

# Vision based autonomous orientational control for aerial manipulation via on-board FPGA

Leewiwatwong Suphachart, Syohei Shimahara, Robert Ladig, Kazuhiro Shimonomura  
Department of Robotics, Ritsumeikan University  
Kusatsu, Shiga, 525-8577 Japan

{gr0220rv, rr0004hi, gr0150ff}@ed.ritsumei.ac.jp, skazu@fc.ritsumei.ac.jp

## Abstract

We describe an FPGA-based on-board control system for autonomous orientation of an aerial robot to assist aerial manipulation tasks. The system is able to apply yaw control to aid an operator to precisely position a drone when it is nearby a bar-like object. This is achieved by applying parallel Hough transform enhanced with a novel image space separation method, enabling highly reliable results in various circumstances combined with high performance. The feasibility of this approach is shown by applying the system to a multi-rotor aerial robot equipped with an upward directed robotic hand on top of the airframe developed for high altitude manipulation tasks. In order to grasp a bar-like object, orientation of the bar object is observed from the image data obtained by a monocular camera mounted on the robot. This data is then analyzed by the on-board FPGA system to control yaw angle of the aerial robot. In experiments, reliable yaw-orientation control of the aerial robot is achieved.

## 1. Introduction

The use of unmanned aerial vehicles (UAVs) has increased in various social applications such as surveillance, rescue missions and environmental data collection, due to their high mobility in three dimensional space. One of the most common applications for UAVs are high altitude tasks, given the UAVs ability to reach commonly difficult to approach areas at great height. Considering the practical application of an aerial robot, it might be desirable to rest the robot at a high altitude and shut down its main propulsion method, the robot's propellers. This can be achieved by grabbing a bar-like object that is serving as a resting station or landing point. After reliably grabbing such a bar, many possibilities open up. For example, in surveillance tasks, an aerial robot would be able to rest at a high altitude location to conserve its energy, instead of hovering in



Figure 1. Picture of the aerial robot platform, previously developed in our research group, with a robotic hand on the top of the airframe.

the air. In search and rescue tasks, especially flood disasters, where there are no safe grounds to land, grabbing an overhead object gives the robot a stable, safe landing opportunity. Our approach is assuming the high altitude resting station as a bar-like object. This shape can be widely found at high altitude locations in a typical urban environment *e.g.* branches of a tree, water pipes or power lines. There are numerous previous researches on aerial robot manipulation. For example, Kim *et al.* and Lucia *et al.* presented an aerial robot with attached two-DOF manipulator [1] [4]. A more complex design of manipulator mounted under a drone is presented by Orsag *et al.* [5]. Another example of an approach to solve the task to grasp an object midair, similar to this work, is the work of Thomas, Loianno *et al.*, who are applying a novel biomimetic method on this problem [8]. However, none of these concepts are suitable for approaching an overhead resting station in an uncontrolled environment due to their design considering only the workspace under the airframe or the reliance on a controlled environment and motion capturing equipment to extract the rigid body dynamics of the robot. In a previous approach of our research group, we have focused on the development

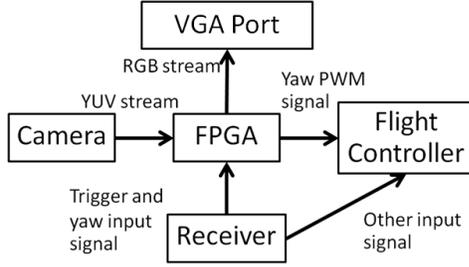


Figure 2. Connection diagram of the sensing and control system used in this study.

