to form a photo image. Therefore, the photo images of river scenes captured by the single-lens camera were RGB images. The dimensions of photo images captured by the single-lens camera were 1280×720 [pixels]. It means that the size of each photo image captured by the single-lens camera was 0.92 [MB]. The segmentation of river scenes was then performed on the original photo images by using the developed algorithm presented in Section 2.3. Figure 2.17 shows the binary images converted from the original photo ones of river scenes captured by the single-lens camera. It can be seen that the river areas were segmented well from the river scenes in the binary images.



(1) Original photo image 1



(2) Original photo image 2

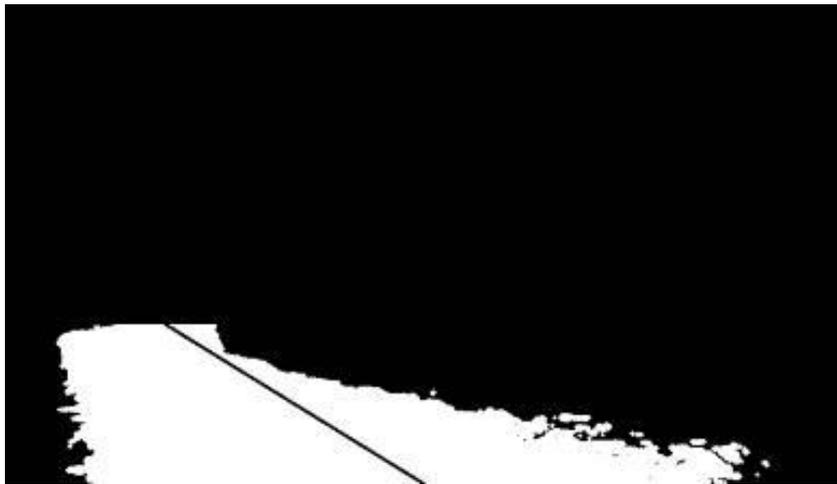


(3) Original photo image 3

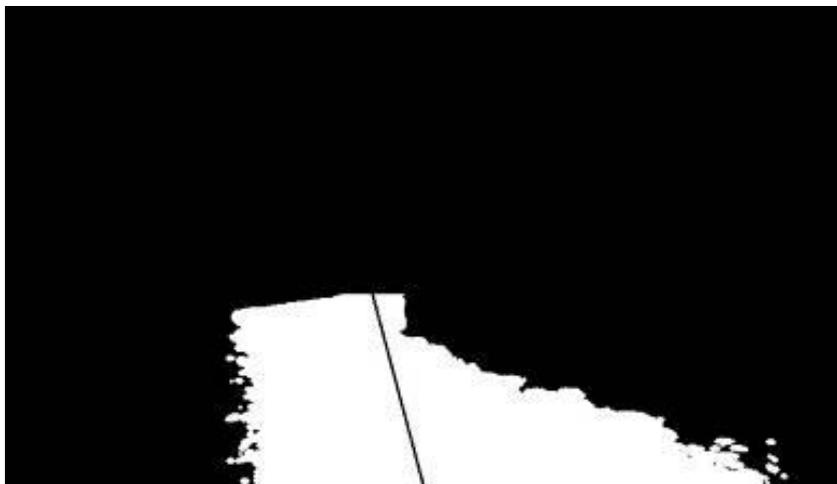
Fig. 2.16 Examples of original photo images of river scenes captured by the single-lens camera



(1) Binary image 1



(2) Binary image 2



(3) Binary image 3

Fig. 2.17 Binary images converted from the original photo ones of river scenes captured by the single-lens camera

2.4.2.2 Photo Images of River Scenes Captured Using Raspberry Pi Camera

Figure 2.18 shows the examples of original photo images of river scenes captured by using the Raspberry Pi Camera. Red and infrared light were used by the Raspberry Pi Camera to form a photo image. Therefore, the photo images of river scenes captured by the Raspberry Pi Camera were near-infrared images. About 100 images of river scenes were captured by the Raspberry Pi Camera in the experiments. The segmentation of river scenes was then performed on the photo images by using the developed algorithm presented in Section 2.3.

Figure 2.19 shows the binary images converted from the original photo ones of river scenes captured by using Raspberry Pi Camera. The dimensions of photo images captured by the Raspberry Pi Camera were 2592×1944 [pixels]. It means that the size of each photo image captured by the Raspberry Pi Camera was 5.0 [MB]. The size of the photo image captured by the Raspberry Pi Camera were five times of that captured by the single-lens camera. Therefore, the processing time of a photo image captured by the Raspberry Pi Camera was also five times of that of a photo image captured by the single-lens camera. The condition made the personal computer used to perform the image processing worked very hard and needed much time to accomplish the image processing. In addition, many superfluous areas were appear in the binary images converted from the original photo ones due to that the photo images captured by the Raspberry Pi Camera have less colors and brightness. Therefore, good binary images were not obtained in the image processing of photo images of river scenes captured by the Raspberry Pi Camera.



(1) Original photo image 1



(2) Original photo image 2



(3) Original photo image 3

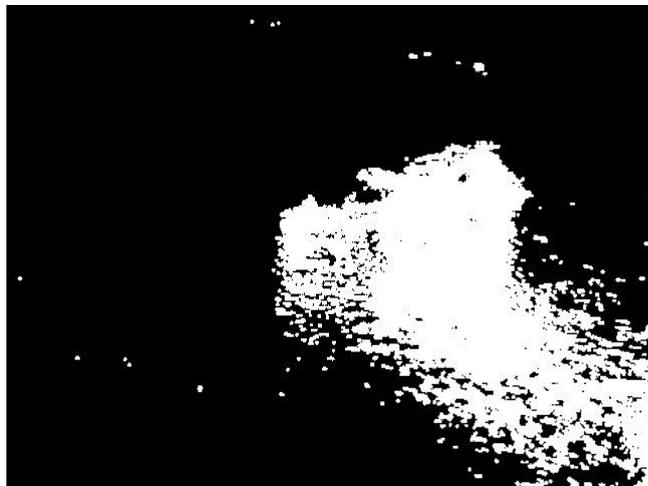
Fig. 2.18 Examples of original photo images of river scenes captured using Raspberry Pi Camera



(1) Binary image 1



(2) Binary image 2



(3) Binary image 3

Fig. 2.19 Binary images converted from the original photo ones of river scenes captured using Raspberry Pi Camera

2.4.2.3 Photo Images of River Scenes Captured Using Thermographic Camera

Figures 2.20 and 2.21 show the examples of RGB and infrared images of river scenes captured by using the thermographic camera in an experiment carried out in the morning. Figures 2.23 and 2.24 show the examples of RGB and infrared images of river scenes captured by the thermographic camera in an experiment carried out in the afternoon. Visible and infrared light were used to form the photo images captured by the thermographic camera (Thermo Shot F30W). Therefore, the photo images captured by the thermographic camera were RGB and infrared images. The 162 images of river scenes were captured by the thermographic camera in the experiments carried out in the morning and afternoon. The dimensions of RGB and infrared images captured by the thermographic camera were 1280×1024 [pixels] and 320×240 [pixels], respectively. The infrared images were then converted to binary images by using the developed algorithm.

Figures 2.22 and 2.25 show the binary images converted from the infrared ones captured by the standard-size-lens of thermographic camera in the experiment carried out in the morning and afternoon, respectively. It can be seen that the binary images converted from the infrared ones captured by the thermographic camera in the afternoon were better than those converted from the infrared images captured in the morning. The temperature of surface of the river and its environment were different in the afternoon but almost same in the morning. Therefore, the river areas were segmented well from the river scenes using infrared images captured by the thermography camera in the afternoon. However, the result shows that the thermographic camera is only good for using in the afternoon.



(1) RGB image 1

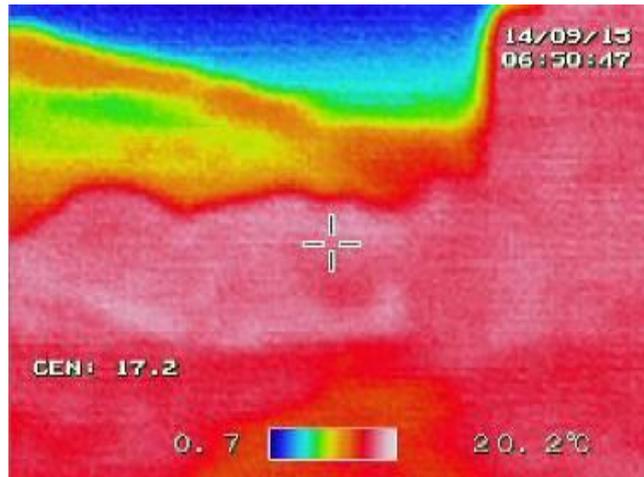


(2) RGB image 2

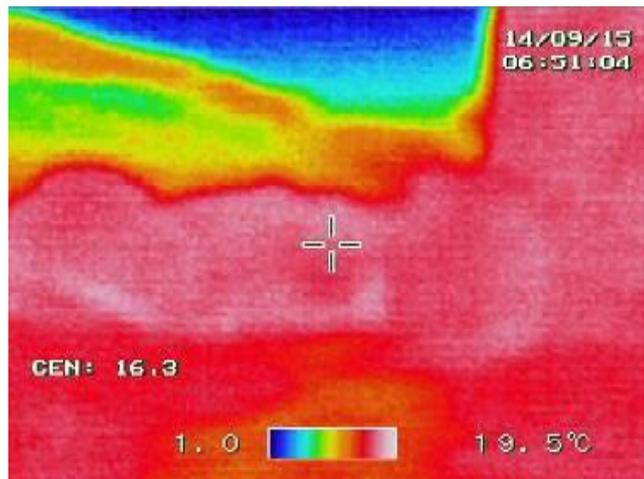


(3) RGB image 3

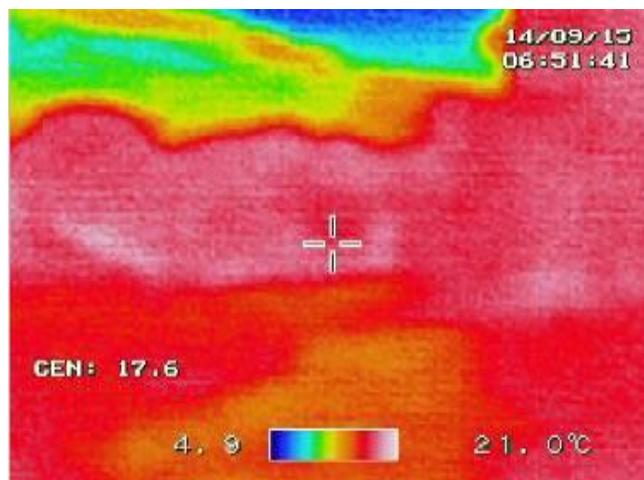
Fig. 2.20 Examples of RGB images of river scenes captured by the small-lens of thermographic camera in the experiment carried out in the morning



(1) Infrared image 1



(2) Infrared image 2



(3) Infrared image 3

Fig. 2.21 Examples of infrared images of river scenes captured by the standard-size-lens of thermographic camera in the experiment carried out in the morning



(1) Binary image 1



(2) Binary image 2



(3) Binary image 3

Fig. 2.22 Binary images converted from the infrared ones captured by the standard-size-lens of thermographic camera in the experiment carried out in the morning



(1) RGB image 1

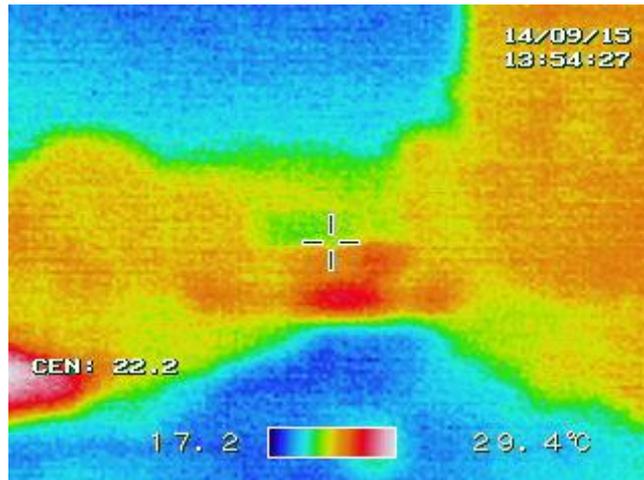


(2) RGB image 2

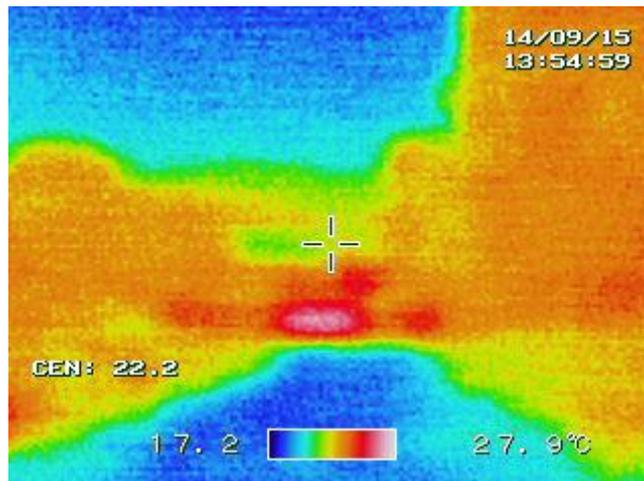


(3) RGB image 3

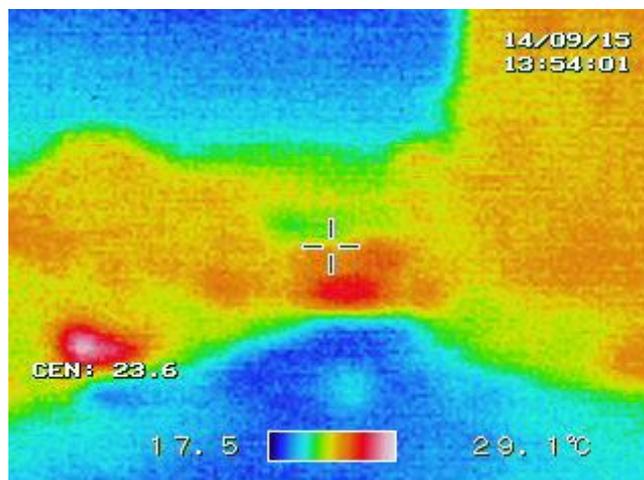
Fig. 2.23 Examples of RGB images of river scenes captured by the small-lens of thermographic camera in the experiment carried out in the afternoon



(1) Infrared image 1



(2) Infrared image 2



(3) Infrared image 3

Fig. 2.24 Examples of infrared images of river scenes captured by the standard-size-lens of thermographic camera in the experiment carried out in the afternoon



(1) Binary image 1



(2) Binary image 2



(3) Binary image 3

Fig. 2.25 Binary images converted from the infrared ones captured by the standard-size-lens of thermographic camera in the experiment carried out in the afternoon

After finishing the segmentation of river scenes using the photo images captured by the three cameras, the results were compared to determine the best camera for using in further experiments. It can be seen that the binary images converted from the photo images captured by the single-lens camera were better than those captured by the Raspberry Pi Camera and the thermographic camera. Therefore, the single-lens camera was selected for using in flying experiments performed by a multi-rotor drone.

2.5 Experiments to Capture Videos Using a Single-Lens Camera

After finishing the image processing of photo images of river scenes, experiments to capture videos of river scenes were carried out by using the single-lens camera. After that, image processing to determine the flying directions using the video images of river scenes were performed.

2.5.1 Method

The method used in the experiments to capture video images of river scenes are same as that used in the experiments to capture photo images of river scenes by using the single-lens camera. In the experiments, the single-lens camera connected to the mainboard of AR. Drone 2.0 was attached to a stick held by a person walking in the center of a river. Then, videos of river scenes were captured by using the single lens camera and then sent to the tablet computer through a wireless connection (Wi-Fi). Then, videos of river scenes were saved in the memory of the tablet computer in 3GP format files.

Experiments to capture videos of river scenes by using the single-lens camera mounted on the multi-rotor drone were carried out in three different weather, namely

cloudy, sunny, and very sunny. After finishing the experiments, the image processing of video images captured by the single-lens camera in the three different weather were performed by using a personal computer.

2.5.2 Results

The rate of capturing of video images of river scenes by the single-lens camera was 30 [fps]. It means that thirty video images were captured by the single-lens camera in every second. Then, the rate of image processing on video images of river scenes captured by the single-lens camera was 1 [fps]. It means that one of thirty video images captured by the single-lens camera was processed in every second by the personal computer.

Examples of video images captured by the single-lens camera in an experiment carried out in cloudy weather and binary images converted from the video ones are shown in Figs. 2.26 and 2.27, respectively. Then, examples of video images captured by the single-lens camera in an experiment carried out in sunny weather and binary images converted from the video ones are shown in Figs. 2.28 and 2.29, respectively. Then, examples of video images captured by the single-lens camera in an experiment carried out in very sunny weather and binary images converted from the video ones are shown in Figs. 2.30 and 2.31, respectively. In the binary images converted from the video ones captured by the single-lens camera in very sunny weather, many superfluous areas were appeared due to the existence of many foliage reflections on the video images. Therefore, the flying directions could not be determined well when a lot of superfluous areas were appeared on the binary image as shown in Fig. 2.31 (c).



