AUTONOMOUS FLIGHT OF UNMANNED MULTICOPTER USING SUPER WIDE-ANGLE STEREO VISION SYSTEM

Tetsuo YAMASHITA*, Yasuhiro KAJIWARA*,
Christopher Thomas RAABE, Takeshi TSUCHIYA**, Shinji SUZUKI****
***Ricoh Co. Ltd, ****the University of Tokyo**

Keywords: *UAV, Visual Odometry, Stereo Camera, Positioning, Obstacle Avoidance*

Abstract

We present an automatic flight system for small unmanned aerial vehicles (UAV, drone) using a super wide-angle stereo camera. Precise real-time positioning method is required to make the drone move more smoothly. An optimal method that can find obstacles in the surrounding areas of the drone is also necessary. We propose a super wide-angle stereo vision system. We used both hardware and software to process positioning task and 3d mapping task in real-time. Its field of view is 120 degrees and obstacles can be detected in the front, up, down, left and right with only one device. The autonomous flight system, which we integrated into a multi-copter, completed an indoor waypoints navigation scenario and an obstacle avoidance scenario.

1 Introduction

Small unmanned aerial vehicles are beginning to be applied to various fields such as surface inspection of structures including buildings, bridges, mapping, field monitoring. For example, by using a drone for the surface inspection, it is possible to realize a surface inspection without a high place work or a scaffolding installation even with a huge structure. Furthermore, the autonomous drone has potential to improve its efficiency more. However, there are difficulties in carrying out the surface inspection task with autonomous drones. In a part of the designed flight path, it is difficult to receive GPS signals and there sometimes are obstacles such as trees or weeds. So non-GPS-based localization technique and sensing and avoidance (SAA) techniques are strongly required.

To realize the self-localization and obstacle avoidance, methods using Lidar, RGB-D Camera and monocular cameras have been proposed[1][2][5][10][12][13][15][14][16].

Lidar has been used for location estimation and map generation[1]. However, since Lidar has a narrow field of view in the vertical direction, it is difficult to find obstacles above and below of the drones on a standalone basis, so it is required to other sensors to detect obstacles such as a wide field of view monocular camera [10].

RGB-D cameras have also been also used for localization and mapping [6]. The RGB-D camera is a camera that can capture a color image and a depth image which has depth values for each pixel at same time. Since the resolution of the depth image is high, position estimation and 3D map generation can be performed with high density and accuracy. However, sunlight has a bad influence on the depth images and it is difficult to use in the outdoor. It is not suited for some application such as surface inspection and so on.

In the method of using monocular cameras, Simultaneous Localization and Mapping (SLAM) and Visual Odometry (VO) technique have been applied to drone's automatic flight [2][5]. However, the monocular SLAM technique cannot measure the real-scale self-position and detect obstacle position. To estimate self-position in real-scale, some other sensors or equipment such as IMU, AR markers are required. In the studies to avoid obstacles with monocular cameras, a method to measure the relative distance to obstacles using the optical flow of the image was presented [11]. However, it is difficult to apply the existing path planning

