Monocular Vision based Lane Following by AR-Drone

Saad Hassan, Sohaib Elahi
Department of Electrical Engineering
SBA School of Science & Engineering, LUMS, Pakistan
{14060031,15100096}@lums.edu.pk

1. Abstract

Nowadays, there are many robotic applications being developed to do tasks autonomously without any interactions or commands from human. Therefore, developing a system which enables an aerial robot to detect and track certain lanes autonomously will lead us to more advanced tasks carried out by robots in the future. In this project, we present a monocular vision based lane detection and following system for micro aerial vehicle (MAV). Robotic platform used in this project for development is AR-Drone quad copter available in lab. Front camera of AR-Drone is used as input sensor for the system. Computer vision algorithms such as Hough Transform are used to detect lines in input image. In this project, developed system is able to detect and track specified lanes in the scene successfully.

2. Introduction

In the recent years, research interest in the field aerial robotics has grown rapidly. There are various applications of aerial robotics in the field of military and civil domains such as aerial manipulation, surveillance and transportation [1]. A micro aerial vehicle (MAV) is class of aerial vehicle which is one of the most flexible and adaptable platform for aerial research due to its easy control and small size. However commercial MAVs are too expensive to be used by students and small research teams [2]. Therefore in this project I will use a low cost AR-Drone available in lab as an experimental platform. Sensors available on this platform are two cameras, 6-degree-of-freedom inertial measurement unit and sonar-based altimeter. The first camera is aimed forward and provides 640 × 480 pixel color image. The second one is mounted on the bottom and provides color image with 176 × 144 pixels [2]. This AR-Drone is shown in Figure 1.

Figure 1: AR-Drone quad copter

Previous work on autonomous flight with quad copters can be categorized into different research areas. One part of the community focuses on accurate quad copter control and a number of impressive results have been published [3]. These works however rely on advanced external tracking systems, restricting their use to a lab environment.

Autonomous lane detection and tracking by ground and aerial robots is a very well studied problem and have many applications. Basically, there are two classes of approaches used in lane detection: the feature-based technique and the model-based technique [4]. The feature based technique detects the lanes in the road images by combining the low-level features, such as painted lines or lane edges [5]. On the other hand, the model-based technique just uses a few parameters to represent the lanes. Assuming the shapes of lane can be presented by either straight line or parabolic curve [6], the processing of detecting lanes is approached as the processing of calculating those model parameters.

3. MY APPROACH

Overall project is divided into two parts: Lane detection (Computer Vision Part) and Lane following (Control Part).

3.1. LANE DETECTION:

In this project a simple, real time and effective approach is used to detect lanes in the image. First of all edges are detected in input image using canny edge detector. After edge detection, straight lines are detected in image using Hough Transform. Output of Hough Transform may include lines which are outside our region of interest (ROI). Therefore after line detection, ROI is extracted using known geometrical information of the environment. In the end, point of intersection of lines is detected. This point of intersection is also known as horizon point. Overall workflow of lane detection part is shown in Figure 2.