(a) $t = 10.0$ [s]

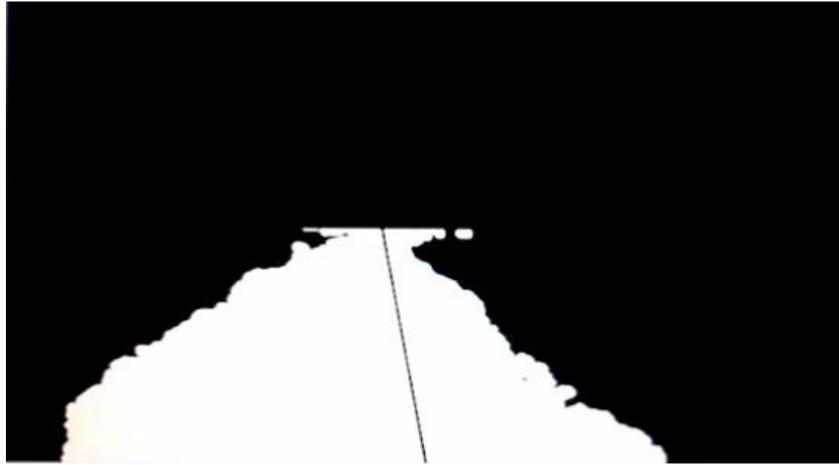


(b) $t = 20.0$ [s]

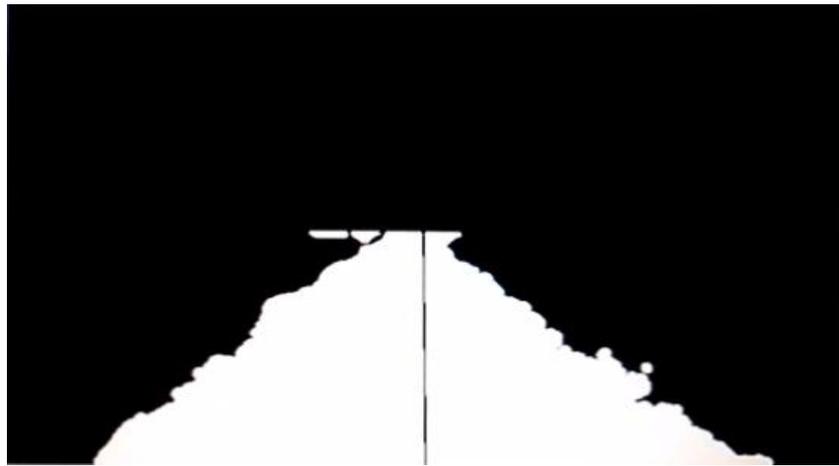


(c) $t = 30.0$ [s]

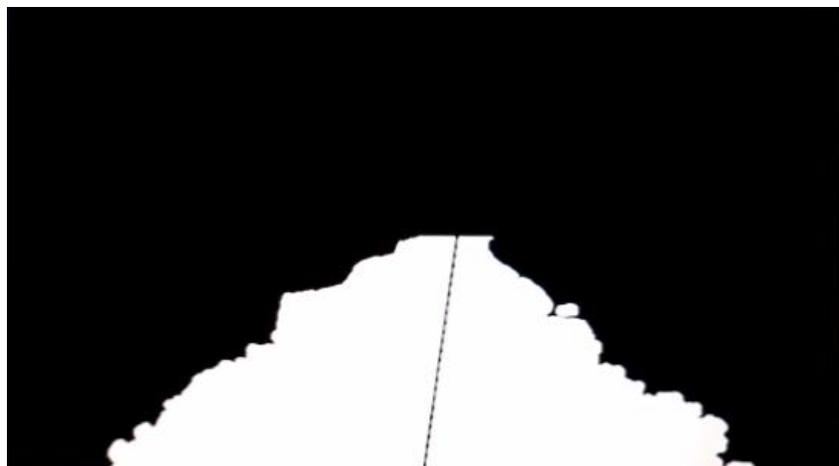
Fig. 2.26 Examples of video images of river scenes captured by the single-lens camera in an experiment carried out in cloudy weather



(a) $t = 10.0$ [s]



(b) $t = 20.0$ [s]



(c) $t = 30.0$ [s]

Fig. 2.27 Binary images converted from the video ones captured by the single-lens camera in an experiment carried out in cloudy weather



(a) $t = 10.0$ [s]



(b) $t = 20.0$ [s]



(c) $t = 30.0$ [s]

Fig. 2.28 Examples of video images of river scenes captured by the single-lens camera in an experiment carried out in sunny weather



(a) $t = 10.0$ [s]



(b) $t = 20.0$ [s]



(c) $t = 30.0$ [s]

Fig. 2.29 Binary images converted from the video ones captured by the single-lens camera in an experiment carried out in sunny weather



(a) $t = 10.0$ [s]



(b) $t = 20.0$ [s]



(c) $t = 30.0$ [s]

Fig. 2.30 Examples of video images of river scenes captured by the single-lens camera in an experiment carried out in very sunny weather



(a) $t = 10.0$ [s]



(b) $t = 20.0$ [s]



(c) $t = 30.0$ [s]

Fig. 2.31 Binary images converted from the video ones captured by the single-lens camera in an experiment carried out in very sunny weather

2.6 Conclusions

The summary of the results is shown below.

- (1) Firstly, the algorithm of image processing to segment the river area from the river scenes and to determine the flying direction had been developed in this study.
- (2) Then, the experiments to capture photo images of river scenes by using three different cameras, namely the single-lens, Raspberry Pi, and thermographic cameras had been carried out.
- (3) Furthermore, the segmentation of river scenes had been performed by using the image processing of the photo images of river scenes captured by the cameras. The result of segmentation shows that the binary images converted from the photo images captured by the single-lens camera were better than those captured by the Raspberry Pi and thermographic cameras.

Chapter 3

Multi-Rotor Drone with a Single-Lens Camera to Fly Autonomously Along a River

3.1 Introduction

The purposes of the study presented in this chapter are to develop an algorithm of image processing to determine flying directions of a multi-rotor drone using photo images of river scenes and to make a multi-rotor drone perform autonomous flights along a river using a single-lens camera and the image processing. In this study, an algorithm to perform an autonomous flight along a river was firstly developed. Then, flying experiments where a multi-rotor drone, namely AR. Drone 2.0 (Parrot) performed autonomous flights along a river were carried out.

Flying experiments where the multi-rotor drone performed manual flights were firstly carried out under off-line and with real-time image processing performed by a personal computer. Then, the algorithm to perform an autonomous flight was installed into the personal computer. After that, flying experiments where the multi-rotor drone performed autonomous flights along a river by using the single-lens camera and the image processing were carried out.

3.2 Flying Experiments Using a Multi-Rotor Drone Flight-Controlled Manually

3.2.1 Method

Figure 3.1 shows the multi-rotor drone (AR. Drone 2.0 (Parrot)) used in the experiments to perform manual flights. In the flying experiments, the multi-rotor drone was equipped with styrofoams in order to protect the multi-rotor drone from wetting by water. A styrofoam was attached on the top side of the multi-rotor drone and two parallel styrofoams under the bottom side of the multi-rotor drone.

Figure 3.2 shows the configuration of the multi-rotor drone. The two cameras, namely front and bottom cameras were mounted on the multi-rotor drone and connected to the main microprocessor. Then, the microprocessor was connected to the wireless connection (Wi-Fi) chipset. The multi-rotor drone was also equipped with some navigation sensors such as, a magnetometer, altimeter, gyroscope, accelerometer, and ultrasonic sensor. The multi-rotor drone was also equipped with four Electronic Speed Controllers (ESCs) to control the rotational speed of the four motors.

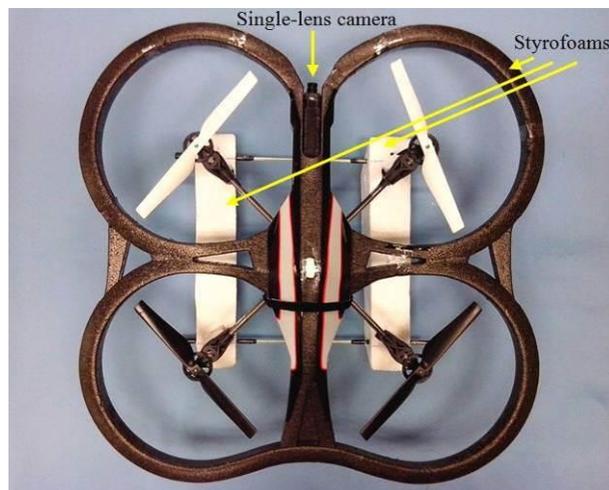


Fig. 3.1 Photo image of the multi-rotor drone (AR. Drone 2.0 (Parrot)) with a single-lens camera and styrofoams to protect the drone from wetting by water

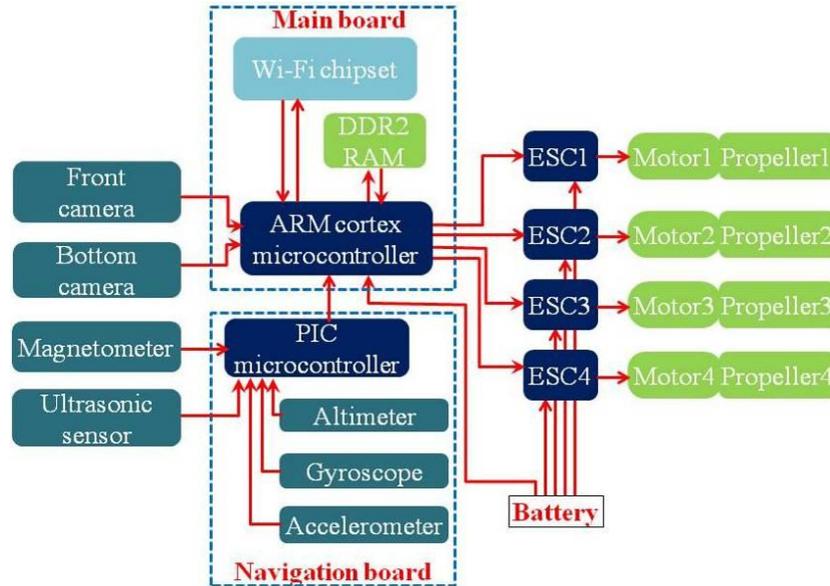


Fig. 3.2 Configuration of the multi-rotor drone (AR.Drone 2.0 (Parrot)) used in the experiments to perform manual flights

In the flying experiments, a personal computer operated by a person was used to control the multi-rotor drone and to perform the image processing. The specification of processor of the personal computer is Intel Core i3 U380 @ 1.33 GHz and the capacity of RAM is 2.00 GB. The multi-rotor drone and the personal computer were connected each other through the Wi-Fi as shown in Fig. 3.3. The maximum distance that the Wi-Fi can reach was 50 [m]. Furthermore, the program having the algorithms to perform a manual flight and the image processing developed by authors was installed inside the personal computer.

Firstly, flying commands were sent by the personal computer to the multi-rotor drone so that it can perform the manual flight over a river. Then, original photo images of river scenes were captured by the single-lens camera (front camera) installed on the multi-rotor drone. After that, the photo image data were automatically sent by the multi-rotor drone to the personal computer through the Wi-Fi. The photo images were then saved in the memory of the personal computer.

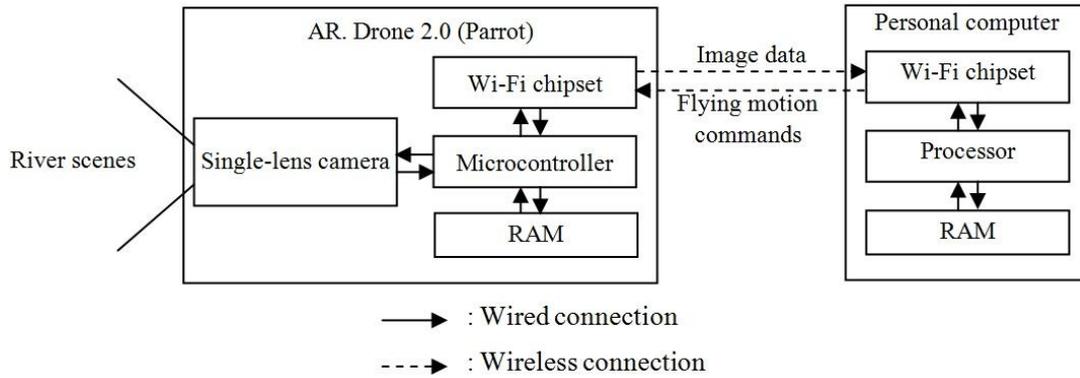


Fig. 3.3 Signal flow of system used in experiments by using a multi-rotor drone flight-controlled manually

Figure 3.4 shows the path and three consecutive stages of the flight in experiments using a multi-rotor drone flight-controlled manually. The three consecutive stages consist of flights from the start position to the center of the river (Stage 1), along the river (Stage 2) and from the center of the river to the end position (Stage 3).



- Red line : Flight path
- ① : Start position at the riverside
- ④ : End position at the riverside
- ① – ② : Stage 1 (Manual flight from the start position to the center of the river)
- ② – ③ : Stage 2 (Manual flight along the river)
- ③ – ④ : Stage 3 (Manual flight from the center of the river to the end position)

Fig. 3.4 Path and three consecutive stages of the flight in experiments using a multi-rotor drone flight-controlled manually

Figure 3.5 shows the person standing in a river to control the flights performed by the multi-rotor drone by using a personal computer installing CV Drone. The CV Drone is the Software Development Kit of AR. Drone integrated with Open Source Computer Vision (OpenCV). In the personal computer, a program to perform a manual flight and image processing was written using the CV Drone.



Fig. 3.5 Person standing in the river to control the flight performed by a multi-rotor drone by using a personal computer installing CV Drone

The flying experiments by using a multi-rotor drone flight-controlled manually were carried out with two stages of image processing, namely off-line and real-time image processing. In the flying experiments with off-line image processing, the image processing of captured photo images of river scenes were performed by the personal computer after the multi-rotor drone finished its flights. On the other hand, the image processing were performed by the personal computer in the same time when the multi-rotor drone performing its flights in the flying experiments with real-time image processing.

Firstly, the flying experiments by using the multi-rotor drone flight-controlled manually were carried out with off-line image processing. The multi-rotor drone was placed on the start position at the riverside to perform a manual flight as shown in Fig. 3.6. The multi-rotor drone took-off the ground at the riverside, then flew to the center of the river, then flew along the river to capture photo images of river scenes. Finally, the multi-rotor drone landed on the ground at the riverside. The photo images of river scenes were saved as videos with AVI format in the personal computer. The image processing of the photo images captured by the multi-rotor drone were performed by the personal computer after the multi-rotor drone completed the manual flights.



Fig. 3.6 Multi-rotor drone on the start position at the riverside in a flying experiment to perform a manual flight with off-line image processing

Then, the flying experiments by using the multi-rotor drone flight-controlled manually were carried out with real-time image processing. The stages of flights performed by the multi-rotor drone in flying experiments with real-time image processing were same as those of off-line image processing. The different thing is that the image processing of photo images of river scenes captured by the multi-rotor drone were performed by the personal computer in the same time when the multi-rotor drone capturing the photo images of river scenes. The results of the image processing were recorded by another camera holded by a person standing in the center of the river.

The rate of capturing photo images of the single-lens camera mounted on the multi-rotor drone was 30 [fps]. It means that thirty photo images of river scenes were captured by the single-lens camera in every second. The photo images of river scenes captured by the single-lens camera were saved as videos with AVI format in the personal computer.

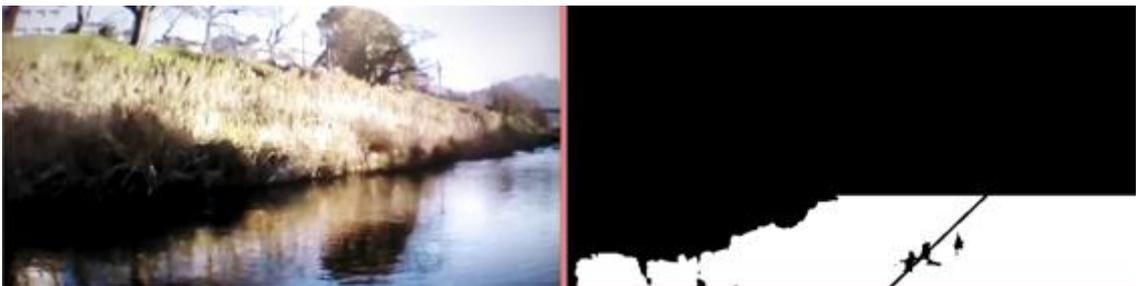
3.2.2 Results

3.2.2.1 Off-Line Image Processing

Figure 3.7 shows the result of off-line image processing of photo images of river scenes captured by the single-lens camera mounted on the multi-rotor drone. The rate of off-line image processing of the photo images was 2 [fps]. It means that two of thirty photo images of river scenes captured by the single-lens camera were processed by the personal computer in every second. The image processing were performed by the personal computer after the multi-rotor drone completed the autonomous flight. The flying directions were found on the binary images by using the image processing as shown in Fig. 3.7.



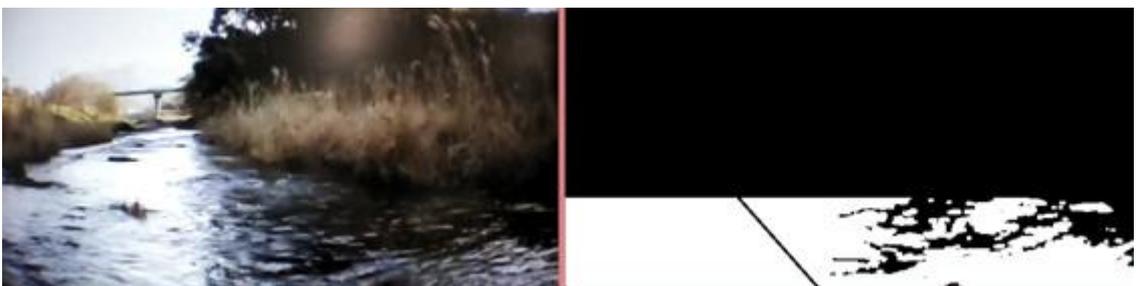
(1) $t = 25.0$ [s]



(2) $t = 45.0$ [s]



(3) $t = 80.0$ [s]



(4) $t = 120.0$ [s]

Fig. 3.7 Result of off-line image processing of photo images of river scenes captured by the single-lens camera mounted on the multi-rotor drone

3.2.2.2 Real-Time Image Processing

Figure 3.8 shows the result of real-time image processing of photo images of river scenes captured by the single-lens camera mounted on the multi-rotor drone. In the flying experiments with real-time image processing, the algorithm used to perform segmentation of river scenes was improved in order to increase the rate of image processing. Then, the rate of real-time image processing was 3 [fps]. It means that three of thirty photo images of river scenes captured by the single-lens camera were processed by the personal computer in every second.