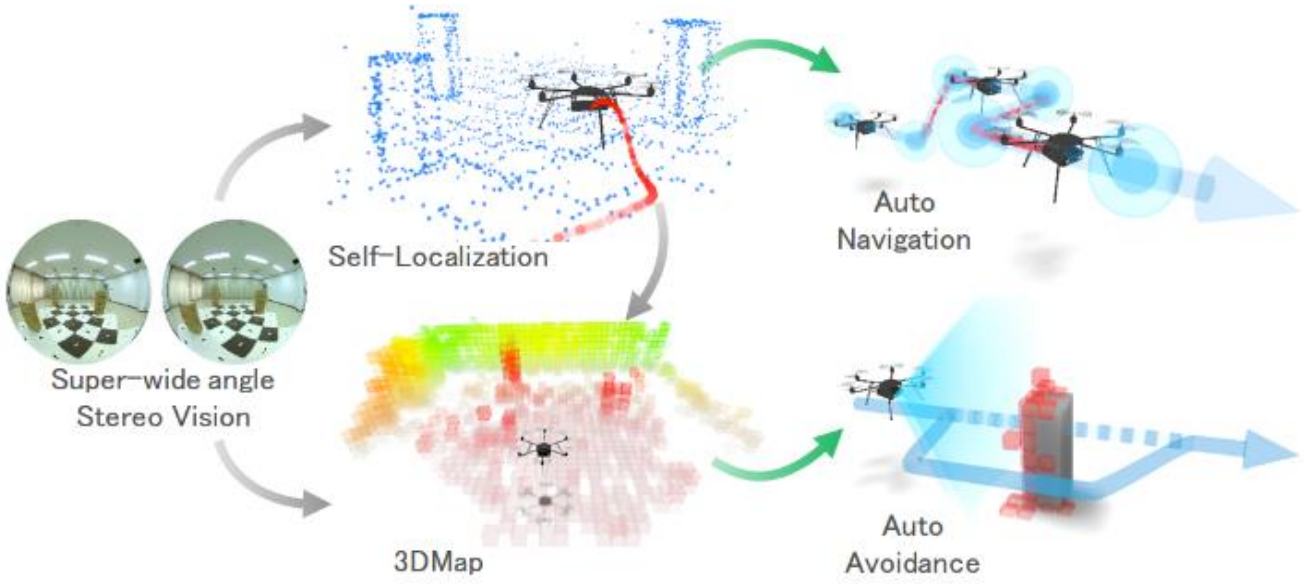

Fig. 1 Overview of 3D Vision System Processing Pipe Line for Auto Navigation and Collision Avoidance

algorithm such as the Dijkstra method, because the position of the obstacle in the map is unknown.

Stereo cameras are also used in many work [12][13][14][16]. A stereo camera is a camera that can capture color images and depth images in the same way as RGB-D cameras. Compared to RGB-D cameras, stereo cameras have an advantage that the depth images are robust in the outdoors. Also, the stereo camera can provide the self-position and obstacle position on the real-size. The self-position can be measured by using the SLAM / VO framework for stereo cameras and the obstacle position can be measured by using the depth image [16].

In drone, it is difficult to load rich computing resources because there are limitations on load weight, capacity, and power.

Fig. 2 Super wide angle stereo vision system

On the other hand, it is known that the delay and the rate of the position estimation adversely affect the speed control and the position control of the drones [15]. Therefore, some methods have been proposed to process these tasks in real time. For example, a method of using onboard system and ground stations [5], a method of simplifying part of processing such as lowering image resolution [2], a method of using a hardware for speeding up the processing of VO have been proposed [14][16]. Method using ground stations, it is necessary to perform communication between the ground station and the drones, limits the flight range in order to ensure communication quality and low delay. And Simplified process can degrade accuracy, and precise automatic flight may not be possible as a result. Schmid et al. [14] speed up the processing using HW, but the delay of VO is 250 milliseconds, and the rate is 15 Hz. It is still slow compared with IMU, and the image resolution that can be handled is low. A method capable of handling higher resolution images at a high speed is required to fly stably and smoothly.

Also, some method to expand the range in which obstacles around the drone can be detected have been proposed. To find obstacles around the entire drone, a method of arranging multiple camera on a round rig has been proposed [5] [6][17]. Considering the limitation of loading