of an aerial robot mounted with upward manipulator and succeeded to design and develop a quadrotor, as shown in Fig.1, able to grasp a bar-shaped object when controlled by a skilled operator [6]. However, even though this previous work implemented a slider mechanism to allow higher error margin of horizontal positioning control, manual control of the aerial robot towards a bar-shaped object, especially considering that the size to the gripper aperture is only slightly bigger than the object to grasp, proved to be a very complicated task. This is why we considered an automatic yaw control system to relieve the operator of another DOF of control. In order to grasp a bar-like object, the following 3 steps must be achieved: 1) Adjusting the position of the robot to make sure the bar is centered relative to the gripper. 2) Adjusting the angle of the robot to be parallel to the bar i.e. perpendicular to the camera perspective and 3), adjusting the z position of the robot to enable the gripper to fully grasp the desired object. The first step is performed by using the previously mentioned sliding mechanism without adjusting the position of the airframe and interfering with the flight control. In this paper, we design a field programmable gate array (FPGA) implementable vision-based orientation control system for the aerial robot and accomplish the task of the second step by using visual feedback control from a camera adjusting the robot’s orientation.

## 2. Hardware design

### 2.1. Hexarotor base

The frame of the robot used in this study is a DJI F550. As a flight controller, a NAZA-M V2 main controller with integrated inertial measurement unit (IMU) is used in addition with its necessary peripheral modules. As transmitter and receiver, the FutabaT8J and the Futaba R2008SB is used. Additionally, we use the same top mounted gripper with sliding mechanism presented in a previous work [6].

### 2.2. Embedded vision system

We use a XC6SLX45 FPGA of the Xilinx Spartan 6 family, for visual processing. The FPGA connects to a camera

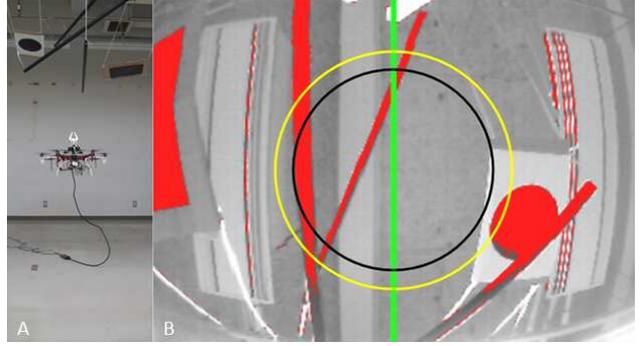


Figure 3. A: Flight experimental setup. B: Output image of the FPGA. The original picture recorded from a monocular camera on the top of the drone in (A) is shown as gray-scale. The red color indicates the result of binarization detecting the object. The green line shows the angle of the nearest bar-shaped object. The black circle defines our valid region, in which targets are considered. The yellow circle shows where we split the image for inner and outer Hough transform.

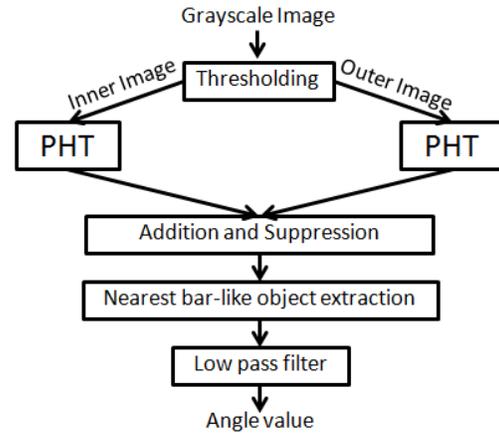


Figure 4. Flowchart of the computational processing, from received image to automatic yaw control.

that supports a resolution of 640x480 pixels and a frame rate of 30 fps. The camera image provides a 10 bit YUV image that is used to analyze the angle of the bar-object. The FPGA generates a pulse-width modulation (PWM) signal able to be recognized by the flight controller, archiving yaw control of the aerial robot. Additional input signals that connect to the FPGA are manual yaw control, that can optionally be forwarded directly to the flight controller and a trigger signal used to activate and deactivate automatic yaw control. This integrated flight controller add-on and signal shaping approach by using an FPGA is based on another previous work of our research group [3]. The other, non-manipulated signal ports of the receiver connect directly to the flight controller. Fig.2 demonstrates how devices connect to each other.