The image processing were performed by the personal computer holded by a person standing in the river. Therefore, many reflections of person and trees existing around the river were appeared on the screen of the personal computer performing the image processing. The flying directions were found by using the image processing as shown in Fig. 3.8 (1), 3.8 (2), and 3.8 (3). However, sometimes many superflous areas were appeared in the lower side of the binary image. Therefore, the personal computer sometimes could not find correctly the flying direction due to the appearance of the superflous areas as shown in Fig. 3.8 (4). Then, the algorithm of image processing were improved in order to obtain better results in further flying experiments.



(1) $t = 35.0$ [s]



(2) $t = 37.0$ [s]



(3) $t = 47.0$ [s]



(4) $t = 51.0$ [s]

Fig. 3.8 Result of real-time image processing of photo images of river scenes captured by the single-lens camera mounted on the multi-rotor drone

3.3 Algorithm for Autonomous Flight Along a River

In order to perform an autonomous flight along a river, an algorithm consisting of three consecutive stages, namely flying to the river (1st stage), flying along the river (2nd stage), and flying to the riverside (3rd stage) was developed. The most important stage of the algorithm was the second stage. The algorithm of the second stage was constructed to perform an autonomous flight along a river. Therefore, the algorithm of the second stage was developed by considering the result of image processing on photo images of river scenes. As for the first and third stages, the algorithm of those stages were developed without image processing.

3.3.1 Flying to the River

Figure 3.9 shows the algorithm for the first stage of flight, namely flying to the river. In this stage, the multi-rotor drone is controlled manually to fly from the riverside to the center of the river. Firstly, the linear velocity in x -direction and z -direction, and the angular velocity in z -direction of the multi-rotor drone are defined. The linear velocity in x -direction of the multi-rotor drone denoted by v_x is defined as 0.5 [m/s]. The linear velocity in z -direction of the multi-rotor drone denoted by v_z is defined as 1.0 [m/s]. The angular velocity in z -direction of the multi-rotor drone denoted by ω_z is defined as 0.5 [rad/s]. Then, the multi-rotor drone takes-off the ground at a riverside and flies up to the altitude of 0.8 [m] from the ground. Then, the altitude of the drone is increased up to 2.5 [m]. After that, the drone flies to the center of the river and then turn to face the upriver side. In that position, the manual mode is changed to autonomous mode in order to perform the second stage of the flight.

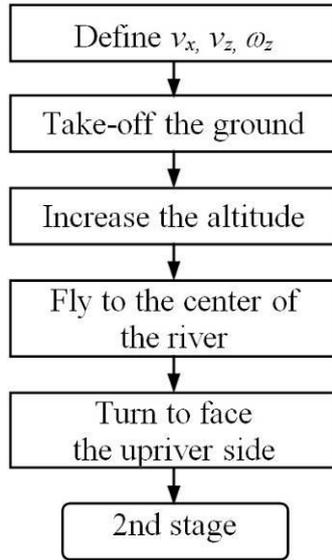


Fig. 3.9 Algorithm for the first stage of flight (flying to the river)

3.3.2 Flying Along the River

Figure 3.10 shows the algorithm for the second stage of the flight, namely flying along the river. In this stage, the flight time of the multi-rotor drone to perform the second stage, width of the binary image, linier velocity in x -direction of the multi-rotor drone, and proportional constant are firstly defined. The flight time of the multi-rotor drone to perform the second stage denoted by t_{23} is defined as 150 [s]. The width of the binary image denoted by w is defined as 640 [pixel]. The linier velocity in x -direction of the multi-rotor drone v_x is defined as 0.6 [m/s]. The proportional constant denoted by K_p is defined as 0.5 [rad/s]. The angular velocity in z -direction of the multi-rotor drone ω_z is calculated by using the value of x -coordinate of the end point coordinates of the flying direction found using the image processing, namely x_2 . After that, a certain range of ω_z is enforced to zero in order to avoid unstable condition of the multi-rotor drone. Finally, the multi-rotor drone autonomously flies along the river with the linier velocity in x -direction v_x and the angular velocity in z -direction ω_z until the flight time reaches

t_{23} . At the end of the second stage, the multi-rotor drone performs hovering for some seconds in order to perform the third stage.

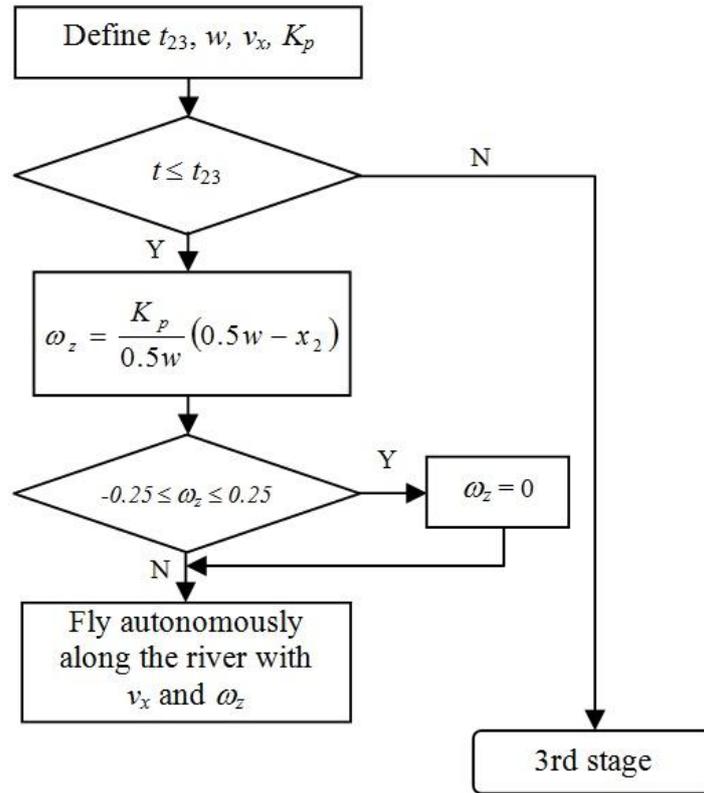


Fig. 3.10 Algorithm for the second stage of flight (flying along the river)

3.3.3 Flying to the Riverside

Figure 3.11 shows the algorithm for the third stage of the flight, namely flying to the riverside. In this stage, the flight time of the multi-rotor drone to perform the third stage, linear velocity in x -direction of the multi-rotor drone, and angular velocity in z -direction of the multi-rotor drone are firstly defined. The flight time of the multi-rotor drone to perform the third stage denoted by t_{34} is defined as 60 [s]. The linear velocity in x -direction of the multi-rotor drone v_x is defined as 0.5 [m/s]. The angular velocity in z -direction of the multi-rotor drone ω_z is defined as 0.5 [rad/s]. Then, the multi-rotor drone autonomously turns to face the riverside with the fixed v_x and ω_z and then flies to

the riverside with the v_x . Finally, the drone lands on the ground at the riverside to complete the autonomous flight a long a river.

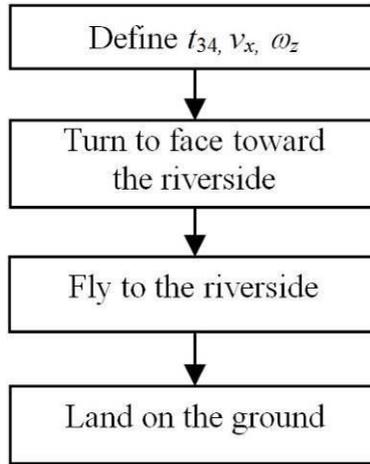


Fig. 3.11 Algorithm for the third stage of flight (flying to the riverside)

The algorithms of image processing presented in Section 2.3 and the autonomous flight presented in this section were then written as a computer program to perform an autonomous flight along a river by using CV Drone that is the Software Development Kit of AR.Drone 2.0.

3.4 Flying Experiments Using a Multi-Rotor Drone Flight-Controlled Autonomously

3.4.1 Method

Figure 3.12 shows the photo image of a multi-rotor drone (AR.Drone 2.0 (Parrot)) used in the flying experiments to perform autonomous flights. The experiments where the multi-rotor drone performs autonomous flights along a river were carried out on three different shapes of rivers, namely a straight river, a curved river with a large slope, and a curved river with a small slope. The multi-rotor drone with two parallel styrofoams attached at the bottom side of the drone was used in the flying experiments

carried out on the straight river and the curved river with a large slope as shown in Fig. 3.12 (a). In the flying experiments carried out on the curved river with a small slope, the multi-rotor drone with new parallel styrofoams as shown in Fig. 3.12 (b) was used. The position of the two parallel styrofoams were shifted to outside the work area of the propellers of the multi-rotor drone in order to minimize the air resistance appeared below the propellers when the propellers is rotating.

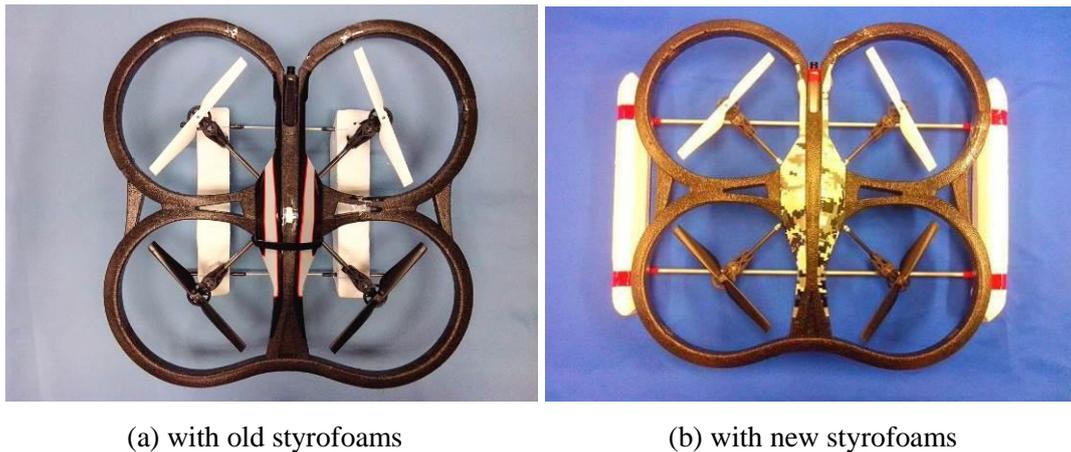


Fig. 3.12 Photo images of the multi-rotor drone (AR. Drone 2.0 (Parrot)) used in the flying experiments to perform autonomous flights along a river

In the flying experiments using the multi-rotor drone flight-controlled autonomously, a personal computer was used to perform real-time image processing. The multi-rotor drone and the personal computer were connected each other through the Wi-Fi as shown in Fig. 3.13. The maximum distance that the Wi-Fi can reach was 50 [m]. Furthermore, the program having the developed algorithms of the image processing and autonomous flight was installed inside the personal computer. The multi-rotor drone performs the autonomous flight, while the personal computer performs the image processing.

Original images of river scenes are captured by the single-lens camera mounted on the multi-rotor drone. After that, the image data are automatically sent to the personal computer. Then, the image processing are performed by the personal computer to determine the flying directions of the multi-rotor drone. The angular velocity in z -direction of the multi-rotor drone ω_z is depended on the x -coordinate of the end point coordinates of the flying direction, namely x_2 . Then, the angular velocity in z -direction ω_z is calculated by the personal computer using the following equation

$$\omega_z = \frac{K_p}{0.5w} (0.5w - x_2). \quad (3.1)$$

After that, a certain range of ω_z is enforced to zero in order to avoid unstable condition of the multi-rotor drone. Then, the new angular velocity in z -direction ω_z^* is expressed by

$$\omega_z^* = \begin{cases} \frac{K_p}{0.5w} (0.5w - x_2) & \text{if } \omega_z \leq -0.25 \text{ or } \omega_z \geq 0.25 \\ 0 & \text{if } -0.25 \leq \omega_z \leq 0.25. \end{cases} \quad (3.2)$$

Furthermore, flying motion commands were automatically generated and sent by the personal computer to the multi-rotor drone so that the multi-rotor drone can perform an autonomous flight.

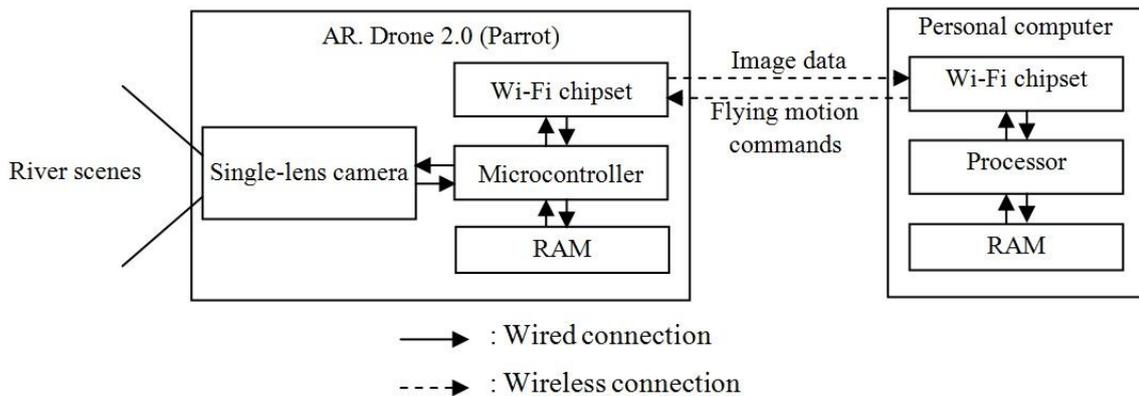


Fig. 3.13 Signal flow of the flying experiment where the multi-rotor drone performed an autonomous flight

The flying experiments using the multi-rotor drone flight-controlled autonomously were firstly carried out on a straight river, then on a curved river with a large slope, then on a curved river with a small slope. The scene of the straight river and the curved rivers, where the flying experiments where the multi-rotor drone performs autonomous flights, are shown in Fig. 3.14, Fig. 3.15, and Fig. 3.16, respectively. In the flying experiments on the curved river with a small slope, the personal computer used to perform real-time image processing was changed with a new one having higher specification of processor and memory in order to increase rate of the image processing. The specification of processor of the new personal computer is Intel Core i7 4510U @ 2.00 GHz and the capacity of RAM is 8.00 GB.



Fig. 3.14 Scene of the straight river where flying experiments using a multi-rotor drone flight-controlled autonomously were carried out

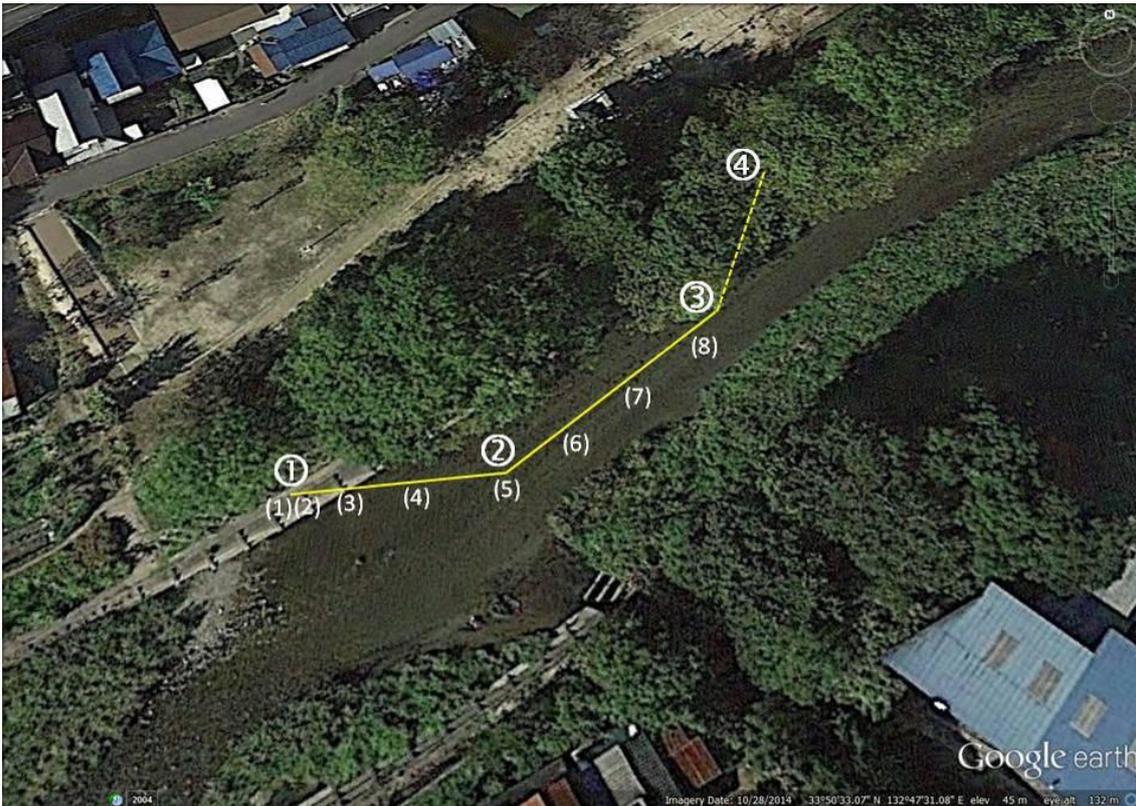


Fig. 3.15 Scene of the curved river with a large slope where flying experiments using a multi-rotor drone flight-controlled autonomously were carried out



Fig. 3.16 Scene of the curved river with a small slope where flying experiments using a multi-rotor drone flight-controlled autonomously were carried out

The flying experiments using the multi-rotor drone flight-controlled autonomously were carried out with three consecutive stages consisting of flights from the start position to the center of the river (Stage 1), along the river (Stage 2) and from the center of the river to the end position (Stage 3). The path and three consecutive stages of the autonomous flights performed by the multi-rotor drone in the flying experiments carried out on the straight river, curved river with a large slope, and curved river with a small slope are shown in Fig. 3.17, Fig. 3.18, and Fig. 3.19, respectively.