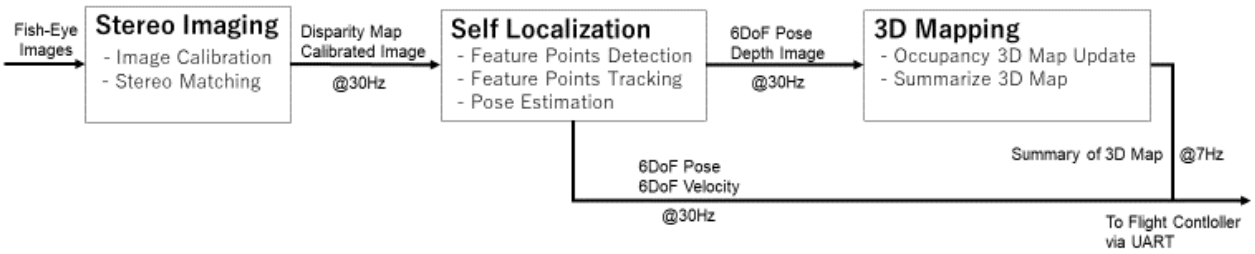

Fig. 3 Processing Pipeline of Super Wide-Angle Stereo Vision System

weight and capacity of the drone, it is desirable to reduce reducing the number of cameras.

In the application to surface inspection, it is required to fly both of indoor and outdoor environment. And it is required fly by avoiding obstacles including moving objects. To realize this, a stereo camera is suitable because it can capture depth images indoors and outdoors and can also be used for detecting obstacles.

We developed a super wide-angle stereo vision system that can perform position estimation and obstacle map generation with high speed and low delay using a super wide-angle stereo camera. The drone which mounts the system can follow a flight path and sense and avoid the obstacle using information from vision system (Fig.1). In the vision system, a part of the position estimation pipeline is processed with HW. It can handle high resolution images (960 x 960 pixels) in real time. The position estimation and obstacle map generation results can be output at 30 Hz. Its delay is about 100 milliseconds. Moreover, by using super wide-angle lens, we can detect the obstacle which is up and down, left and right with one stereo camera. In this paper, we report the super wide-angle stereo camera system and the result of automatic flight with it.

2 Super Wide-Angle Stereo Vision System

We developed a super wide-angle stereo vision system for automatic flight and automatic obstacle avoidance of drone (Fig. 2). The super wide-angle stereo vision system is equipped with two fisheye cameras, and it estimates the self-position and velocity of 6 degrees of freedom and makes 3D obstacle map using images taken with two fish-eye cameras. Its feature is to process these processes at high speed with onboard computing resources. This feature was realized by processing part of the pipeline of position estimation processing with HW. It is introduced in the following section.

2.1 Stereo Imaging with Super Wide-Angle Cameras

The stereo imaging process is a process to generate a calibrated image and a disparity image from the images of the left and right fish-eye cameras. This process includes calibration task and stereo matching task. The stereo camera developed is a kind of parallel stereo camera. The left and right cameras are equipped with the same lens, attached in the same direction, and capture triggers are electrically synchronized. However, it is difficult to make ideal parallel stereo cameras due to factors such as individual differences of the left and right lenses and assembly error of the camera. Therefore, calibration task is required before stereo matching task. These errors are calibrated with respect to the captured image, convert them into images taken with an ideal parallel stereo camera. A disparity image is an image in which each pixel has depth information. Calibration task and disparity calculation task require large computational effort, but in this system, they are processed in real time using HW. The output of this system is images obtained by cutting out an area of 120 degrees top and bottom and left and right to handle easily in the position estimation processing.