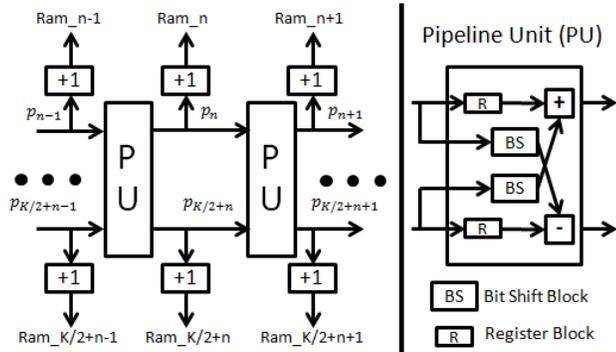


Figure 5. Parallel Hough transform architecture.

### 2.3. Output capture

We use a video grabber from Epiphan Video. The video grabber is used to record a 24bit RGB steam at 28 frames per second over a VGA port, programmed and implemented in FPGA. The external wiring is connected in a way that experimental results are not influenced.

## 3. Software design

### 3.1. Overview

The proposed flight assistance system is designed to detect and analyze a bar-like object. In some situations, a bar-like target is not distinguishable from other objects such as in Fig.3(A). To specify as a desired target, certain specifications must be met. First, the distance from the neutral axis of the object and the center of the camera must be within a certain range. This is indicated by a black circle, defining our valid region, in Fig.3(B). Second, the object length needs to be above a certain threshold and must not have large gaps when compared with the longest object within the black circle. Third, the object diameter has to be in a certain pre-determined range. If there is more than one detected object, the system will select the closest object, chosen by finding the largest diameter of the detected objects, then finding the shortest distance from the neutral axis of all detected objects to the center of the camera. The overall process of image processing is shown in Fig.4. Initially, the Y channel of the recorded image is thresholded, assuming that the target luminance is lower than that of the background. The whole thresholded image is then transformed to Hough space in order to distinguish between a bar and paired lines. This approach is practical in parallel Hough transform (PHT) since the number of valid object pixels make no difference in term of processing time. However, when applying this method, Hough transform (HT) can interpret non-bar-like objects or discontinuous bars as valid object. This problem is overcome by our method by performing Hough transform in an inner and an outer area of

the thresholded image. This inner area is located inside a certain region, shown as the yellow circle in Fig.3(B) and rest of the image is considered as the outer area. The diameter of this region equals the diameter of the region where objects are considered for detection (shown as the black circle in Fig.3(B)) plus the maximum acceptable bar-like object diameter.

### 3.2. Measurement of the angle of the object

For detecting a bar-like object in most environments, Hough transform (HT) is an essential algorithm in classification because of its resistance to noise. However, HT is not suitable for FPGA implementation due to its heavy resource requirements. Fast Increment Hough Transform 2 (FIHT2) however, developed by Koshimizu and Numada, has proved to be a comprehensive alternative to usual HT [2]. This is because it does not require any trigonometric calculations to generate the Hough distributions but still works as a line detector. In the work of Tagzout *et al.*, an implementation of FIHT2 in FPGA is presented [7]. We derive some of their work for our algorithm as defined by the following expression:

$$\begin{cases} p_{n+1} = p_n + \varepsilon p_{K/2+n} & 0 \leq n < K/2 \\ p_{K/2+n+1} = p_{K/2+n} - \varepsilon p_n \end{cases} \quad (1)$$

With  $p_0 = x, p_{K/2} = y$ , where  $n, \varepsilon$  and  $K$  are indices, the resolution and the number of quantizations in the  $\theta$  space. With this algorithm, the length of the Hough generator pipeline is divided by two because for each  $\theta$  value this algorithm can generate two  $p$  values at a time. Moreover, when  $\varepsilon$  is selected carefully, the multiplication in eq.(1) could be replaced by a shift operation. When implemented in FPGA, we model parallel Hough transform (PHT) as shown in Fig.5. In the figure, each ram contains both Hough space in inner and outer area for the reason that there is no requirement to access the rams at the same time. This leads to a significant decrease of FPGA slices to construct rams, while raising the number of ram accesses. Each pipeline unit requires only two registers, two bit shift blocks, one adder and one subtractor. The calculated value of  $p_n$  or  $p_{K/2+n}$  is treated as the corresponding memory address in the ram index  $n$  or  $K/2 + n$  with the accumulator of the value in this address incremented by one. The number of ram is related to the number of quantizations in the  $\theta$  space,  $K$ .