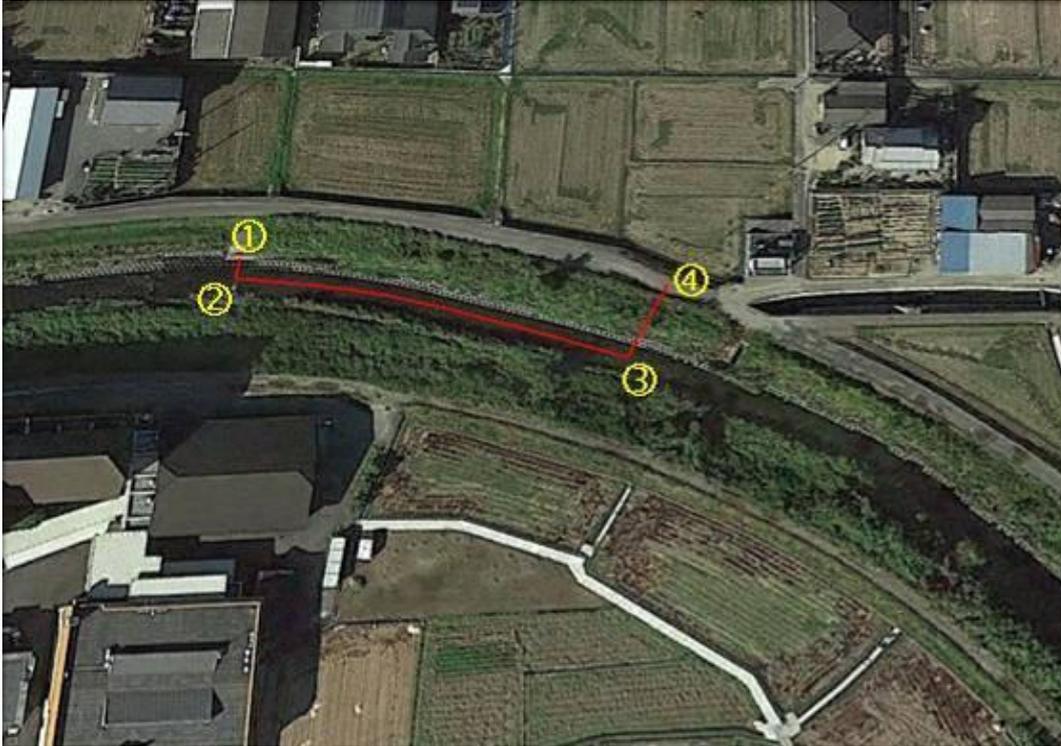
- Yellow line : Desired flight path (The distance from ② to ③ is 50 [m])
- ① : Start position at the riverside
 - ④ : End position at the riverside
 - ① – ② : Stage 1 (Autonomous flight from the start position to the center of the river)
 - ② – ③ : Stage 2 (Autonomous flight along the river)
 - ③ – ④ : Stage 3 (Autonomous flight from the center of the river to the end position)
 - (1) – (8) : Positions of the multi-rotor drone in the flying experiment
(Refer to Fig. 3.24)

Fig. 3.17 Desired path and three consecutive stages of the autonomous flight performed by the multi-rotor drone for flying experiments carried out on a straight river



- Red line : Desired flight path (The distance from ② to ③ is 100 [m])
- ① : Start position at the riverside
- ④ : End position at the riverside
- ① – ② : Stage 1 (Manual flight from the start position to the center of the river)
- ② – ③ : Stage 2 (Autonomous flight along the river)
- ③ – ④ : Stage 3 (Autonomous flight from the center of the river to the end position)

Fig. 3.18 Desired path and three consecutive stages of the autonomous flight performed by the multi-rotor drone for flying experiments carried out on a curved river with a large slope



- Red line : Desired flight path (The distance from ② to ③ is 90 [m])
- ① : Start position at the riverside
- ④ : End position at the riverside
- ① – ② : Stage 1 (Manual flight from the start position to the center of the river)
- ② – ③ : Stage 2 (Autonomous flight along the river)
- ③ – ④ : Stage 3 (Autonomous flight from the center of the river to the end position)

Fig. 3.19 Desired path and three consecutive stages of the autonomous flight performed by the multi-rotor drone for flying experiments carried out on the curved river with a small slope

Figure 3.20 and Figure 3.21 show the multi-rotor drone held by a person in the center of river to perform an autonomous flight in an experiment carried out on the curved rivers with a large and a small slope. In the flying experiments, the Stage 2 of the autonomous flight were firstly carried out. In this case, the multi-rotor drone was holded by a person in the center of the river as the start position of the multi-rotor drone to perform the autonomous flights. Then, the Stage 1 and Stage 2 of the autonomous flight were carried out on the curved river with a small slope as shown in Fig. 3.22 and Fig. 3.23.



Fig. 3.20 Multi-rotor drone held by a person in the center of river to perform an autonomous flight in an experiment carried out on the curved river with a large slope



Fig. 3.21 Multi-rotor drone held by a person in the center of river to perform an autonomous flight in an experiment carried out on the curved river with a small slope



Fig. 3.22 Scene of a flying experiment carried out on the curved river with a small slope using a multi-rotor drone to perform an autonomous flight along a river



Fig. 3.23 Person holding a personal computer performing image processing and another person recording a video of an experiment carried out on the curved river with a small slope

In the flying experiments carried out on the curved river with a small slope, the minimum and maximum threshold values of the “hue” and “saturation” parameters were set in the default value, namely 0 and 255. Only the minimum threshold value of the “value” parameter was adjusted to segment the river area from the river scenes. The minimum and maximum threshold values of the “value” parameter used in the experiments were 65 and 255, respectively.

3.4.2 Results

3.4.2.1 Flying Experiment on a Straight River

Figure 3.24 shows the photo images of the multi-rotor drone performing an autonomous flight in an experiment carried out on the straight river. In the experiment, the flying to the center of the river (Stage 1) and flying along a river (Stage 2) of the autonomous flight were performed by the multi-rotor drone. In the same time, the image processing were performed by a personal computer holded by a person in the center of the river.

At the beginning of the Stage 1, the multi-rotor drone was placed at the riverside as the start position to perform an autonomous flight as shown in Fig. 3.24(1). Then, the multi-rotor drone took-off the start position and performed hovering over the start position at the height of 0.8 [m] as shown in Fig. 3.24(2). After that, the multi-rotor drone flew to the center of the river as shown in Figs. 3.24(3) and 3.24(4). After reached the center of the river, the multi-rotor drone turned to face toward the upriver side. Then, the multi-rotor drone performed hovering over the center of the river for some seconds.

Furthermore, the multi-rotor drone performed the Stage 2 of the autonomous flight. In the Stage 2, the multi-rotor drone with a single-lens camera flew along the river as shown in Figs. 3.24(5), 3.24(6), and 3.24(7). Finally, the multi-rotor drone felled down on the surface of the river as shown in Fig. 3.24(8). In this flying experiment, the distance that the multi-rotor drone could fly was 50 [m]. The multi-rotor drone couldn't reach the end position because the flight distance might exceed the maximum distance that the wireless connection (Wi-Fi) can reach.



(1) $t = 0.0$ [s]



(2) $t = 5.0$ [s]



(3) $t = 10.0$ [s]



(4) $t = 15.0$ [s]



(5) $t = 20.0$ [s]



(6) $t = 25.0$ [s]



(7) $t = 30.0$ [s]



(8) $t = 33.0$ [s]

Fig. 3.24 Photo images of a multi-rotor drone performing an autonomous flight in an experiment carried out on a straight river (Refer to Fig. 3.17)

3.4.2.2 Flying Experiment on Curved Rivers

Figure 3.25 shows the photo images of the multi-rotor drone performing an autonomous flight in an experiment carried out on the curved river with a large slope. The photo images of the multi-rotor drone were captured by a person standing on the right side of the river. In the flying experiments carried out on the curved river with a large slope, the multi-rotor drone could not flew well from the riverside to to the center of river (Stage 1) due to the large difference of altitude between the riverside and the center of the river. Then, experiments where the multi-rotor drone performed the autonomous flight along a river (Stage 2) were carried out. In the experiments, the real-time image processing were performed by a personal computer holded by a person in the center of the river.

Firstly, the multi-rotor drone was holded by a person in the center of the river as the start position to perform the autonomous flight along the river as shown in Fig. 3.25(1). Then, the multi-rotor drone took-off the start position and performed hovering over the start position at the height of 0.8 [m] as shown in Fig. 3.25(2). After that, the multi-rotor drone flew along the river as shown in Figs. 3.25(3), 3.25(4), and 3.25(5). Finally, the multi-rotor drone felled down on the surface of the river as shown in Fig. 3.25(6). In this experiment, the distance that the multi-rotor drone could fly was 10 [m]. The multi-rotor drone could not fly farther than that distance due to the weak lift force of the multi-rotor drone. Therefore, the position of the two parallel styrofoams attached at the bottom side of the multi-rotor drone were shifted to outside the work area of the propellers in order to get rid of the air resistance when the propellers of the multi-rotor drone is rotating. Then, the multi-rotor drone with new position of two parallel styrofoams was used in the further flying experiments.



(1) $t = 0.0$ [s]



(2) $t = 4.0$ [s]



(3) $t = 8.0$ [s]



(4) $t = 12.0$ [s]



(5) $t = 16.0$ [s]



(6) $t = 20.0$ [s]

Fig. 3.25 Photo images of the multi-rotor drone performing an autonomous flight along a river in an experiment carried out on the curved river with a large slope

Figure 3.26 shows the photo images of river scenes captured by the single-lens camera mounted on the multi-rotor drone when performing an autonomous flight along the curved river with a large slope.

The photo image of a river scene when the multi-rotor drone was being holding at the center of the river as the start position is shown in Fig. 3.26(1). Then, the photo image of a river scene when the multi-rotor drone was performing a hovering over the start position is shown in Fig. 3.26(2). Furthermore, the photo images of river scenes when the multi-rotor drone was flying along the curved river are shown in Figs. 3.26(3), 3.26(4), 3.26(5), and 3.26(6).

After 10 [s] of the experiment, the data of photo images of river scenes sent by the multi-rotor drone to the personal computer could not be saved by the personal computer due to the distance between the multi-rotor drone and the personal computer became larger. Therefore, in the flying experiments carried out on the curved river with a small slope, the personal computer used to perform real-time image processing was changed with a new one having higher specification of processor and memory in order to increase rate of the image processing. The specification of processor of the new personal computer is Intel Core i7 4510U @ 2.00 GHz and the capacity of RAM is 8.00 GB.

In the flying experiments carried out on the curved river with a small slope, the distance between the multi-rotor drone and the personal computer was kept around 10 [m] in order to ensure that all data sent by the multi-rotor drone were saved by the personal computer. Two results of experiments where the multi-rotor drone performed autonomous flights along a river were presented in this research.



(1) $t = 0.0$ [s]



(2) $t = 2.0$ [s]



(3) $t = 4.0$ [s]



(4) $t = 6.0$ [s]



(5) $t = 8.0$ [s]



(6) $t = 10.0$ [s]

Fig. 3.26 Photo images of river scenes captured by the single-lens camera mounted on the multi-rotor drone when performing an autonomous flight along the curved river with a large slope

After finishing the flying experiments on the curved river with a large slope, then flying experiments on the curved river with a small slope were carried out. Figure 3.27 shows photo images of the multi-rotor drone performing the flight from ① to ② (Stage 1) in the first experiment carried out on the curved river with a small slope. The photo images of the multi-rotor drone were captured by a person standing on the ground at the riverside. At this stage, the multi-rotor drone was controlled manually using the personal computer. At the beginning of this stage, the multi-rotor drone took-off the start position at the riverside as shown in Fig. 3.27(1). The multi-rotor drone increased its altitude and performed hovering over the start position at the height of 0.8 [m] as shown in Fig. 3.27(2). After that, the multi-rotor drone increased its altitude up to 2.0 [m] and then flew to the center of the river as shown in Fig. 3.27(3). After reached the center of the river, the multi-rotor drone turned to face toward the upriver side. At this position, the manual control mode was changed to autonomous control one and the personal computer started the real-time image processing.

Figure 3.28 shows photo images of the multi-rotor drone performing an autonomous flight along a river, photo images of the river scenes captured by the single-lens camera mounted on the drone, and binary images converted from the photo images of the river scenes in the first experiment carried out on the curved river with a small slope from ② to ③ (Stage 2). At this stage, the multi-rotor drone autonomously flew along the curved river using the single-lens camera and the image processing. At the end of the flight, the multi-rotor drone landed on the grass at the riverside. In the experiment, the multi-rotor drone could autonomously fly along the curved river using the single-lens camera and the image processing as far as 78 [m].



(1) Take off ($t = 4$ [s])



(2) Hovering ($t = 8$ [s])



(3) Fly to the center of the river ($t = 17$ [s])

Fig. 3.27 Photo images of the multi-rotor drone performing the flight from ① to ② (Stage 1) in the first experiment carried out on the curved river with a small slope



(1-1) Flying along a river ($t = 30$ [s])



(1-2) Flying along a river ($t = 55$ [s])



(1-3) Flying along a river ($t = 80$ [s])



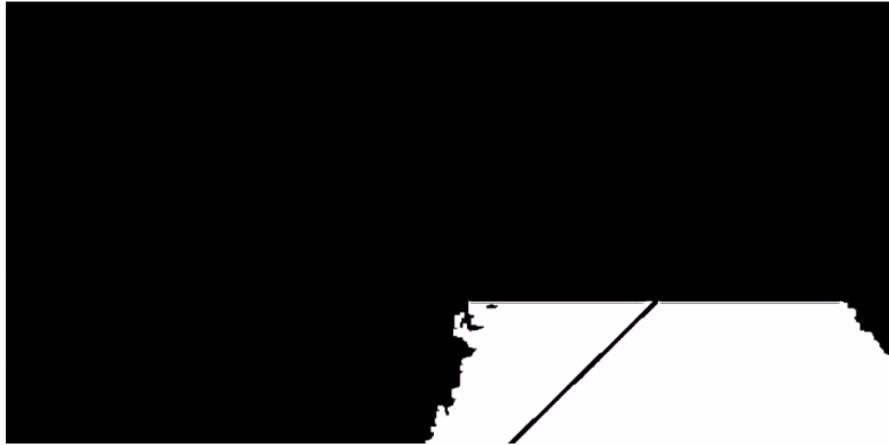
(2-1) Photo image of the river scene ($t = 30$ [s])



(2-2) Photo image of the river scene ($t = 55$ [s])



(2-3) Photo image of the river scene ($t = 80$ [s])



(3-1) Binary image converted from the photo image of the river scene ($t = 30$ [s])



(3-2) Binary image converted from the photo image of the river scene ($t = 55$ [s])



(3-3) Binary image converted from the photo image of the river scene ($t = 80$ [s])

Fig. 3.28 Photo images of the multi-rotor drone performing an autonomous flight along a river, photo images of the river scenes captured by the single-lens camera mounted on the drone, and binary images converted from the photo images of the river scenes in the first experiment carried out on the curved river with a small slope from ② to ③ (Stage 2)

After finished the first experiment, then the second experiment on the curved river with a small slope was carried out. Figure 3.29 shows photo images of the multi-rotor drone performing the flight from ① to ② (Stage 1) in the second experiment carried out on the curved river with a small slope. The photo images of the multi-rotor drone were captured by a person standing on the ground at the riverside. At this stage, the multi-rotor drone was controlled manually using the personal computer. At the beginning of this stage, the multi-rotor drone took-off the start position at the riverside as shown in Fig. 3.29(1). The multi-rotor drone increased its altitude and performed hovering over the start position at the height of 0.8 [m] as shown in Fig. 3.29(2). After that, the multi-rotor drone increased its altitude up to 2.5 [m] and then flew to the center of the river as shown in Fig. 3.29(3). After reached the center of the river, the multi-rotor drone turned to face toward the upriver side. At this position, the manual control mode was changed to autonomous control one and the personal computer started the real-time image processing.

Figure 3.30 shows photo images of the multi-rotor drone performing an autonomous flight along a river, photo images of the river scenes captured by the single-lens camera mounted on the drone, and binary images converted from the photo images of the river scenes in the second experiment carried out on the curved river with a small slope from ② to ③ (Stage 2). At this stage, the multi-rotor drone autonomously flew along the curved river using the single-lens camera and the image processing. At the end of the flight, the multi-rotor drone landed on the grass at the riverside. In this experiment, the multi-rotor drone could autonomously fly along the curved river using the single-lens camera and the image processing as far as 83 [m].