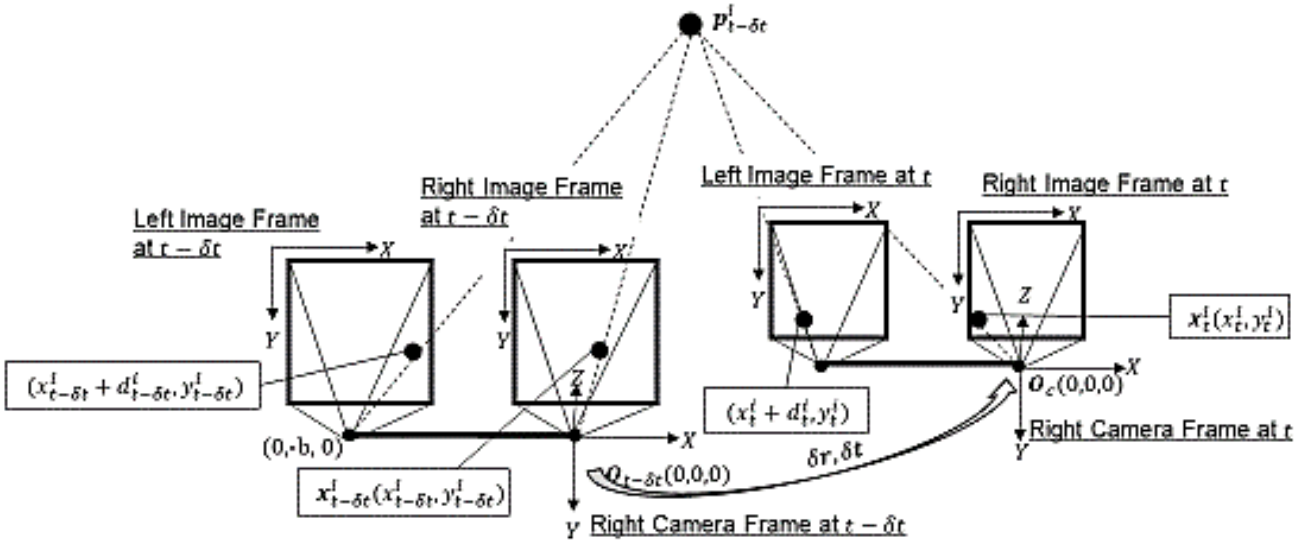

Fig. 4 Relative pose estimation using feature points

2.2 Self-Localization

Self-localization is done after stereo processing. We have developed a SW and HW co-designed system for self-localization. It can handle a feature point-based VO pipeline for stereo camera. VO pipeline consists of feature extraction task and feature tracking task and pose estimation task. Feature extraction task is a task to extract feature points from the image. Feature tracking task is a task to search for the same feature point between sequential frames. Pose estimation task is a task to estimate the movement of the drone by using information on feature points of the previous frame and feature points of the current frame. HW processes feature extraction and tracking task. SW processes pose estimation task (Fig. 3).

The first task of the VO pipeline is feature point extraction. The feature point extraction is a process of searching for a local region that can be distinguished from other local regions. There are various feature point extraction algorithms, in our system Harris operator[18] was adopted as a feature point extraction algorithm and this algorithm was implemented in HW. This HW can extract up to 900 feature points from 960 x 960 images. It can extract feature points 30 times in a second and its delay is several milliseconds.

We also developed feature points tracking HW. Its tracking algorithm is pyramidal Kanade-Lucas-Tomashi feature tracker (KLT) [19]. It can track up to 3200 feature points at a time and process 30 frames in a second. There are two types feature points tracking algorithm. One is a method using feature descriptor such as ORB. Another is a method using optical flow like KLT. The former is adopted in many SLAM frameworks. However, we use the latter. In this system, the field of view of the camera is wide and the lag-time of frames is small. It can track feature points sufficiently even with the optical flow based method.

Pose estimation task is processed after feature point tracking task (Fig. 4). The pose estimation task is a task to estimate the relative pose of the drone from previous frame to the current frame and estimate global pose. The pose is six degrees of freedom (3 degrees of freedom translation and 3 degrees of freedom rotation).

The relative pose can be obtained from the three-dimensional position of the feature point in the previous frame and the image position in the current frame (Fig. 5). The three-dimensional position $\mathbf{p}_i$ in a frame coordinate is defined in Equation (1). In $\mathbf{x}_i(x_i, y_i)$ is the feature point position in the image, d_i is the disparity value of the feature point, f is the focal length of the camera, b is the distance between left and right cameras and c_x and c_y are the camera center.

$$\boldsymbol{p}_i = \mathrm{b}/d_i \begin{bmatrix} x_i - c_x \\ y_i - c_y \\ f\ (x_i - c_x) \end{bmatrix} \quad (1)$$