### 3.3. Addition and Suppression

As mentioned in section 3.1., we separate the image into inner and outer area. Each area is transformed to Hough space. The inner area Hough space verifies that only bars passing the valid region are detected. An addition and suppression process is applied to produce the final Hough result

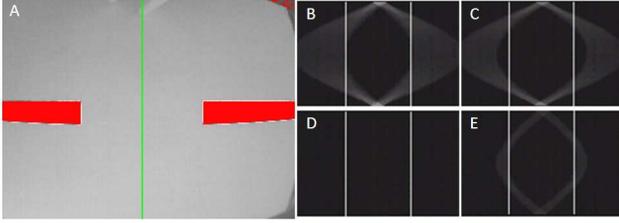


Figure 6. A: Output image after analyzing a discontinuous bar-like object. B: Hough space of the whole image. C: Hough space of the outer-area-image. D: The result of addition and suppression processing Hough space. E: Hough space of the inner-area-image.

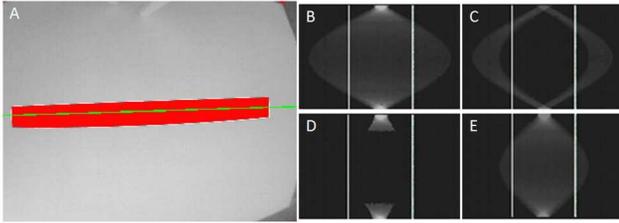


Figure 7. A: Output image after analyzing a continuous bar-like object. B: Hough space of the whole-image. C: Hough space of the outer-area-image. D: The result of addition and suppression processing Hough space. E: Hough space of the inner-area-image.

0	0	X	0	0	0	0	0	X
0	X	1	0	0	0	0	X	1
X	1	1	X	1	1	X	1	1
0	X	1	0	X	1	0	0	0
0	0	X	0	0	X	0	0	0
0	1	1	0	0	0	0	1	1
0	1	1	0	1	1	0	1	1
X	1	1	X	1	1	X	1	1
0	1	1	0	1	1	0	1	1
0	1	1	0	1	1	0	0	0

Figure 8. Tip template images. X indicates irrelevant values.

from each Hough space. The procedure scans the whole inner Hough space if the vote value is more than a specific value, proportional to the diameter of the yellow circle in Fig.3(B), adding the value in outer Hough space. Else the value at the same position in outer Hough space is replaced by zero. The effect of the method is shown in Fig.6 and Fig.7. In Fig.6(A), no bar is passing the valid region which results in blank Hough space as shown in Fig.6(D). On the other hand, In Fig.7(D), the main components in the Hough space still remain while insignificant values are eliminated. Moreover, a maximum vote value is searched simultaneously in this process within acceptable distance range, indicated by a black circle in image space in Fig.3(B) and by white lines in Hough space as seen in Fig.6 and 7.

**Data:** Hough space

**Result:** Nearest bar-like object

initialization;

**for** each angle index  $n$  **do**

start tip matching;

**for** each distance  $p$  **do**

**if** tip is found **then**

Save the tip, stop tip matching and start tail searching;

**end**

**if** tail is found **then**

**if** length is in acceptable range AND (length > local length OR (length = local length AND nearer to the vertical center) ) AND (tip+tail)/2 is in acceptable range

**then**

set local length to length and set local tip to tip ;  
// (tip+tail)/2 range is shown as white  
//lines in output

**end**

start tip matching and stop tail searching;

**end**

**end**

stop tip matching and tail searching;