(1) Take off ($t = 3$ [s])



(2) Hovering ($t = 6$ [s])



(3) Fly to the center of the river ($t = 15$ [s])

Fig. 3.29 Photo images of the multi-rotor drone performing the flight from ① to ② (Stage 1) in the second experiment carried out on the curved river with a small slope



(1-1) Flying along a river ($t = 30$ [s])



(1-2) Flying along a river ($t = 55$ [s])



(1-3) Flying along a river ($t = 80$ [s])



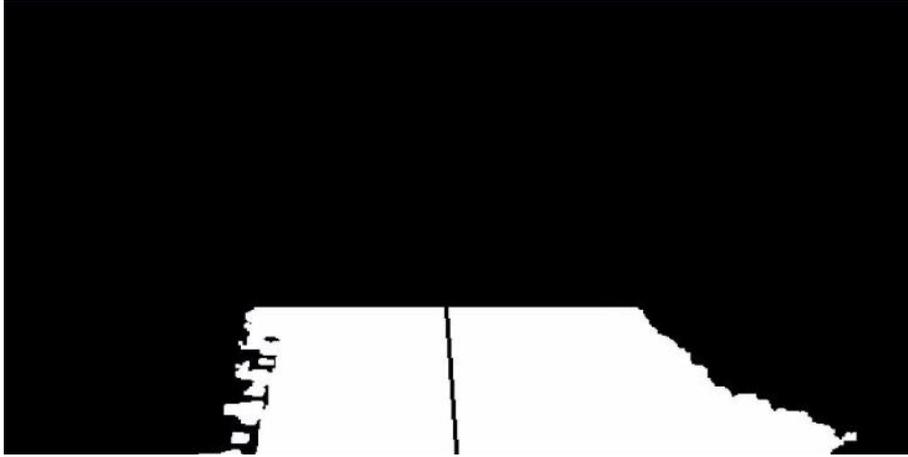
(2-1) Photo image of the river scene ($t = 30$ [s])



(2-2) Photo image of the river scene ($t = 55$ [s])



(2-3) Photo image of the river scene ($t = 80$ [s])



(3-1) Binary image converted from the photo image of the river scene ($t = 30$ [s])



(3-2) Binary image converted from the photo image of the river scene ($t = 55$ [s])



(3-3) Binary image converted from the photo image of the river scene ($t = 80$ [s])

Fig. 3.30 Photo images of the multi-rotor drone performing an autonomous flight along a river, photo images of the river scenes captured by the single-lens camera mounted on the drone, and binary images converted from the photo images of the river scenes in the second experiment carried out on the curved river with a small slope from ② to ③ (Stage 2)

3.5 Conclusions

The summary of the results is shown below.

- (1) Firstly, the algorithm to perform an autonomous flight along a river had been developed in this study.
- (2) Then, the experiments where the multi-rotor drone performs autonomous flights along a river had been carried out. The experimental result shows that the multi-rotor drone could autonomously fly along the curved river using a single-lens camera and the image processing as far as 83 [m].

Chapter 4

3D Map Modeling of Environment around a River

4.1 Introduction

The purposes of the study presented in this chapter are to create a 3D textured map by using image processing of static photo images and video images of an environment around a river and a 3D map model of the environment by using a 3D color printer. In this study, a flying experiment where a multi-rotor drone with a fisheye-lens camera captured static photo images and video images of the targeted environment around the river was firstly carried out. Then, image processing consisting of five main stages were performed by using a photogrammetric image processing software (Agisoft PhotoScan Professional ver.1.1). In the first stage, the captured images were loaded to the workspace pane of the Agisoft PhotoScan. In the second stage, the images were aligned in order to find matching points between overlapping images. The positions and orientations of the images were estimated and “sparse point clouds” were built by the Agisoft PhotoScan based on the matching points. In the third stage, “dense point clouds” were built by using the estimated positions and orientations and the “sparse point clouds”. In the fourth stage, polygonal meshes were built by using the “dense point clouds”. In the fifth stage, 3D textured maps were built by using the polygonal meshes. A thickened 3D map was then built by using the 3D textured map of the static photo images. Finally, the 3D map model of the environment around the river was created by using the thickened 3D map and a 3D color printer (IRIS 3D printer).

4.2 Flying Experiment Using a Multi-Rotor Drone with a Fisheye-Lens Camera

4.2.1 Method

Figure 4.1 shows the multi-rotor drone (Parrot Bebop Drone) with a fisheye-lens camera used in the study presented in this chapter. The multi-rotor drone equips with a fisheye-lens camera, a memory card, and a GPS. Static photo images and video images of an environment around a river are captured by the fisheye-lens camera mounted in the front side of the multi-rotor drone and stored in the memory card mounted inside the multi-rotor drone. In addition, 3D coordinates of the multi-rotor drone's positions when capturing the static photo images are measured by the GPS mounted inside the multi-rotor drone.

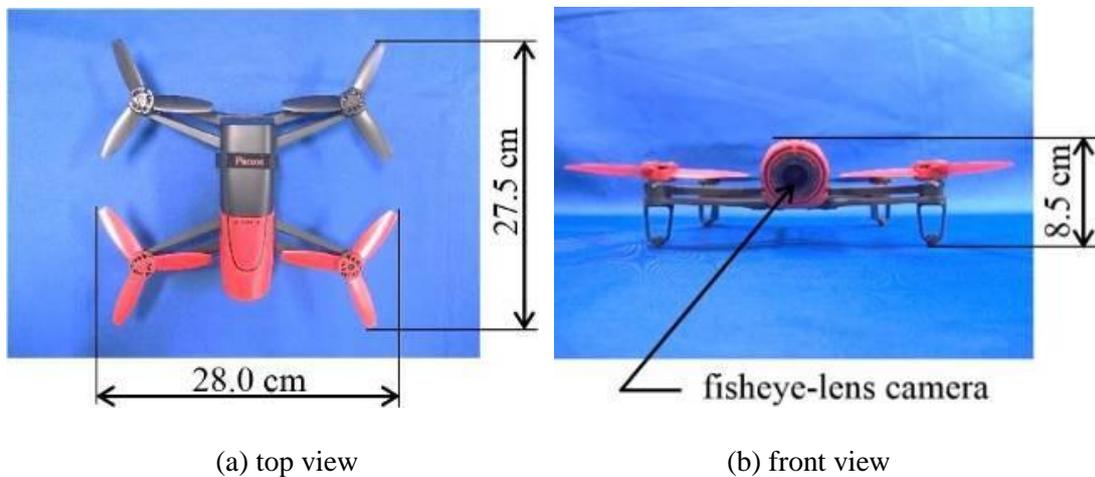


Fig. 4.1 Multi-rotor drone (Parrot Bebop Drone) with a fisheye-lens camera

Figure 4.2 shows the targeted environment around a river that is used in this study. The environment is located in a considerable distance from resident's houses, namely around 150 [m]. The environment used in the experiment consists of a curved river with the average width of 5 [m], bush and farmlands at both sides of the river. The static photo images and video images of the environment are captured using a fisheye-lens

camera mounted on the multi-rotor drone with a resolution of 14 [Mpx] and a field of view of 180 [°]. With the large field of view, the fisheye-lens camera mounted on the multi-rotor drone (Parrot Bebop Drone) can capture images of wide areas of targeted environment.



Fig. 4.2 Environment around a river that is used in this study

Figure 4.3 shows the positions where static photo images and video images of the environment around the river are captured by the fisheye-lens camera mounted on the multi-rotor drone in two straight lines. The drone firstly performed a flight to capture static photo images and video images in the first line. After finishing the flight in the first line, the multi-rotor drone repeated the flight in the second line which parallels to the first one. The 3D coordinates of positions to capture the images were measured by the GPS and saved in the memory card mounted inside the multi-rotor drone.

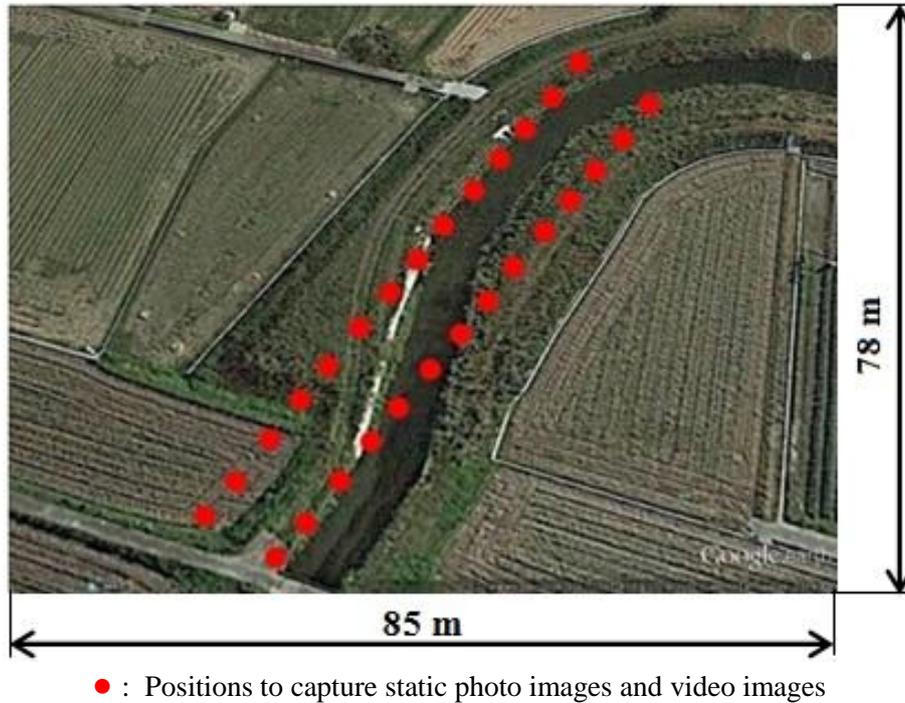


Fig. 4.3 Positions where static photo images and video images were captured by the fisheye-lens camera mounted on the multi-rotor drone

Figure 4.4 shows the scene in which static photo images and video images of the environment around the river were captured by the fisheye-lens camera installed on the Parrot Bebop Drone flight-controlled manually. The tablet computer to control the multi-rotor drone communicated with the multi-rotor drone through a wireless connection (Wi-Fi). The maximum distance that the Wi-Fi can reach is 150 [m].

Firstly, the multi-rotor drone took off the ground at the riverside. Then, the altitude of the multi-rotor drone was increased until 15 [m]. At the altitude, the multi-rotor drone carried out a hovering for several seconds. After that, the multi-rotor drone performed the flights to capture the static photo images and video images of the environment around the river in the two straight lines. The static photo images were captured in every 6 [s]. The 3D coordinates of the multi-rotor drone's positions when capturing the static photo images were measured by the GPS. Then, the static photo images with 3D

coordinate data were saved in the memory card mounted inside the drone. The video images were captured in 30 [fps]. However, the video images were saved without 3D coordinates of the drone's positions when capturing the video images.



Fig. 4.4 Scene in which static photo images and videos images of the environment around the river were captured by the fisheye-lens camera installed on the Parrot Bebop Drone flight-controlled manually

4.2.2 Result

4.2.2.1 Static Photo Images of an Environment around a River

Figure 4.5 shows the examples of overlapping static photo images of the environment around the river captured in every 6 [s] by the fisheye-lens camera mounted on the multi-rotor drone. There are 45 overlapping static photo images captured and saved as JPEG format in the flying experiment. The length and the width of the static photo images are 1,920 [pixels] and 1,088 [pixels], respectively. Each static photo image is overlapping by 80 % with the adjacent photo image.



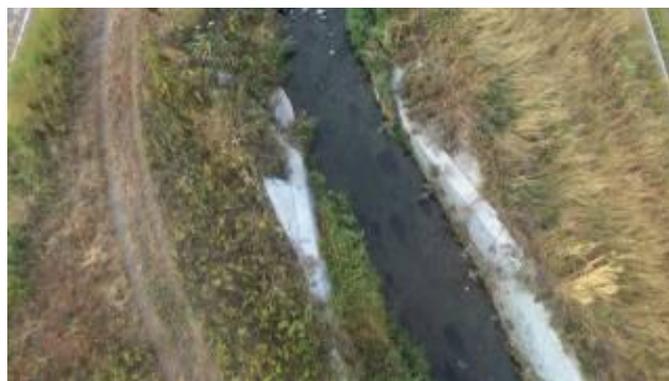
(a) $t = 13$ [s]



(b) $t = 19$ [s]



(c) $t = 25$ [s]



(d) $t = 31$ [s]

Fig. 4.5 Examples of overlapping static photo images of the environment around the river captured in every 6 [s] by the fisheye-lens camera mounted on the multi-rotor drone

4.2.2.2 Video Images of an Environment around a River

Figure 4.6 shows the examples of overlapping video images of the environment around the river captured by the fisheye-lens camera mounted on the multi-rotor drone in the flying experiment. The 89 overlapping video images were extracted from the two videos captured by the fisheye-lens camera on the two parallel path of flight.

The video images were slightly different with the static photo images captured by the fisheye-lens camera in the flying experiment. The length and the width of the video images are 1,920 [pixels] and 1,080 [pixels], respectively while those of the static photo images are 1,920 [pixels] and 1,088 [pixels], respectively. The dimension of the video images were a bit smaller than those of the static photo images captured by the fisheye-lens camera. In addition, the file size of the video images were also a bit smaller than those of the static photo images. Then, unlike the static photo images, the video images extracted from the videos captured by the fisheye-lens camera do not have 3D coordinate data of multi rotor drone's position when capturing the video images. Based on those conditions, it is reasonable that the qualities of the static photo images were better than that of the video images.



(a) Frame No. 23



(b) Frame No. 24



(c) Frame No. 60



(d) Frame No. 61

Fig. 4.6 Examples of overlapping video images extracted from videos of environment around the river captured by the fisheye-lens camera mounted on the multi-rotor drone

4.3 3D Mapping Using Image Processing of Static Photo Images and Video Images

In order to build 3D textured maps of the environment around the river, the captured static photo images and video images were processed by using the Agisoft PhotoScan Professional (Ver.1.1). There are five consecutive stages in the 3D mapping, namely the loading and the aligning the images, the building the “dense point clouds”, the polygonal meshes, and the 3D textured maps as shown in Fig. 4.7. The image processing were performed using the commands in the “Workflow” menu of the Agisoft.

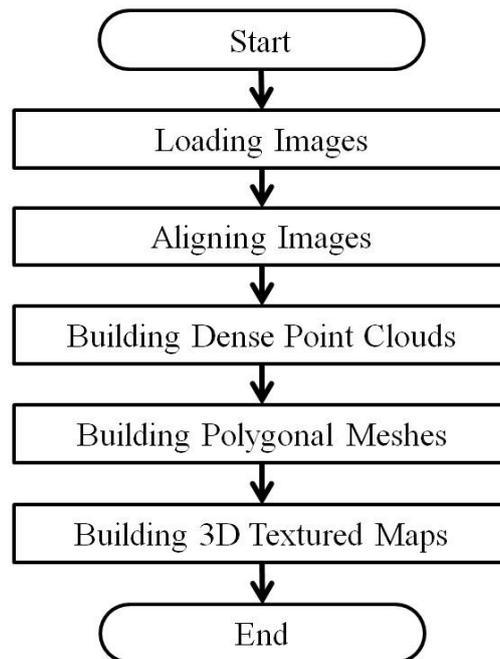


Fig. 4.7 Five consecutive stages in the 3D mapping to create 3D textured maps using static photo images and video images of river scenes

4.3.1 Loading Images

Firstly, two project files of Agisoft PhotoScan were opened to process the static photo images and the video images of the environment around the river. Then, the 45 overlapping static photo images were loaded to the workspace pane in the first project file. The 3D coordinates of the multi-rotor drone’s positions when capturing the static

photo images measured by the GPS were detected by the Agisoft. After that, the positions of the multi-rotor drone when capturing the 45 static photo images displayed in two lines in the model pane of the Agisoft as shown in Fig. 4.8. Based on the positions, 18 static photo images having no overlapping area with the static photo images in the other line were removed from the workspace pane.

In order to obtain good result, the static photo images were calibrated by the Agisoft. In the tools menu of the Agisoft, the “camera calibration” was selected. Then, the type of camera was selected in the drop down menu of the “camera type”. There are three types of camera in the menu, namely “frame”, “fisheye”, and “spherical”. Then, the “fisheye” was selected in this stage.

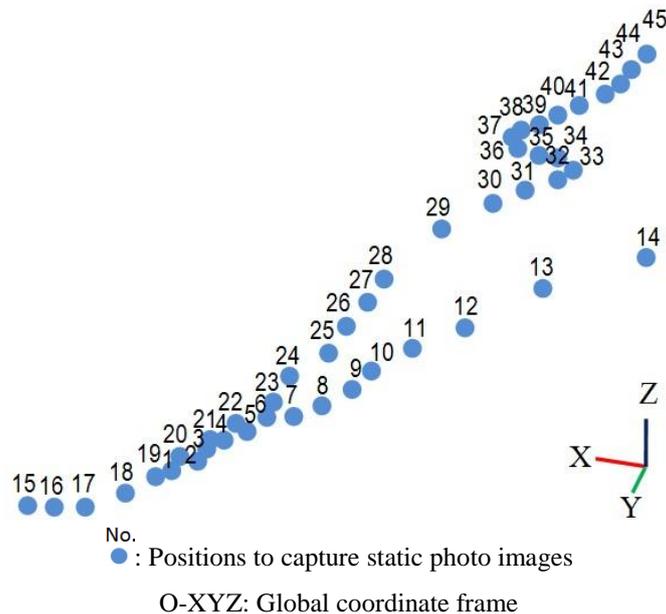


Fig. 4.8 Positions of the multi-rotor drone when capturing the forty-five static photo images displayed in two lines in the model pane of the Agisoft

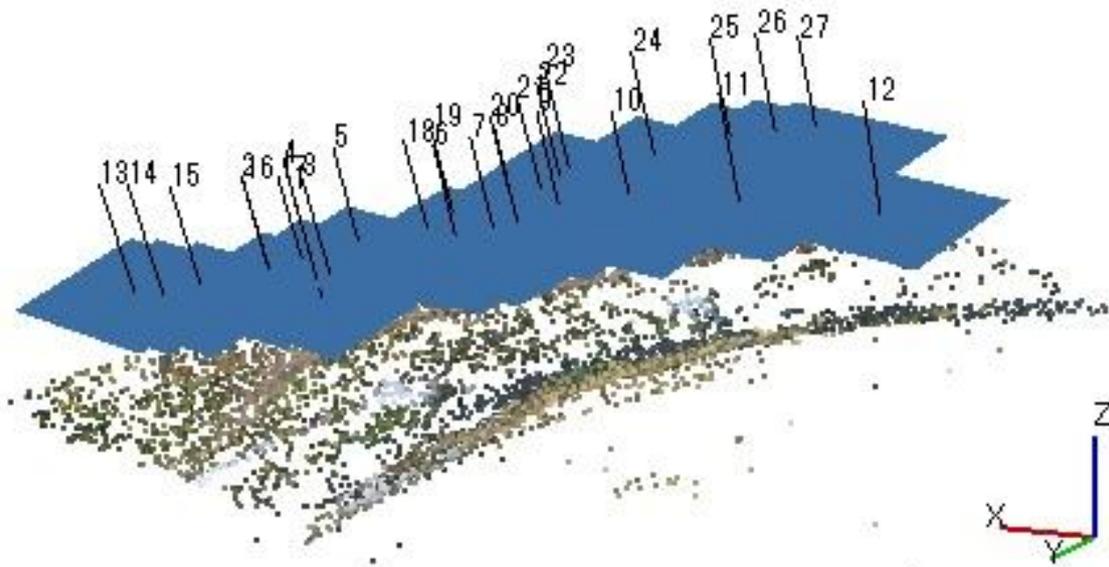
Furthermore, the 89 overlapping video images of the environment around the river were loaded to the workspace pane in the second project file. However, there was no figure of positions to capture the video images displayed in the model pane due to that the video images do not have 3D coordinates of the positions.