Relative pose is presented by 3 DoF rotation parameters and 3 DoF translation parameters. The parameters are obtained by minimizing the sum of the re-projection errors from the right camera and the left camera. The reprojection error of ith point from the right camera $\mathrm{e}_\mathrm{i}^\mathrm{r}(\mathbf{r},\mathbf{t})$is defined by equation (2), and the re-projection error of ith point from the left camera $\mathrm{e}_\mathrm{i}^\mathrm{l}(\mathbf{r},\mathbf{t})$ is defined by equation (3).

$$\mathrm{e}_\mathrm{i}^\mathrm{r}(\mathbf{r},\mathbf{t}) = \left\| \boldsymbol{x}_t^i - K\left(\boldsymbol{q}\left(\boldsymbol{p}_{t-\delta t}^i, \boldsymbol{r}, \boldsymbol{t}\right)\right)\right\| \quad (2)$$

$$\mathrm{e}_\mathrm{i}^\mathrm{l}(\mathbf{r},\mathbf{t}) = \left\| \boldsymbol{x}_t^i + \boldsymbol{d}_t^i - K(\boldsymbol{q}(\boldsymbol{p}_{t-\delta t}^i, \boldsymbol{r}, \boldsymbol{t}) + \boldsymbol{b})\right\| \quad (3)$$

$\boldsymbol{x}_t^i$ is K is the projection function of the camera defined by equation (4), and projects the 3D point to the image plane. The two-dimensional point obtained by K is the position of the feature point in the pixel coordinate.

$$K(\boldsymbol{p}) = \begin{bmatrix} f_x p_{px}/p_{pz} + c_x \\ f_y p_{py}/p_{pz} + c_y \end{bmatrix} \quad (4)$$

Also $\boldsymbol{q}$ is a function to transform the 3d point with rotation parameters $\mathbf{r}$ and $\mathbf{t}$. It is defined as equation (5). $\boldsymbol{R}$ is a function to convert rotation parameters into rotation matrix. We parameterized the rotation by using Rodrigues' formula [20].

$$\boldsymbol{q}(\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{t}) = \boldsymbol{R}(\boldsymbol{r})\boldsymbol{p} + \boldsymbol{t} \quad (5)$$

In equation (2), the position of feature point converts to 3d space in the previous frame and it is projected onto the right camera coordinate system. The error is defined as the distance between the position of the re-projected point and actually measured point. Equation (3) is a re-projection error of the left camera. It is almost the same as right one, but the distance of the baseline and the disparity value are used to calculate position of points in the left camera image.

We solve equation (6) to obtain the parameter r and t. The parameters are obtained by minimizing the sum of the reprojection errors of each feature point.

$$[\mathbf{r},\mathbf{t}] = \arg\min_{[\boldsymbol{r}\ ,\ \boldsymbol{t}]} \sum_{i=1}^{N} (e_i^r(\boldsymbol{r},\boldsymbol{t}) + e_i^l(\boldsymbol{r},\boldsymbol{t})) \quad (6)$$

This minimization problem can be solved using a nonlinear minimization method if there are at least three feature points. We solved this problem by Levenberg-Marquardt method.

Also, in addition to this, we use RANSAC to estimate pose robustly. Feature points and disparity values include outliers. They are due to such as stereo matching, feature point tracking, moving objects. In order to reduce these effects, the pose estimation was processed by the following steps.

1. Sample three points randomly from all tracked feature points
2. Estimate a relative pose with extracted points.
3. Count up inlier feature points by using estimated relative pose. Inlier feature points are points which have small reprojection error.
4. Compare the number of inliers with the number of inliers obtained at the best pose. If the number of inliers is larger than the number of inliers of best pose, update r and t.
5. Repeat step 1 to step 4 N times.
6. Refine the pose using only inlier points

After estimating relative pose, we use the equation (7) to estimate the pose in global frame. The origin of global frame is the first pose of the camera. The pose at time t in global frame is obtained by multiplying relative pose with pose at time $\mathrm{t} - \delta\,\mathrm{t}$ in global frame.

$$\begin{aligned} \boldsymbol{R}(\boldsymbol{r}_t) &= \boldsymbol{R}(\delta\boldsymbol{r})\boldsymbol{R}(\boldsymbol{r}_{t-\delta t}) \\ \boldsymbol{t}_t &= \boldsymbol{R}(\delta\boldsymbol{r})\boldsymbol{t}_{t-\delta t} + \delta\boldsymbol{t} \end{aligned} \quad (7)$$

Pose estimation task was implemented in SW. This process has less computational effort than feature point extraction and feature point tracking. It was able to process in real time even with software. The VO pipeline we developed achieved to process at 30 time in a second and its delay is about 100 msec.