**if** local tip = global tip AND local length <

$P * \text{global length}$  AND previous angle index = current index-1 **then**

**if** local length > global length **then**

set global length to local;

**end**

calculate average angle index and average distance, set global tip to local and set previous angle index to current;

**else**

**if** local length > global length **then**

set global length to local, set global tip to local and set new average angle index and distance;

**end**

**end**

set local length to zero;

**end**

**Algorithm 1:** Nearest bar-like object extraction

### 3.4. Nearest bar-like object extraction

In order to extract essential information from Hough space, thresholding is a fundamental step. The threshold value is determined from our selected ratio multiplied with the maximum vote from the previous process. If the maximum vote is lower than our specific constant, the threshold value is set to a certain minimum constant. Due to the

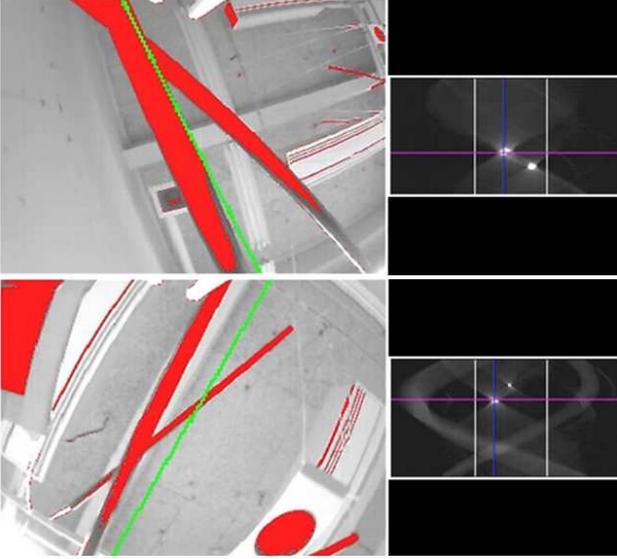


Figure 9. left: Output image without splitting area method. right: Hough space result without splitting area method.

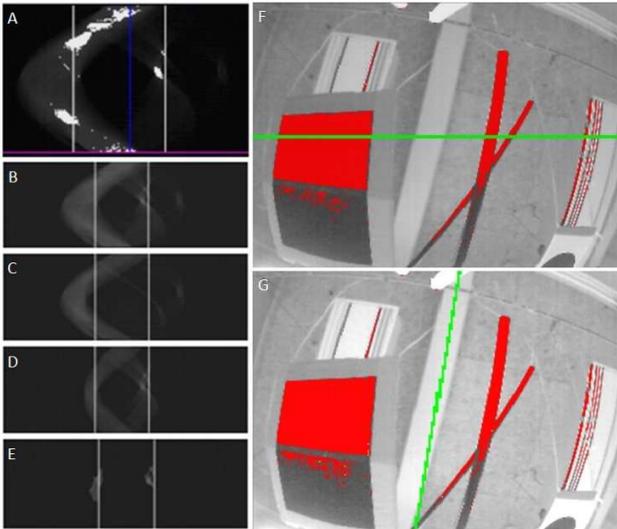


Figure 10. A: Hough space result without applying the method. B: Whole-image Hough space. C: Outer-area-image Hough space. D: Inner-area-image Hough space. E: The result of the method processing Hough space. F: Output image without splitting area method. G: Output image with the method

characteristics of Hough space, we follow the natural assumption that each bar-like object in image is transformed to a diamond, eclipse or hexagonal shaped object in Hough space. With this assumption we design tip template images as shown in Fig.8 to find tips of every object in Hough space. For tail searching, we just simply check for blank pixels. The pseudo code shown in Algorithm.1 is demon-

strating how nearest bar-like objects are extracted. The function of the algorithm is to find the largest acceptable diameter bar-like object from Hough space. In case there are more than one largest diameter objects i.e. two bars with exactly the same diameter, it will select the object which is closest to the vertical center. If the recognized tip is blunt, the result will be calculated from averaging blunt tips.