4.3.2 Aligning Images

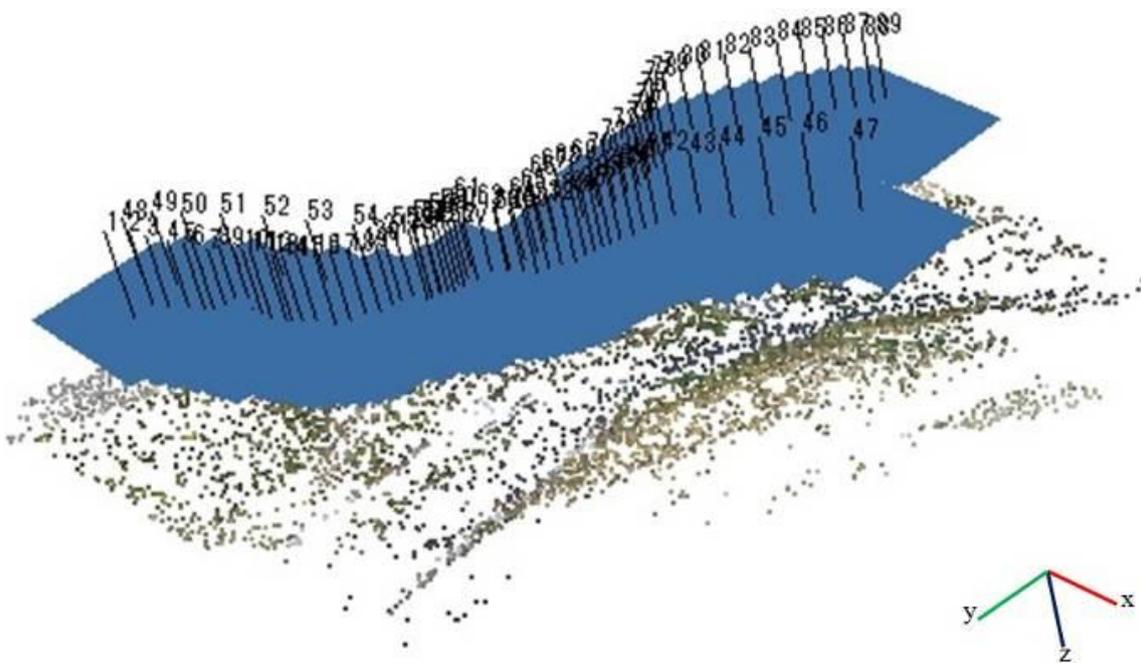
In this stage, the loaded static photo images and video images were aligned by the Agisoft in order to estimate the positions and the orientations of those images and to build “sparse point clouds” as the initial data of 3D textured maps. The accuracy of the positions and the orientations of those images was chosen by selecting an accuracy level from the “Accuracy” menu of the Agisoft. There are three levels of accuracy in the “Accuracy” menu, namely “high”, “medium”, and “low”. Then, the “medium” accuracy level was selected in this stage in order to reduce the complexity of the surface of 3D textured maps and to reduce the processing time of those images.

After completing the alignment parameters, the alignment process were started by the Agisoft. The point data of the static photo images were firstly detected by the Agisoft. Then, the point data of each static photo image were compared to their adjacent ones in order to detect matching points between the two static photo images. After that, the matching points and 3D coordinate data were used to estimate the positions and the orientations of static photo images and to build a “sparse point cloud”. The 7,698 points were generated in the “sparse point cloud” of the static photo images.

On the other hand, the same process to build a “sparse point cloud” was performed by the Agisoft using the video images. Due to the absence of 3D coordinate data, only the point data of the video images were used to estimate the positions and the orientations of video images and to build a “sparse point cloud”. The 8,240 points were generated in the “sparse point cloud” of the video images. Figure 4.9 shows the estimated positions and orientations of the 27 static photo images and the 89 video images shown by the squares and the “sparse point clouds” below the positions displayed in the model pane of the Agisoft.



(a) Using twenty-seven static photo images



(b) Using eighty-nine video images

O-XYZ: Global coordinate frame

O-xyz: Image coordinate frame

■: Square, ^{No.} | : Number of image, ☉: Sparse point cloud

Fig. 4.9 Estimated positions and orientations of static photo images and video images shown by the squares and the sparse point clouds below the positions displayed in the model pane of the Agisoft

The “sparse point clouds” were 3D point data built from the 2D static photo images and video images. The points were the matching points found by the Agisoft after compared the data of the 2D static photo images and video images. Due to the number of the video images was larger than that of the static video images, the number of matching points found after compared the data of the video images was larger than that found after compared the data of the static photo images.

Furthermore, the qualities of the 27 static photo images and the 89 video ones were estimated by the Agisoft by using the point data of the images. The quality of an image is treated by a numerical value in a range from 0.000 to 1.000 where a large value means high quality. The images whose qualities are less than 0.500 are not feasible in using for further stages.

To perform the image quality estimation, all images in the photos pane of the Agisoft were firstly selected. Then, the quality estimation was performed by selecting “Estimate Image Quality” in the “Context” menu of the Agisoft. As the result, the qualities of the static photo images and the video ones were shown in “Quality” column on the Photos pane of the Agisoft.

Table 4.1 and 4.2 shows the qualities of the 27 static photo images and 89 video images captured by the fisheye-lens camera estimated by the Agisoft. The result of the image quality estimation shows that the qualities of the static photo images and the video ones were more than 0.500. The average qualities of the static photo images and the video ones were 0.779 and 0.766, respectively. It means that the whole images captured by the fisheye-lens camera were feasible to use them for further stages. Then, the average quality of the static photo images was higher than the video ones.

Table 4.1 Qualities of the twenty-seven static photo images captured by the fisheye-lens camera analyzed by the Agisoft

Image Number	Image Quality
1	0.745083
2	0.743067
3	0.735803
4	0.745009
5	0.753286
6	0.803459
7	0.809441
8	0.812982
9	0.826883
10	0.801322
11	0.827492
12	0.834738
13	0.772399
14	0.722790
15	0.728472
16	0.729319
17	0.742430
18	0.765186
19	0.799133
20	0.843583
21	0.808682
22	0.827436
23	0.811727
24	0.740953
25	0.766224
26	0.787132
27	0.766841
Average	0.779662

Table 4.2 Qualities of the eighty-nine video images captured by the fisheye-lens camera analyzed by the Agisoft

Image Number	Image Quality
1	0.814737
2	0.823035
3	0.820135
4	0.792666
5	0.785837
6	0.768261
7	0.770142
8	0.772941
9	0.753259
10	0.740550
11	0.714179
12	0.711131
13	0.721076
14	0.707799
15	0.701586
16	0.697519
17	0.712061
18	0.709344
19	0.714943
20	0.694715
21	0.697750
22	0.727151
23	0.724875
24	0.730003
25	0.724116
26	0.725415
27	0.728170
28	0.763218
29	0.765347
30	0.762360

Image Number	Image Quality
31	0.767900
32	0.764077
33	0.806768
34	0.799969
35	0.783147
36	0.786612
37	0.787240
38	0.801032
39	0.789796
40	0.807089
41	0.800883
42	0.795212
43	0.803891
44	0.803213
45	0.798060
46	0.823472
47	0.815056
48	0.799072
49	0.815814
50	0.798883
51	0.796868
52	0.795996
53	0.756932
54	0.711621
55	0.706715
56	0.713550
57	0.733890
58	0.726249
59	0.728190
60	0.730146
61	0.745934

Image Number	Image Quality
62	0.738752
63	0.736204
64	0.762301
65	0.783635
66	0.773437
67	0.792798
68	0.789837
69	0.831052
70	0.821158
71	0.839559
72	0.813140
73	0.775406
74	0.825668
75	0.801725
76	0.814279
77	0.802451
78	0.795808
79	0.771571
80	0.751525
81	0.739230
82	0.722768
83	0.744568
84	0.770644
85	0.764121
86	0.781175
87	0.780646
88	0.778877
89	0.753016
Average	0.766550

4.3.3 Building Dense Point Clouds

In the third stage, the “sparse point clouds” and the estimated positions and orientations of the static photo images and the video images were used by the Agisoft to build “dense point clouds”. In this stage, a lot of additional 3D points were generated by the Agisoft to build the “dense point clouds”. To perform this stage, the “Build Dense Cloud” command was selected from the “Workflow” menu of the Agisoft. Then, the “quality” parameter was selected from the dialog box displayed by the Agisoft.

There are five quality parameters to build the “dense point clouds”, namely “ultra-high”, “high”, “medium”, “low” and “lowest”. The higher the quality becomes, the more complex the resulting “dense point clouds” will become. The more complex the “dense point clouds”, the more complex the 3D textured maps will become. Then, it was difficult to create a 3D map model of a complex 3D textured map by using the IRIS 3D printer that was used in this research. Therefore, the “dense point clouds” were built with the “medium” quality in order to reduce the complexity of the “dense point clouds” and 3D textured maps. In this stage, the 749,813 points were generated in the “dense point cloud” of the static photo images and the 977,575 points in the “dense point cloud” of the video ones. Figure 4.10 shows the “dense point clouds” built from the “sparse point clouds” and the estimated positions and orientations of the static photo images and video images.



(a) Using twenty-seven static photo images



(b) Using eighty-nine video images

O-XYZ: Global coordinate frame
 O-xyz: Image coordinate frame

Fig. 4.10 Dense point clouds built using the sparse point clouds and the estimated positions and orientations of the static photo images and video images

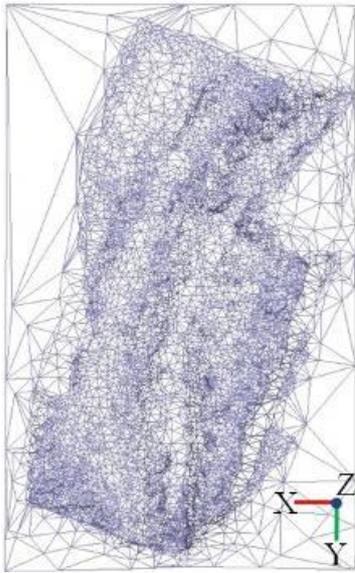
4.3.4 Building Polygonal Meshes

In the fourth stage, polygonal meshes were built by using the “dense point clouds” of the static photo images and video images. To perform this stage, the “Build Mesh” command was selected from the “Workflow” menu of the Agisoft. Then, the “Surface type” and “Source data” parameters were selected from the dialog box displayed by the Agisoft.

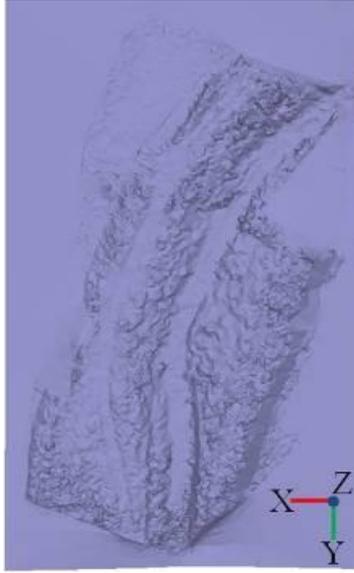
There are two types of surfaces on the parameters to build the polygonal meshes, namely the options of “arbitrary” and “height field”. The option of “arbitrary” is used for the surfaces of isolated objects and the option of “height field” is used for the surfaces of terrains. Then, the option of “height field” was selected as the type of surface to build the polygonal meshes of the static photo images and video images.

There are two options of the source of data that could be chosen to build the polygonal meshes, namely “Sparse cloud” and “Dense cloud”. Then, the option of “Dense cloud” was selected as the source of data to build the polygonal meshes of the static photo images and video images.

The polygonal meshes were displayed in three types of views, namely “wireframe”, “solid”, and “shaded” views. In the “shaded” views, the existing texture data in the “dense point clouds” of the static photo images and video images were used to texture the “solid” view of the polygonal meshes. However, some parts in the “dense point clouds” do not have texture data. Therefore, some parts in the “shaded” views also do not have texture data. Then, the additional textures were generated by the Agisoft in the next stage. Figure 4.11 shows the polygonal meshes built by using the “dense point clouds” of the static photo images and video images.



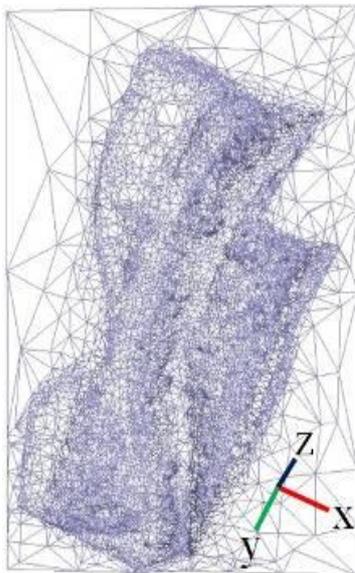
(a.1) Wireframe view built by using static photo images



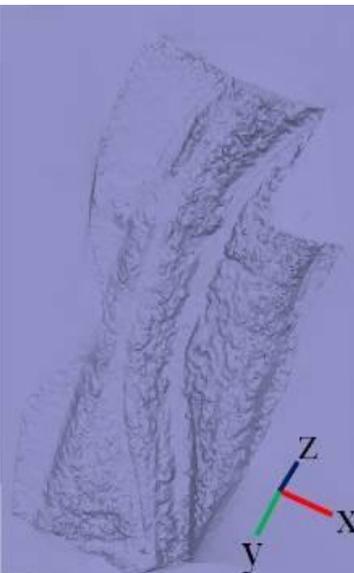
(a.2) Solid view built by using static photo images



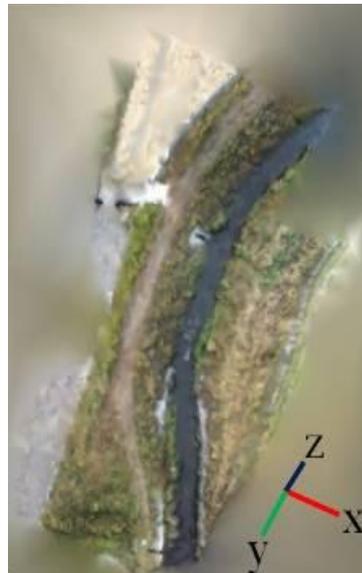
(a.3) Shaded view built by using static photo images



(b.1) Wireframe view built by using video images



(b.2) Solid view built by using video images



(b.3) Shaded view built by using video images

O-XYZ: Global coordinate frame
O-xyz: Image coordinate frame

Fig. 4.11 Polygonal meshes built by using the dense point clouds of static photo images and video images

4.3.5 Building 3D Textured Maps

In the fifth stage, 3D textured maps were built by using the “shaded” views of the polygonal meshes of the static photo images and video images. To perform this stage, the “Build Texture” command was selected from the “Workflow” menu of the Agisoft. Then, the “Mapping mode” parameter was selected from the dialog box displayed by the Agisoft.

There are six modes in the drop down menu of the “Texture mapping” modes, namely “Generic”, “Adaptive orthophoto”, “Orthophoto”, “Spherical”, “Single photo”, and “Keep uv”. The selected mode was used to determine the way of texturing the 3D map. The best mode to process aerial photographs is “Orthophoto” mode. Then, the option of “Orthophoto” mode was selected to create the 3D textured maps of the static photo images and video images.

There are four modes in the drop down menu of the “Blending” modes, namely “Mosaic”, “Average”, “Max intensity”, and “Min intensity”. The selected mode was used to determine the way of combining pixel values of different images in the final 3D textured map. Then, the option of “Mosaic” mode was also selected to create the 3D textured maps of the static photo images and video images were built with”.

Figure 4.12 shows the 3D textured maps showing the geometry and the textures of the environment around the river built using the “shaded” views of the polygonal meshes of the static photo images and video images. The texture data of the “shaded” views were firstly read by the Agisoft. Then, new texture data calculated using the existing ones were generated by the Agisoft for the parts without texture data. Furthermore, 3D textured maps were built by texturing the whole parts of the “shaded” views using the existing and new texture data.



(a) Using twenty-seven static photo images



(b) Using eighty-nine video images

O-XYZ: Global coordinate frame

O-xyz: Image coordinate frame

Fig. 4.12 3D textured maps showing the geometry and the texture of the environment around the river built by using the polygonal meshes of static photo images and video images

The 3D textured maps represent the geometry and the texture of the environment around the river. The lengths and the widths of the maps representing the environment correspond to 27 [m] and 20 [m], respectively for the actual one. It can be seen that the 3D textured map of the static photo images is better than the video ones due to that the average quality of the static photo images is higher than that of the video ones. In addition, the orientation of the 3D textured map of static photo images is more accurate than that of the video ones due to that the static photo images have 3D coordinate data while the video ones do not have 3D coordinate data. Then, the 3D textured map of static photo images was used in the 3D map modeling to create the 3D map model of the environment around the river.