Fig. 5 Japan Drone 2017 Flight Demo (left: scenario1, right: scenario 2)

2.3 3D Mapping

The 3D mapping is a task after VO pipeline. 3D mapping task consists of a task to update the 3D map and extract obstacles from the map. 3D map is updated using the distance image and the current global pose. In this system the map is 3D occupancy grid map. The 3D occupancy grid map is a map whose space is divided into cells, and each cell has the probability of occupation. We used OctoMap[10] to handle 3D occupancy grip map. In order to reduce computational effort, we set the grid size to 30 cm, making the map update range only within the range of 5m from the drone and distance images were down-sampled to 60 x 60 pixels. After 3d mapping task, nearby obstacles in the map are extracted and tell the positions of obstacles to flight controller.

3 Autonomous Flight System

We jointly developed a system for autonomous flight without GPS and auto obstacle avoidance using the vision system. The vision system mounted on a drone looking downward at an angle of 30 degrees. Its field of view covers downwards and front area. The vision system communicates with a flight controller over UART. The drone is MikroKopter Hexa and its flight controller board is a FlighCtrl_V2.0 and its software is customized to use our vision system. The vision system sends two packets to the flight controller. One packet includes current position, attitude and velocity of the drone. Another packet includes summarized 3D map information. The flight controller can autonomously control to flight along a preset route and avoid some obstacles using data from the vision system. A preset route consists of some points on the route (waypoints).

3.1 Result of Autonomous Flight and Avoidance on a Demonstration

We demonstrated two scenarios about autonomous flight and avoidance on “Japan Drone 2017” that was held from March 23th to

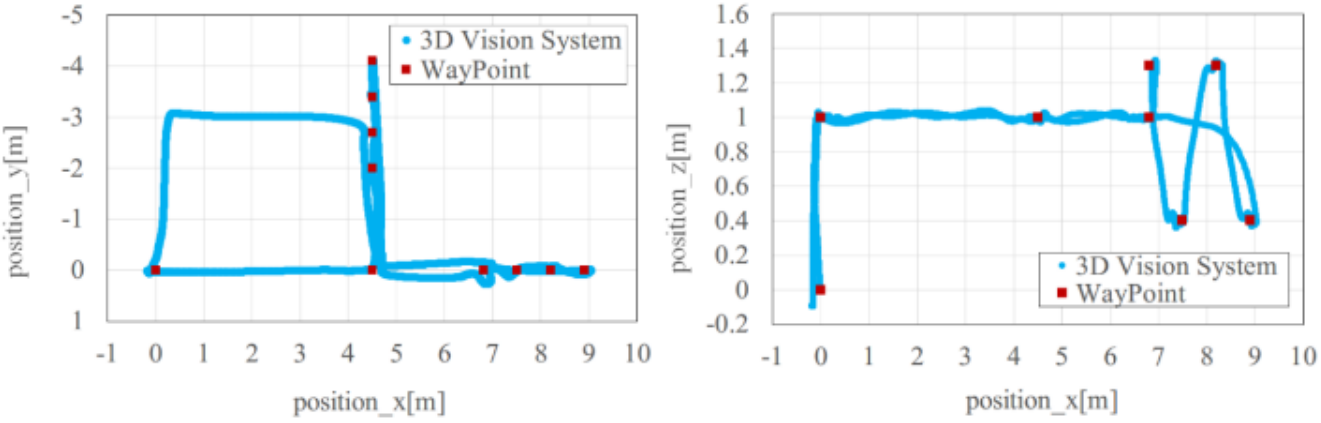

Fig. 6 Measured Trajectories of Scenario 2 at Japan Drone 2017 (left: x-y, right: x-z)