In our algorithm,  $length$  is equal to  $tip - tail + 1$ .  $Local\ length$  is the maximum  $length$  in each angle index  $n$ .  $Global\ length$  is the maximum  $length$  since the start of the process.  $Local\ tip$  and  $global\ tip$  are to verify the connection for calculating the average of  $angle\ index$  and  $distance$  in case the tip is blunt.  $P$  is gain in range 0 to 1.

### 3.5. Low pass filter

We implement a low pass filter to prevent the drone from chaotic and faulty rotation. Faulty yaw-rotation may occur when it is difficult to extract a bar-like object from the environment or the system is activated in an environment with no bar-objects. In general, yaw orientation between target and the robot can be considered constant in most environments since the yaw orientation is stabilized at a certain position by a merger of flight controller and IMU. This means the orientation of the system can be operated with intermittent yaw control signals. According to this fact, our filter is designed to collect and compare the collected data. If there is a change of yaw with an unreasonable high value, the filter will command the system to not rotate the robot. Else the filter will send an average of the data as a final output, initiating yaw control.

### 3.6. Yaw control

We create a PWM signal as the control signal for the robot to orient its angle perpendicular towards the preferred bar-like object. Duty cycle ( $D$ ) of the signal is determined as follows.

$$D = \begin{cases} T_{stable} + K(90 - angle) & \text{if } 45 \leq angle \leq 135 \\ C_+ & \text{if } 0 \leq angle < 45 \\ C_- & \text{if } 135 < angle \leq 180 \end{cases} \quad (2)$$

where  $T_{stable}$  is the duty cycle that almost makes a rotation,  $C_+$  is the duty cycle that makes allowed maximum counter clockwise rotational speed,  $C_-$  is the duty cycle that makes allowed a maximum clockwise rotational speed and  $K$  is the constant gain, which is equal to  $(C_+ - C_-)/2$ . We decide  $C_+$  and  $C_-$  from our gain experiment.

## 4. Experiments

### 4.1. Image Processing experiment

Examining the results of the algorithm when used without separated areas processing in Hough transform, we can

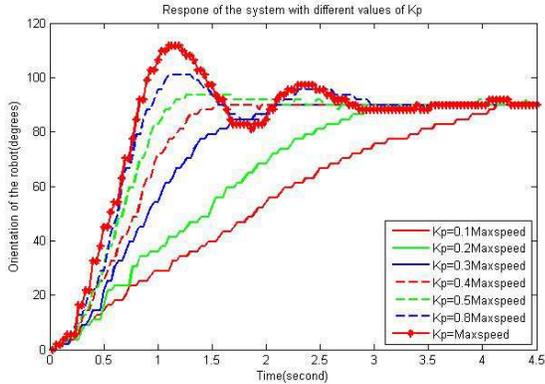


Figure 11. Response of the system with different values of  $K_p$

show the limitation of normal Hough transform and illustrate the advantage of our method. Without separation of the areas, we are able to extract the desired object when the object is clearly detected or the object is separated from non-bar objects. A successful result without separation is shown in Fig.9. In the right of Fig.9, the two vertical white lines indicate the valid region, the vertical blue line and horizontal purple line indicate the distance and angle index result from the nearest bar-like object extraction block, white color indicates the result of thresholding Hough space in nearest bar-like extraction. However, when the target is far from the center or object reflection is effecting the detection, it can make non-bar-like object more significant in Hough space, as the difference of target and other objects decrease. This leads to faulty detection of the object as shown in Fig.10(F). The reason in this particular example is the voting value from the square object in the background on the left side combined with the two black bars on the right side. It is high enough compared with the values from an unclear bar objects to result in a wrong detection. Our solution is to separate an area for transforming to Hough space as stated earlier. In Fig.10, we illustrate the usage of this method. The wrong detection, shown by the intersection of a blue line and a purple line, shown at the bottom of Hough space in Fig.10(A) is eliminated by our method as shown in Fig.10(E). The successful result of our enhanced detection method by separating areas for Hough transform is shown in Fig.10(G).