4.4 3D Map Modeling Using 3D Textured Map of Static Photo Images

In order to obtain better 3D visualization of the environment around the river, a 3D map model was built by using a selected area in the 3D textured map of the static photo images. In this 3D map modeling, the selected area in the 3D textured map of the static photo images was thickened by using the Blender (Ver.2.7.6b) that is an open software of 3D computer graphics. Then, the 3D map model was built using the thickened 3D map. After that, the 3D printing to build the 3D map model was performed by using the IRIS 3D printer that is a 3D color printer. In the 3D printing, the 337 pages of A4 size papers were used as the material to build the 3D map model. After finishing the 3D printing, the 3D map model inside the 337 pages of A4 size papers was removed out from the IRIS 3D printer. Finally, all unnecessary parts around the 3D map model were removed in order to obtain the 3D map model of the environment around the river.

4.4.1 Thickening 3D Textured Map

Firstly, a rectangular area was selected in the 3D textured map of the static photo images using a bounding box in the model pane of the Agisoft. Then, a new 3D map was built by using the selected area. After that, the new 3D map was exported to an OBJ file by using the Agisoft PhotoScan software. The OBJ file contains the 3D geometry data of the new 3D map, namely the 3D coordinates of each point in the new 3D map. Both a JPG file containing the texture data of the new 3D map as shown in Fig. 4.13 and an MTL file containing the parameters to map the texture data were generated by the Agisoft in order to keep the texture of the new 3D map. Then, the new 3D map in the OBJ file was thickened by using the Blender.



Fig. 4.13 Texture data of the new 3D map saved in the JPG file generated by the Agisoft

Figure 4.14 shows the new 3D map and the thickened 3D map built by using the selected area in the 3D textured map of static photo images of the environment around the river. The length, the width, and the height of the thickened 3D map are 21.5 [cm], 15.5 [cm], and 3.0 [cm], respectively. The scale ratio of the thickened 3D map to the real environment was 1:126.



(a) New 3D map



(b) Thickened 3D map

O-XYZ: Global coordinate frame

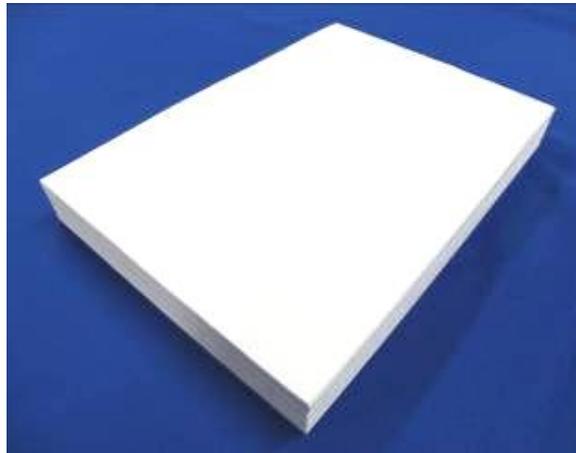
Fig. 4.14 Both new 3D map and thickened 3D map (scale ratio of 1:126) built using the selected area in the 3D textured map of static photo images of the environment around the river

4.4.2 Preparing Material and Color 3D Printer

Figure 4.15 shows both the IRIS 3D printer connected to a personal computer installing the SliceIT and the A4 size papers prepared to create the 3D map model of the environment around the river. A 3D color printer, here the IRIS 3D printer was used to build the 3D map model of the environment around the river. The IRIS 3D printer is connected to a personal computer installing the SliceIT (Ver.6.1.5) that is the IRIS 3D printer's software. Then, A4 size papers were used as the materials to build the 3D map model.



(a) IRIS 3D printer used to create the 3D map model



(b) A4 size papers used as material to create the 3D map model

Fig. 4.15 Both IRIS 3D printer connected to a personal computer installing the SliceIT and A4 size papers prepared to create the 3D map model of the environment around the river

Some processes were performed before starting the 3D printing. Firstly, the thickened 3D map was exported to a new OBJ file by using the Blender software. Both a new JPG file and a new MTL file were also generated by the Blender. Then, those three new files were inputted to the personal computer installing the SliceIT. The thickened 3D map with texture data was read by the SliceIT and displayed in the screen of the personal computer as shown in Fig. 4.16. Then, the thickened 3D map was sliced into 337 colored layers by the SliceIT. After that, those layers were printed on 337 pages of A4 size papers by using an ordinary color printer connected to the personal computer. The colors printed on each page of paper describe the geometry and the texture of each layer of the thickened 3D map. Furthermore, those 337 pages of colored papers and hundreds of clear papers were loaded into the paper tray of the IRIS 3D printer as shown in Fig. 4.17. Both a knife and a bottle of adhesive glue were set inside the IRIS 3D printer to build the 3D map model of the environment around the river using the 337 pages of colored papers.

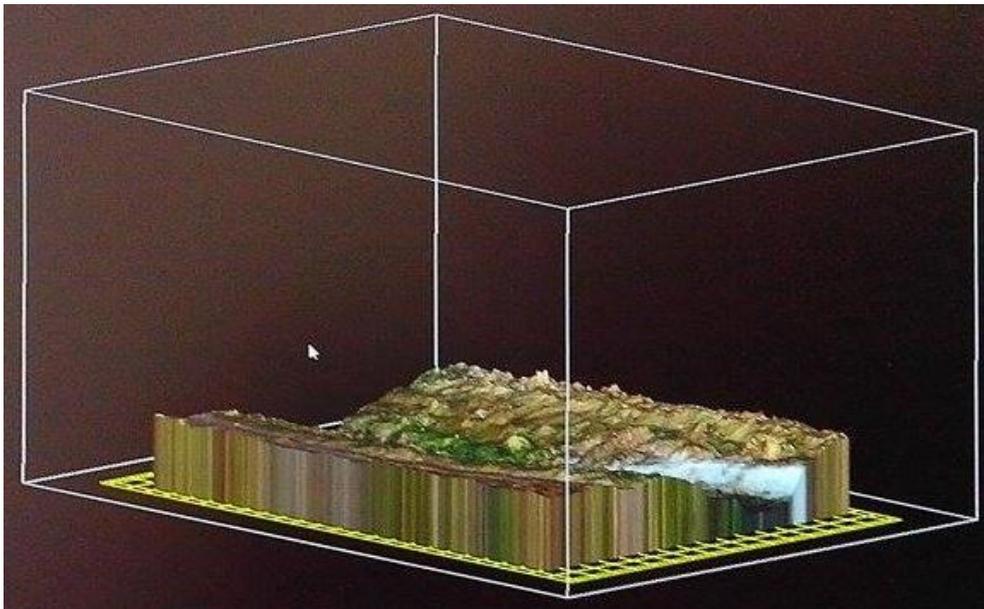


Fig. 4.16 Thickened 3D map with texture data displayed in the screen of the personal computer installing the SliceIT

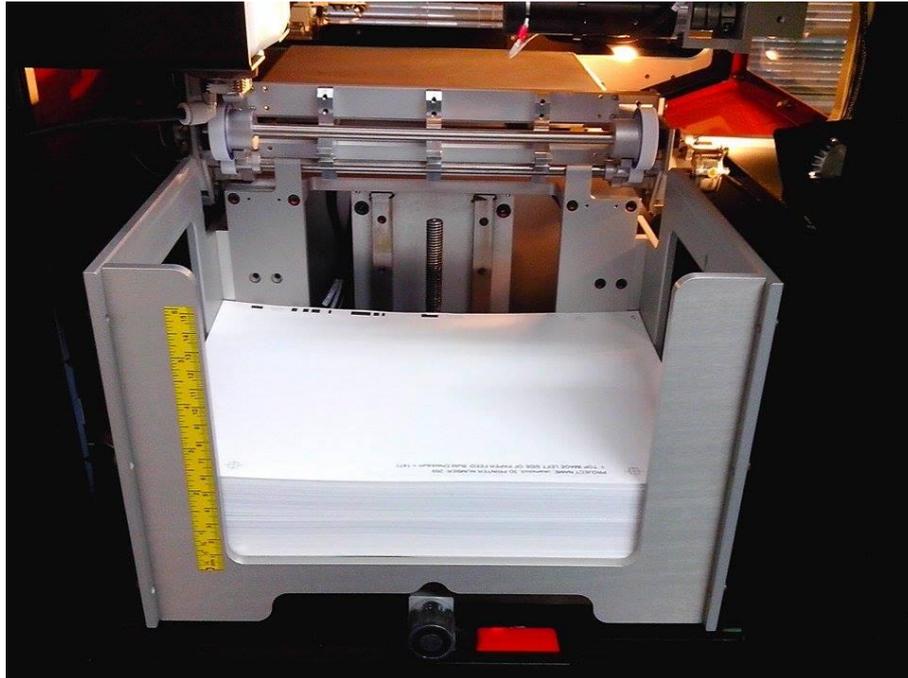


Fig. 4.17 The 337 pages of colored papers and hundreds of clear papers loaded into the paper tray of the IRIS 3D printer as the material to create the 3D map model

4.4.3 Printing 3D Map Model

The 3D printing was performed after finishing the preparation of the material and the IRIS 3D printer. Firstly, the first page of papers was pulled from the paper tray and placed on a build plate inside the IRIS 3D printer. Then, the geometry data of the first layer were read by the SliceIT. After that, the commands of both cutting and applying adhesive were generated by the SliceIT based on the geometry data. The commands were sent from the personal computer to the IRIS 3D printer. The processes of the cutting and applying adhesive were performed by the IRIS 3D printer on the first page of papers. After finishing the processes on the first page, the second page was pulled and placed on the first page. Both pages were pressed by the heat plate inside the IRIS 3D printer. Then, the whole processes were repeated from the second page to the 337th page of papers. After that, the 337 pages of papers containing the 3D map model were

taken out from the IRIS 3D printer. Figure 4.18 shows the 3D map model inside 337 pages of A4 size papers built by using the static photo images of the environment around the river.



Fig. 4.18 3D map model inside 337 pages of A4 size papers built using the static photo images of the environment around the river

4.4.4 Result

After finishing the 3D printing, all unnecessary parts around the 3D map model were removed from the 3D map model. Finally, the 3D map model of the environment around the river built by using the thickened 3D map and the IRIS 3D printer was obtained as shown in Fig. 4.19. The dimensions and the scale ratio of the 3D map model were same as those of the thickened 3D map. By using the 3D map model, the geometry and texture of the environment around the river can be seen clearly from all views. The top, front, left, and right views of the 3D map model of the environment around the river are shown in Fig. 4.20.

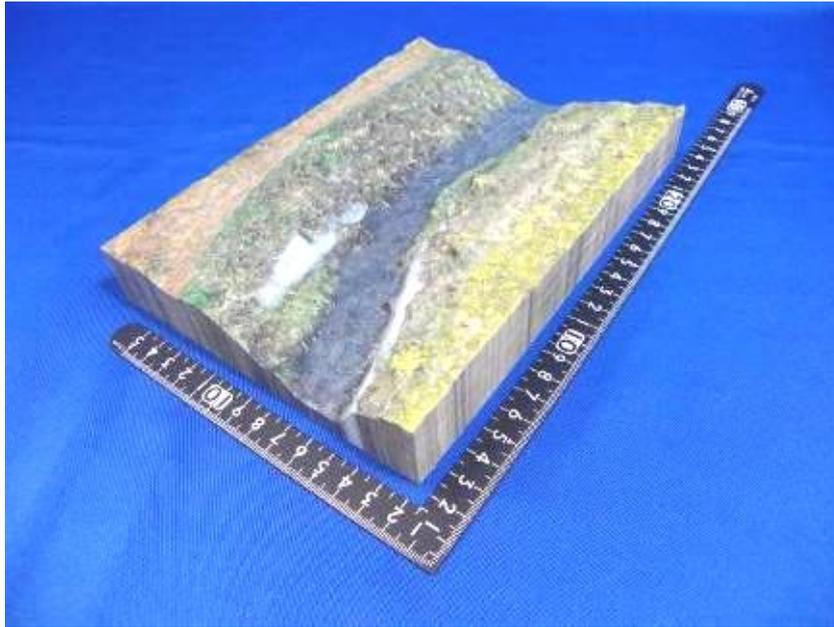


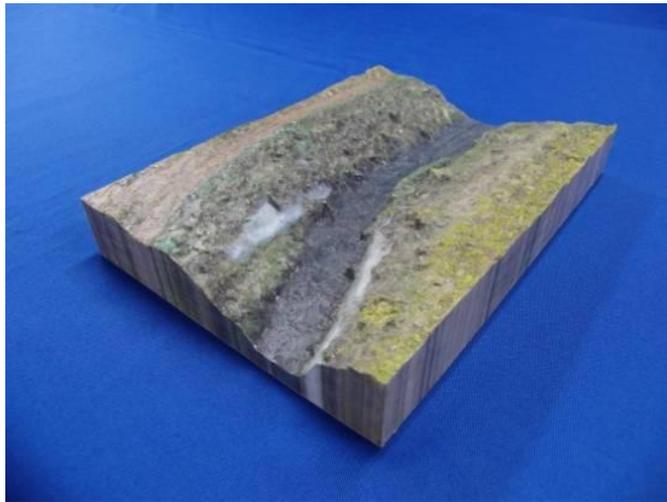
Fig. 4.19 Final 3D map model of the environment around the river built by using the thickened 3D map and the IRIS 3D printer



(a) top view



(b) front view



(c) left view



(d) right view

Fig. 4.20 Top, front, left, and right views of the 3D map model of the environment around the river

4.5 Conclusions

The summary of the results is shown below.

- (1) Firstly, the flying experiment in which a multi-rotor drone with a fisheye-lens camera captures overlapping static photo images and video images of an environment around a river had been carried out in this study.
- (2) Then, the 3D textured maps of the static photo images and video images had been created by using the Agisoft PhotoScan Professional. Furthermore, the thickened 3D map had been built by using the selected area in the 3D textured map of the static photo images.
- (3) Finally, the 3D map model of the environment around the river had been created by using the thickened 3D map and the IRIS 3D printer. The scale ratio of the 3D map model to the real environment is 1:126.

Chapter 5

Summary of Conclusions

The summary of the results of study on multi-rotor drone to fly autonomously along a river is shown below.

- (1) The algorithms to segment the river area from the river scenes and to perform an autonomous flight along a river had been developed in this study.
- (2) The flying experiments where a multi-rotor drone performs autonomous flights along a straight and a curved river using a single-lens camera and the image processing had been carried out. The experimental result shows that the multi-rotor drone could autonomously fly along the curved river for the distance of 83 [m].

The summary of the results of study on 3D map modeling of an environment around a river is shown below.

- (1) The 3D textured maps of the static photo images and video images had been created by using the Agisoft PhotoScan Professional. The thickened 3D map had been built by using the selected area in the 3D textured map of the static photo images.
- (2) The 3D map model of the environment around the river had been created by using the thickened 3D map and the IRIS 3D printer.

References

- [1] S. Scherer, J. Rehder, S. Achar, H. Cover, A. Chambers, S. Nuske & S. Singh, “River Mapping from a Flying Robot: State Estimation, River Detection, and Obstacle Mapping”, *Journal of Autonomous Robots*, Vol. 33 (1-2), 2012, pp. 189–214.
- [2] Chambers, S. Achar, S. Nuske, J. Rehder, B. Kitt, L. Chamberlain, J. Haines, S. Scherer & S. Singh, “Perception for a River Mapping Robot”, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 227–234.
- [3] S. Achar, B. Sankaran, S. Nuske, S. Scherer & S. Singh, “Self-Supervised Segmentation of River Scenes”, *IEEE International Conference on Robotics and Automation*, 2011, pp. 6227–6232.
- [4] S. Rathinam, P. Almeida, Z. Kim, S. Jackson, A. Tinka, W. Grossman & R. Sengupta, “Autonomous Searching and Tracking of a River Using an UAV”, *American Control Conference*, 2007, pp. 359–364.
- [5] S. Yang, S. Scherer & A. Zell, “An Onboard Monocular Vision System for Autonomous Takeoff, Hovering and Landing of a Micro Aerial Vehicle”, *Journal of Intell Robot Syst*, Vol. 69, 2012, pp. 499–515.
- [6] J. Chen & D. Dawson, “UAV Tracking with a Monocular Camera”, *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 3873–3878. <http://doi.org/10.1109/CDC.2006.377532>.
- [7] J. Roberts, T. Stirling, J.-C. Zufferey, & D. Floreano, “Quadrotor using minimal sensing for autonomous indoor flight”, *EMAV*, 2007.

- [8] M. Achtelik, S. Weiss & R. Siegwart, “Onboard IMU and Monocular Vision Based Control for MAVs in Unknown In and Outdoor Environments”, *Proceedings of International Conference on Robotics and Automation*, 2011.
- [9] M. Blosch, S. Weiss, D. Scaramuzza, & R. Siegwart, “Vision Based MAV Navigation in Unknown and Unstructured Environments”, *IEEE International Conference on Robotics and Automation*, 2010, pp. 21–28. <http://doi.org/10.1109/ROBOT.2010.5509920>.
- [10] F. Fraundorfer, L. Heng, D. Honegger, G.H. Lee, L. Meier, P. Tanskanen, M. Pollefeys, “Vision-Based Autonomous Mapping and Exploration Using a Quadrotor MAV”, *IROS*, 2012, pp. 4557–4564. <http://doi.org/10.1109/IROS.2012.6385934>.
- [11] Bachrach, S. Prentice, R. He, P. Henry, A. S. Huang, M. Krainin, D. Maturana, D. Fox, and N. Roy, “Estimation, Planning, and Mapping for Autonomous Flight Using an RGB-D Camera in GPS-Denied Environments”, *The International Journal of Robotics Research*, Vol. 31, no. 11, pp. 1320–1343, 2012.
- [12] F. Andert, F.M. Adolf, L. Goormann, & J.S. Ditttrich, “Autonomous Vision-Based Helicopter Flights Through Obstacle Gates”, *Journal of Intelligent & Robotic Systems*, Vol. 57(1–4), 2010, pp. 259–280.
- [13] F. Ruffier, and N. Franceschini, "Visually Guided Micro-Aerial Vehicle: Automatic Take Off, Terrain Following, Landing and Wind Reaction", *International Conference on Robotics and Automation*, 2004.
- [14] L. Meier, P. Tanskanen, F. Fraundorfer, & M. Pollefeys, “PIXHAWK: a System for Autonomous Flight Using Onboard Computer Vision”, *IEEE International Conference on Robotics and Automation*, 2011, pp. 2992–2997.