Fig. 7 Images of Alphabet Boards Captured with a Camera Mounted on the Drone

March 25th 2017 in Makuhari Messe, Chiba-city, Chiba, Japan (Fig.5). In the demonstration, we used only our vision system as self-positioning tool, did not use any infrastructure facilities such as GPS. The demonstration's movies have uploaded to Ricoh official channel[21] on YouTube.

Scenario 1: The drone autonomously avoids an approaching obstacle while hovering.
The drone autonomously takes off and start hovering. Then we close an obstacle to the drone. The drone detects the obstacle and automatically moves to left to avoid it. When the obstacle gets away, the drone returns to initial hovering position.

Scenario 2: The drone autonomously flights that assumes infrastructure inspection and automatically avoids an obstacle which appears on the route. The drone autonomously flights along some waypoints in this scenario. It assumes infrastructure inspection such as bridge, tunnel and plant using drone with cameras. Eight waypoints are set so that a camera mounted forwards on the top of drone captures some target boards (A ~ H) on the structure during the flight. If the drone accurately flight along all waypoints, all target boards are captured in the center of captured images. Although the drone automatically returns after completing the inspection, we set an obstacle on the route. The drone detects it, then flights on a route to avoid it and return initial position.

In the movie of scenario 1 demonstration on Japan Drone 2017, you can see that the drone stably hovered and automatically avoid so as not too close to an approaching obstacle. The drone returned to initial hovering position after the obstacles went away.
Fig. 6 shows all waypoints we set in advance and the trajectory measured by the vision system mounted on the drone at the movie of scenario 2. The left figure shows a trajectory (x-z) as seen from the above and the right one shows a trajectory (x-y) as seen from the side. Theirs origin is take-off position, positive direction of x-axis, y-axis, z-axis indicate respectively forward direction, right direction and upward direction from take-off points. All waypoints are on the trajectory and it means that the drone was automatically controlled to pass through all waypoints in according to self-position measured by the vision system.
Fig. 7 shows alphabet board images captured by a camera set on upper of the drone on scenario 2. The camera could shoot all alphabet board approximately in center on this flight. This indicates that self-position measured by the vision system was accurate and the drone was automatically controlled through expected path with its self-position. At point (x = 4.5m, y = 0m) on the left of Fig. 6, the drone returned to take-off point in a roundabout way. It shows the drone autonomously avoided to an obstacle appeared on the path.

4 Conclusion

We present an automatic flight system for small unmanned aerial vehicles (UAV, drone) using a super wide-angle stereo camera. We developed a pipeline for stereo VO. It consists of hardware

part and software part for high speed response. Due to its wide field of view, it is able to sense obstacles in the front, left, right, up and down. We integrated our system in a multi-copter. The autonomous drone completed an indoor waypoints navigation scenario and an obstacle avoidance scenario.

Although our system was able to complete indoor flight scenarios, the actual scenarios will be more complex and its distance will be more longer. Our future work is to improve the performance of the vision system and explore its applications.

References

[1] Grzonka S, Grisetti G, and Burgard W. *Towards a navigation system for autonomous indoor flying*, Proc. of the IEEE Int. Conf. On Robotics and Automation (ICRA), pp. 2878–2883, 2009.

[2] Raabe C, Henell D, Saad E, Vian J. *Aggressive navigation using high-speed natural feature point tracking*, Aerospace Conference IEEE, vol. 1, no. 13, pp. 1-8, 2014.