## 4.2. Gain adjustment

We have designed a safe experiential setup in order to investigate gain response and compose accurate gains for P-control of the robot. The setup can be separated into three parts: inner core, a connector and the robot. The inner core consists of an iron bar of 148cm length and a 5.3 kg weight base. A PVC pipe of slightly larger diameter than the iron bar is put around it, able to slide freely in an upward and

downward motion, and is mounted with an ABS 3D printed connector for attaching the robot to this sliding pipe. The large contact area between bar and pipe has the advantage of allowing upwards, downwards and rotational movement, making a test flight stable and limited to two-DOF, one of which is needed to be analyzed in detail. The experiment is done in two steps. In the first step, the robot automatically rotates its yaw in a 0 degree position so that the initial condition in every experiment is the same. Next, our proposed method is tested, making the aerial robot rotate its yaw to be perpendicular to a bar, clearly visible and located above airframe. The result of the experiment is shown in Fig.11. From those results, we found that a value between 0.4 and 0.5 *Maxspeed* gain is a suitable maximum to be implemented into the system because it is combining fast response and critical dampening properties.

## 4.3. Flight experiment

We have flown the aerial robot manually and approached black bars, 3 cm in diameter, placed at 1.6 m in height over the aerial robot. We then activated the trigger to switch to automatic yaw control mode. Our system smoothly adjusted the yaw of the robot to be perpendicular to the bar and maintained itself at this angle. The result of the experiment is shown in Fig.12 and the output from the FPGA attached on the robot is shown on Fig.13.

## 5. Discussion

While the experiment showed the success in orientation towards a bar, there are some situations in which it is difficult for the robot to accomplish the task. Those situations include a swinging bar or situations where strong side wind makes the robot rotate around yaw in a hover state, seeing as we use P control in the visual feedback system, which is insufficient to counter these problems. A solution would be implement a proper PID or PD control. In this research, we also assume that the object has low brightness and perform static thresholding to classify objects from background. In poor light situations such as at night or in a place with high intensity light, this assumption can mislead the classification. The solution would be to use a better thresholding method or apply preprocessing before thresholding. There are a few suitable adaptive threshold methods such as Otsu-, Kittler-Llingworth- and Local Entropy thresholding. Histogram Equalization for preprocessing could be useful in increasing contrast in the image. Our nearest bar-like object extraction relying on tip template matching can be improved by applying whole shape template matching instead of only the tip. However, various shapes of desired objects have to be investigated. Distortion resulting from a non-calibrated camera is not a significant source of error with the current camera lens, but when using a wider angled lens, the proposed method would benefit from an integrated camera

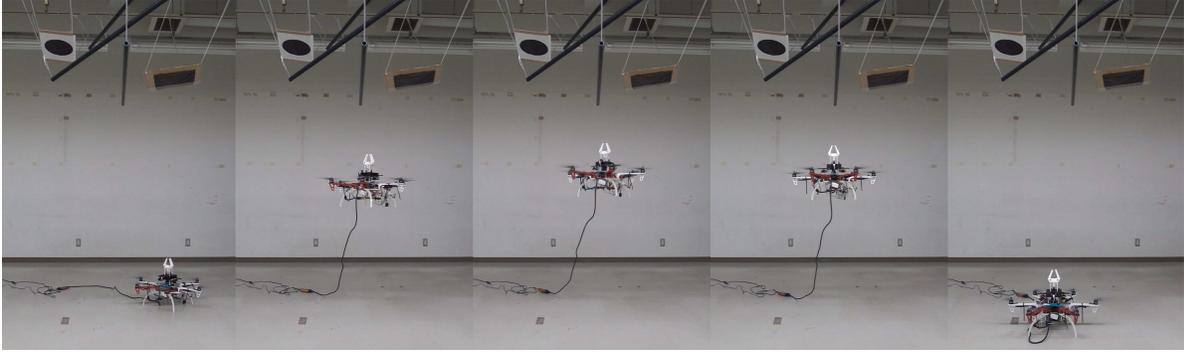


Figure 12. Flight experiment result of yaw orientating (from left to right): The robot was placed under the bar with its relative angle to the bar to about 45 degrees. Then, the robot is flown at some altitude and the automatic yaw control is activated while keeping it in a hovering state. Our proposed system adjusts the robot angle to be exactly perpendicular to the bar, making it easy now for the operator to grasp the bar-shaped object with the top mounted gripper. The robot is then safely landed on ground.