- [15] S. Scherer, S. Singh, L. Chamberlain, & M. Elgersma, “Flying Fast and Low Among Obstacles: Methodology and Experiments”, *The International Journal of Robotics Research*, Vol. 27(5), 2008, pp. 549–574.
- [16] S. Scherer, D. Ferguson, & S. Singh, “Efficient C-Space and Cost Function Updates in 3D for Unmanned Aerial Vehicles”, *IEEE International Conference On Robotics and Automation*, 2009, pp. 2049–2054.
- [17] Viquerat, L. Blackhall, A. Reid, & S. Sukkarieh, “Reactive Collision Avoidance for Unmanned Aerial Vehicles Using Doppler Radar”, *Proceedings of the International Conference on Field and Service Robotics*, 2007.
- [18] S. Weiss, M. Achtelik, L. Kneip, D. Scaramuzza, & R. Siegwart, “Intuitive 3D Maps for MAV Terrain Exploration and Obstacle Avoidance”, *Journal of Intelligent & Robotic Systems*, Vol. 61, 2011, pp. 473–493.
- [19] Wu, E. Johnson, and A. Proctor, “Vision-Aided Inertial Navigation for Flight Control”, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005, pp. 1–13.
- [20] Khemraj, J. Kumar, A. Srivastava & G. Srivastava. “Autonomous UAV (Unmanned Aerial Vehicle) for Navigation & Surveillance Purposes”, *Global Journal of Researches in Engineering Automotive Engineering*, Vol. 12, Issue 2, 2012.
- [21] E. Frazzoli, M. A. Dahleh, E. Feron, “A Hybrid Control Architecture for Aggressive Maneuvering of Autonomous Helicopters”, *Proceedings of the 38th Conference on Decision & Control Phoenix, Arizona USA, 1999*, pp. 2471-2476.

- [22] V. Gavrilets, I. Martinos, B. Mettler, & E. Feron, “Aggressive Maneuvering Flight Tests of a Miniature Robotic Helicopter”, *Proceeding of the 8th International Symposium on Experimental Robotics*, 2002.
- [23] S. Saripalli, J. Montgomery, & G. Sukhatme, “Vision-Based Autonomous Landing of an Unmanned Aerial Vehicle”, *Proceedings of International Conference on Robotics and Automation*, 2002.
- [24] Shakernia, R. Vidal, C. Sharp, Y. Ma, & S. Sastry, “Multiple View Motion Estimation and Control for Landing an Unmanned Aerial Vehicle,” *Proceedings of International Conference on Robotics and Automation*, 2002.
- [25] J. Woo, K. Son, T. Li, G. Kim, I.S. Kweon, “Vision-Based UAV Navigation in Mountain Area”, *IAPR Conference on Machine Vision Applications*, 2007.
- [26] S. First, E.D. Dickmanns, “A Vision Based Navigation System for Autonomous Aircraft”, *Robotics and Autonomous Systems*, Vol. 8, 1999, pp. 173-184. Elsevier Science B.V. PII: S0921-8890(99)00015-9.
- [27] P. Baker, B.K. Parsi, “Using Shorelines for Autonomous Air Vehicle Guidance”, *Journal of Computer Vision and Image Understanding*, Vol. 114, 2010, pp. 723–729. Elsevier Inc. doi:10.1016/j.cviu.2010.01.009.
- [28] J. Courbon, Y. Mezouar, N. Gue´nard, P. Martinet, “Vision-Based Navigation of Unmanned Aerial Vehicles, *Journal of Control Engineering Practice*, Vol. 18 , 2010, pp. 789–799. Elsevier Ltd. doi:10.1016/j.conengprac.2010.03.004.
- [29] E. Frew, T. McGee, Z. Kim, X. Xiao, S. Jackson, M. Morimoto, S. Rathinam, J. Padial, & R. Sengupta, “Vision-Based Road Following Using a Small Automonous Aircraft”, *Proc. IEEE Aerospace Conference*, 2004.

- [30] S. Rathinam, Z. Kim, A. Soghikian, & R. Sengupta, "Vision Based Following of Locally Linear Structures Using an Unmanned Aerial Vehicle", *44th IEEE Conference on Decision and Control and European Control Conference*, 2005.
- [31] K. Celik, S.J. Chung, M. Clausman, & A.K. Somani, "Monocular Vision SLAM for Indoor Aerial Vehicles", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1566-1573.
- [32] M. Jun, I. Stergios, I. Roumeliotis, & G.S. Sukhatme. "State Estimation of an Autonomous Helicopter using Kalman Filtering", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1999, pp. 1346-1353.
- [33] S. Saripalli, S.G. Sukhatme, & J.F. Montgomery, "An Experimental Study of the Autonomous Helicopter Landing Problem", *Proceedings of International Symposium on Experimental Robotics (ISER)*, 2002.
- [34] S. Scherer, L. Chamberlain, S. Singh, "Autonomous Landing at Unprepared Sites by a Full-Scale Helicopter", *Journal of Robotics and Autonomous Systems*, Elsevier B.V. doi:10.1016/j.robot.2012.09.004.
- [35] P.A. Pabbeel, A. Coates , Y.N. Andrew, "Autonomous Helicopter Aerobatics through Apprenticeship Learning", *International Journal of Robotics Research*, 2010.
- [36] S. Scherer, J. Rehder, S. Achar, H. Cover, A. Chambers, S. Nuske & S. Singh, "River Mapping from a Flying Robot: State Estimation, River Detection, and Obstacle Mapping", *Journal of Autonomous Robots*, Vol. 33 (1-2), 2012, pp. 189–214.

- [37] Chambers, S. Achar, S. Nuske, J. Rehder, B. Kitt, L. Chamberlain, J. Haines, S. Scherer & S. Singh, “Perception for a River Mapping Robot”, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 227–234.
- [38] M. Kedzierski & A. Fryskowska, “Application of Digital Camera with Fisheye-lens in Close Range Photogrammetry”, *ASPRS Annual Conference*, 2009.
- [39] M. Kedzierski M & P. Walezykowski, “Fisheye-lens Camera System Application to Cultural Heritage Data Acquisition”, *XXI International CIPA Symposium*, 2007.
- [40] H. Zhao & J. K. Aggarwal, “3D Reconstruction of an Urban Scene from Synthetic Fish-Eye Images - Image Analysis and Interpretation”, *Proceedings of the 4th IEEE Southwest Symposium*, 2000, pp. 219-223, doi: 10.1109/IAI.2000.839603.
- [41] F. Remondino, L. Barazzetti, F. Nex, M. Scaioni & D. Sarazzi D, “UAV Photogrammetry for Mapping and 3D Modeling – Current Status and Future Perspectives”, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVIII-1, 2011, pp. 25-31.
- [42] L. Barazzetti & M. Scaioni, “Orientation and 3D Modeling from Markerless Terrestrial Images: Combining Accuracy with Automation”, *The Photogrammetric Record* 25(132), 2010, pp. 356–381.
- [43] L. Zongjian, “UAV for Mapping—Low Altitude Photogrammetric Survey”, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVII, Part B1, 2008, pp. 1183-1186.
- [44] C. Zhang, “An UAV-Based Photogrammetric Mapping System for Road Condition Assessment”, *International Archives of the Photogrammetry, Remote*

Sensing and Spatial Information Sciences, Vol. XXXVII, Part B5, 2008, pp. 627-631.

- [45] H. Eisenbeiss, "A Mini Unmanned Aerial Vehicle (UAV): System Overview and Image Acquisition", *Proceeding of International Workshop on Processing and Visualization Using High-Resolution Imagery*, 2004.
- [46] P. Gong, Y. Sheng, & G.S. Blging, "3D Model-Based Tree Measurement from High-Resolution Aerial Imagery", *Photogrammetric Engineering & Remote Sensing*, Vol. 68, No. 11, 2002, pp. 1203-1212.
- [47] C. Frueh & A. Zakhor, "3D Model Generation for Cities Using Aerial Photographs and Ground Level Laser Scans", *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2.2, 2001, pp. 31-38.
- [48] P. L. Falkingham, "Acquisition of High Resolution Three-Dimensional Models Using Free, Open-Source, Photogrammetric Software", *Palaeontologia Electronica*, Vol. 15, Issue 1, 2012.
- [49] T. Ishida, "Activities and Technologies in Digital City Kyoto", *Digital Cities 2003, LNCS 3081*, 2005, pp. 166-187.
- [50] H. Eisenbeiss, "The Autonomous Mini Helicopter: a Powerful Platform for Mobile Mapping", *IAPRS&SIS*, Vol. 37(B1), 2008, pp. 977-983.
- [51] J. Everaerts, "The Use of Unmanned Aerial Vehicles (UAVs) for Remote Sensing and Mapping", *IAPRS&SIS*, Vol. 37(B1), 2008, pp. 1187-1192.
- [52] H. Eisenbeiss, "The Potential of Unmanned Aerial Vehicles for Mapping", *Dieter Fritsch (Ed.)*, 2011, pp. 135-145.

- [53] H. Pueschel, M. Sauerbier, & H. Eisenbeiss, “A 3D Model of Castle Landenberg (CH) from Combined Photogrammetric Processing of Terrestrial and UAV-Based Images. *IAPRS&SIS*, Vol. 37(B6), pp. 96-98.
- [54] G. Guidi, F. Remondino, M. Russo, F. Menna, A. Rizzi, & S. Ercoli, “A Multi-Resolution Methodology for the 3D Modeling of Large and Complex Archeological Areas”, *International Journal of Architectural Computing*, Vol. 7, Issue 1, 2009, pp. 39-55.
- [55] T. Ohdake, & H. Chikatsu, “3D Modeling of High Relief Sculpture Using Image-Based Integrated Measurement System”, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5/W17).
- [56] L. Pe´nard, N. Paparoditis, and M. Pierrot-Deseilligny, “3D Building Facade Reconstruction Under Mesh Form from Multiple Wide Angle Views”, *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5/W17).
- [57] L. Barazzetti, F. Remondino, & M. Scaioni, “Automation in 3D Reconstruction: Results on Different Kinds of Close-Range Blocks”, *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVIII, 2010, pp. 55-61.
- [58] G. Vosselman & S. Dijkman, “3D Building Model Reconstruction From Point Clouds And Ground Plans”, *International Archives of Photogrammetry and Remote Sensing*, Vol. XXXIV-3/W4, 2001.
- [59] A. Nuechter, K. Lingemann, J. Hertzberg, & H. Surmann, “6D SLAM for 3D Mapping Outdoor Environments”, *Journal of Field Robotics (JFR) - Special Issue*

on Quantitative Performance Evaluation of Robotic and Intelligent Systems, Vol. 24(8-9), 2007, pp. 699-722.

- [60] Rafael Valencia, Ernesto H. Teniente, Eduard Trulls, and Juan Andrade-Cetto, “3D Mapping for Urban Service Robots”, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 3076-3081.
- [61] V. Sequeira, G. M. Joao, G. Alves, & M. I. Ribeiro, “3D Environment Modelling Using Laser Range Sensing”, *Robotics and Autonomous Systems*, Vol. 16, 1995, pp. 81-91.
- [62] D. Hahnel, W. Burgard, S. Thrun, “Learning Compact 3D Models of Indoor and Outdoor Environments with a Mobile Robot”, *Journal of Robotics and Autonomous Systems*, Vol. 44, 2003, pp. 15-27.
- [63] H. Surmann, A. Nuchter, & J. Hertzberg, “An Autonomous Mobile Robot with A 3D Laser Range Finder for 3D Exploration and Digitalization of Indoor Environments”, *Journal of Robotics and Autonomous Systems*, Vol. 45, Issue 3, 2003, pp. 181-198.
- [64] S. Thrun, D. Fox, & W. Burgard, “A Real-Time Algorithm for Mobile Robot Mapping with Application to Multi Robot and 3D Mapping”, *Proceedings of the IEEE International Conference on Robotics and Automation*, 2000.
- [65] G. Romanescu, V. Cotiuga, A. Asandulesei, & C. Stoleriu, “Use of the 3D Scanner in Mapping and Monitoring the Dynamic Degradation of Soils: Case Study of the Cucuteni-Baiceni Gully on the Moldavian Plateau (Romania), *Hydrol. Earth Syst. Sci.*, Vol. 16, 2012, pp. 953–966, doi:10.5194/hess-16-953-2012.

- [66] S. Syed, P. Dare, & S. Jones, "Semi-Automatic 3D Building Model Generation From Lidar and High Resolution Imagery", *Proceedings of SSC2005 Spatial Intelligence, Innovation and Praxis*, 2005.
- [67] J. M. Cuschieri, "Three-Dimensional Map Generation from Side-Scan Sonar Images", *Journal of Energy Resources Technology*, Vol. 112, 1990, pp. 96-102.
- [68] M. Olmo, V. Ortega, A. Banos, C. Guirad, V. Regi, D. Arcos, & A. Baeza, "In-Vivo Behavior of Si-Hydroxyapatite/Polycaprolactone/DMB Scaffolds Fabricated by 3D Printing", *Journal of Biomed Mater Res, Part A*, 2012:00A.
- [69] M.C. Metzger, B.H. Majert, et. al. "Manufacturing Splints for Orthognathic Surgery Using a Three-Dimensional Printer", *Oral Surg Oral Med Oral Pathol Oral Radiol Endod*, Vol. 105, No. 2, 2008, pp. 1-5.
- [70] C.X.F. Lam, X.M. Mo, S.H. Teoh, & D.W. Hutmacher, "Scaffold Development Using 3D Printing with a Starch-Based Polymer", *Materials Science and Engineering*, Vol. 20, 2002, pp. 49–56.

List of Papers and Awards

International Journals (Full papers with reviews)

1. “Multi-Rotor Drone to Fly Autonomously along a River Using a Single-Lens Camera and Image Processing”, A. Taufik, S. Okamoto, J. H. Lee, *International Journal of Mechanical Engineering (IJME)*, Vol. 4, No. 6, pp. 39-50, 2015.
2. “3D Map Modeling of Environment around a River Using Static Photo Images and Video Images Captured by a Fisheye-Lens Camera Mounted on a Multi-Rotor Drone”, A. Taufik, S. Okamoto, J. H. Lee, *International Journal of Mechanical and Mechatronics Engineering IJMME-IJENS*, Vol. 16, No. 3, pp. 144-149, 2016.

Proceeding of International Conferences (Full papers with reviews)

1. “Multi-rotor Drone with Single-Lens Camera that Can Autonomously Fly along a River”, A. Taufik, S. Okamoto, J. H. Lee, *Conference Proceeding of APCEAS (Asia-Pacific Conference on Engineering and Applied Sciences)*, pp. 195-208, Osaka, Japan, August 2015.
2. “3D Mapping of an Environment around a River Using Image Processing of Photo Images Captured by a Multi-Rotor Drone and 3D Map Modeling Using a 3D Printer”, A. Taufik, S. Okamoto, J. H. Lee, *Conference Proceeding of ECBA-2016 (Engineering & Technology, Computer, Basic & Applied Sciences 2016)*, Vol. 127, No. 5, pp. 1-9, Seoul, South Korea, April 2016.

Proceeding of a Domestic Conference

1. “Image Segmentation for Autonomous Rotorcraft Flying along a River”, A. Taufik, S. Okamoto, J. H. Lee, *Conference Proceeding of Shikoku Branch Society of Instrument and Control Engineers 2014*, pp. 92-94, Matsuyama, Japan, November 2014.

Awards

1. **Best Paper Award:** “Multi-Rotor Drone to Fly Autonomously along a River Using a Single-Lens Camera and Image Processing”, A. Taufik, S. Okamoto, J. H. Lee, *International Journal of Mechanical Engineering (IJME)*, Vol. 4, No. 6, pp. 39-50, November 2015.
2. **Best Presentation Award:** “3D Mapping of an Environment around a River Using Image Processing of Photo Images Captured by a Multi-Rotor Drone and 3D Map Modeling Using a 3D Printer”, A. Taufik, S. Okamoto, J. H. Lee, *International Conference on Society of ECBA (Engineering & Technology, Computer, Basic & Applied Sciences)*, Seoul, South Korea, April 2016.

Acknowledgments

Be grateful and all glory to the Almighty God ALLAH SWT for having given a great opportunity to carry out this research. Salawat is addressed to Rasulullah SAW.

Expressions of gratitude are deservedly given to Directorate General of Higher Education of Indonesia, the State Polytechnic of Ujung Pandang and Ehime University for having given the opportunity to be a doctoral student in Robotics Laboratory of Ehime University.

Great and high appreciations are specially addressed to Prof. Shingo Okamoto for his guidances and support as my main supervisor for three years in Robotics Laboratory of Ehime University. Great and high appreciations are also addressed to Associate Prof. Jae Hoon Lee for his guidances and support as my co-supervisor.

Big appreciation and gratitude are addressed to my parents, my wife and my children for their endless support, care, and prayers. Then, many thanks are addressed to officers of the factory of Ehime University for their help in the 3D modeling of my object research. Many thanks are also addressed to Mr. Hidayat, Mr. Imamura and Dr. Abdul Kadir for their help and support when I was carrying out my flying experiments. Finally, many thanks are expressed to all members of Robotics Laboratory of Ehime University and my friends who had supported me during the period of this study.

Curriculum Vitae

Name : Akhmad Taufik

Place/Date of Birth : Wonomulyo, Indonesia / April 13, 1976

Gender : Male

Background of Education

1. Master Degree : Electrical Energy Engineering,
Hasanuddin University, Indonesia (2005-2008)
2. Bachelor Degree : Electrical Energy Engineering,
Hasanuddin University, Indonesia (1994-1999)

Occupation : Lecturer

Institution : State Polytechnic of Ujung Pandang, Indonesia

Department : Mechanical Engineering

Study Program : Mechatronics

Personal Email Address : ataufik_e94@yahoo.com