[3] Weiss S, Scaramuza D, Slegwart R, *Monocular-SLAM–Based Navigation for Autonomous Micro Helicopters in GPS-Denied Environments.* J. Field Robot., vol. 28, no. 6, pp. 854–874, 2011

[4] Artieda J, Sebastian J. M, Campoy P, Mondragon I. F, Mart´ınez C and Olivares M, *Visual 3D SLAM from UAVs*, J. of Intelligent and Robotic Systems archive, vol. 55, Issue 4-5, Pages 299 – 321, 2009

[5] Engel J, Sturm J, Cremers D, *Scale-Aware Navigation of a Low-Cost Quadrocopter with a Monocular Camera*, Robotics and Autonomous Systems, Volume 62, Issue 11, November 2014, Pages 1646-1656

[6] Moore J. D. R, Dantu K, *Autonomous MAV guidance with a lightweight omnidirectional vision sensor*, IEEE Intl. Conf. Robotics and Automation (ICRA), 2014

[7] Huang S. A, Bachrach A, Henry P, Krainin M, Maturana D, Fox D, and Roy N, *Visual odometry and mapping for autonomous flight using an RGB-D camera*, in Proc. IEEE International Symposium of Robotics Research (ISRR), 2011

[8] Engel J, Stueckler J, Cremers D. *Large-Scale Direct SLAM with Stereo Cameras*, Intl. Conf. on Intelligent Robot Systems (IROS), 2013

[9] Hornung A, Wurm K. M, Bennewitz M, Stachniss C, and Burgard W. *OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees in Autonomous Robots*, 2013.

[10] Huh S, Shim D. H, and Kim J. *Integrated Navigation System using Camera and Gimbaled Laser Scanner for Indoor and Outdoor Autonomous Flight of UAVs*, Intl. Conf. on Intelligent Robot Systems (IROS), 2013

[11] Mori T. and Scherer S, *First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles*. In Intl. Conf. on Robotics and Automation (ICRA), 2013

[12] Hrabar S, Sukhatme G S., Corke P, Usher K and Roberts J, *Combined Optic-Flow and Stereo-Based Navigation of Urban Canyons for a UAV*, Intl. Conf. on Intelligent Robot Systems (IROS), 2005

[13] Hrabar S, *3D Path Planning and Stereo-based Obstacle Avoidance for Rotorcraft UAVs*, Intl. Conf. on Intelligent Robot Systems (IROS), 2008

[14] Fu C, Carrio A, Campoy P, *Efficient Visual Odometry and Mapping for Unmanned Aerial Vehicle Using ARM-based Stereo Vision Pre-Processing System*, Intl. Conf. on Unmanned Aircraft Systems (ICUAS), 2015

[15] Schmid K, Ruess F, Suppa M and Burschka D, *State Estimation for highly dynamic Flying Systems using Key Frame Odometry with varying Time Delays*, Intl. Conf. on Intelligent Robots and Systems (IROS), 2012

[16] Schmid K, Tomi´c T, Ruess F, Hirschm¨uller H and Suppa M, Stereo Vision based indoor/outdoor Navigation for Flying Robots, Intl. Conf. on Intelligent Robot Systems (IROS), 2013

[17] Gohl P, Honegger D, Omari S, Achtelik M, Pollefeys M and Siegwart R, *Omnidirectional Visual Obstacle Detection using Embedded FPGA*, Intl. Conf. on Intelligent Robots and Systems (IROS), 2012

[18] Harris C and Stephens M.J., *A combined corner and edge detector*, in 4th Alvey Vision Conference Manchester, UK, 1988, pp. 147–151.

[19] J. Y. Bouguet, *Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm*, Technical Report, Intel Microprocessor Research Labs, 1999.

[20] Murray R M, Li Z, Sastry S S, *A mathematical introduction to robotic manipulation*, 1994

[21] Ricoh Official Channel: Ricoh 3D Vision System × Drone, https://www.youtube.com/watch?v=GEV9fnEcqX8, (2017)

Contact Author Email Address

tetsuo.yt.yamashita@jp.ricoh.com
yasuhiro.kajiwara@jp.ricoh.com
tsuchiya@mail.ecc.u-tokyo.ac.jp
tshinji@mail.ecc.u-tokyo.ac.jp

Copyright Statement