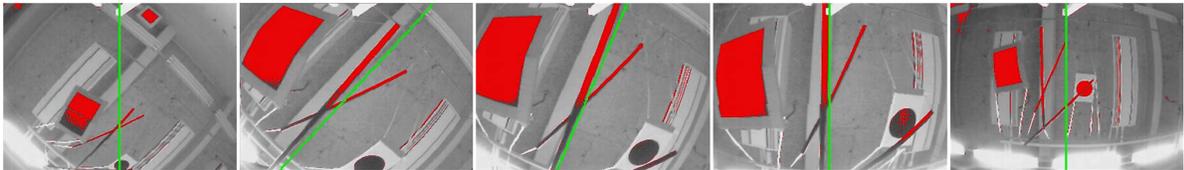


Figure 13. Video output of the robot in Fig. 12. The green line shows the currently detected most desired angle.

calibration method. Since the correct automatic horizontal positioning of a top mounted gripper is solved by the sliding mechanism of the gripper we use in this work and the correct autonomous rotational control of the aerial robot has now been solved, we are planning to implement a solution for automatic height control followed by the merging of horizontal position, yaw and height control via integrated FPGA computation to eliminate the need of an operator entirely and advance this project to make a completely autonomous system for aerial manipulation.

## 6. Summary

This paper describes an integrated on-board FPGA system able to autonomously align an aerial robot to a bar-like object by utilizing a novel separation method of image areas for Hough transform to achieve reliable recognition of bar-like objects for visual feedback control. In experiments, the robot succeeds to automatically orient itself to in a way that makes it convenient to proceed with further aerial manipulation tasks, such as grasping a bar-shaped object. The task of bar-like object grasping is essential in several aerial manipulation applications, such as parking at high altitude places for energy conservation or in case no other suitable landing spaces are available.

## References

- [1] S. Kim, S. Choi, and H. Kim. Aerial manipulation using a quadrotor with a two dof robotic arm. *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference*, pages 4990–4995, 2013.
- [2] H. Koshimizu and M. Numada. Fiht2 algorithm: a fast incremental hough transform. *IEICE TRANSACTIONS on Information and Systems*, pages 3389–3393, 1992.
- [3] R. Ladig and K. Shimonomura. Fpga-based fast response image analysis for autonomous or semi-autonomous indoor flight. *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 682–687, 2014.
- [4] S. D. Lucia, G. D. Tipaldi, and W. Burgard. Attitude stabilization control of an aerial manipulator using a quaternion-based backstepping approach. *Mobile Robots (ECMR), 2015 European Conference*, pages 1–6, 2015.
- [5] M. Orsag, C. Korpela, M. Pekala, and P. Oh. Stability control in aerial manipulation. *American Control Conference (ACC)*, pages 5581–5586, 2013.
- [6] S. Shimahara, R. Ladig, L. Suphachart, S. Hirai, and K. Shimonomura. Aerial manipulation for the workspace above the airframe. *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference*, pages 1453–1458, 2015.
- [7] S. Tagzout, K. Achour, and O. Djekoune. Hough transform algorithm for fpga implementation. *Signal Processing Systems, 2000. SiPS 2000. 2000 IEEE Workshop*, pages 384–393, 2000.
- [8] J. Thomas, G. Loianno, K. Sreenath, and V. Kumar. Toward image based visual servoing for aerial grasping and perching. *Robotics and Automation (ICRA), 2014 IEEE International Conference*, pages 2113–2118, 2014.