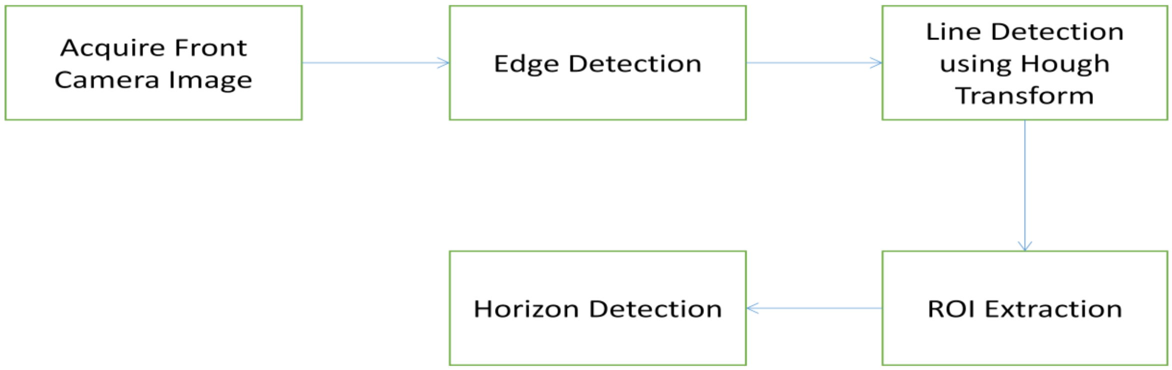
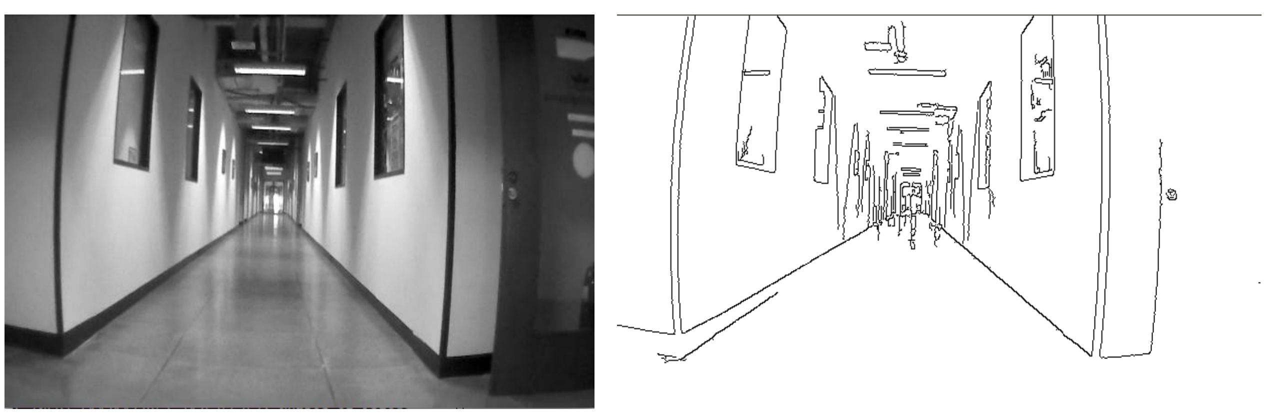


Figure 2: Workflow of lane detection part

3.1.1. Edge Detection:

First step after acquiring image from front camera of AR-Drone is to detect edges in the image. This is done using canny edge detector. Canny edge detector takes gray scale image as its input, and returns a binary image of the same size as before but with 1's where the function finds edges and 0's elsewhere. The Canny method finds edges by looking for local maxima at the gradient of the image. The gradient is calculated using the derivative of a Gaussian filter. This method uses two thresholds, to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. Canny edge detector is a very well known and reliable method for edge detection. First of all, input image is converted from RGB to GRAY scale and then edges are detected in the image using canny edge detector. Result of edge detection part can be seen in Figure 3.



Original Image Edge Detection

Figure 3: Canny Edge Detector

3.1.2. LINE DETECTION:

After edge detection, next step is to detect lines in the image. This is done using Hough Transform. The Hough Transform is a feature extraction technique which is mostly used for detecting lines, but can also be extended to detect different shapes like circles and ellipses. The simplest case of Hough transform is the linear transform for detecting straight lines. In the image space, the straight line can be described as $y = mx + b$. In the Hough transform, a main idea is to consider the characteristics of the straight line not as image points $(x1, y1)$ and $(x2, y2)$ etc., but instead, in terms of its parameters, i.e., the slope parameter m and the intercept parameter b. Based on that fact, the straight line $y = mx + b$ can be represented as a point (b, m) in the parameter space. However, one faces the problem that vertical lines give rise to unbounded values of the parameters m and b. For computational reasons, it is therefore better to use a different pair of parameters, denoted as r and Θ, for the lines in the Hough transform. These are the Polar Coordinates. The parameter 'r' represents distance between line and origin and Θ is angle of vector from origin to the line. This can be seen in Figure 4.

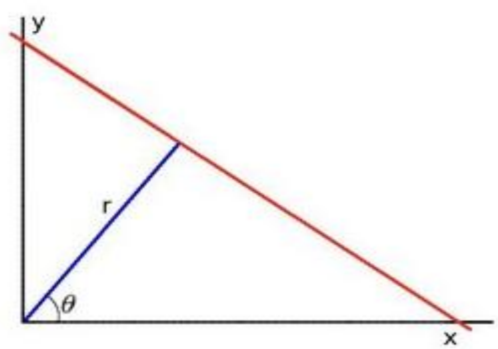


Figure 4: Line in polar coordinates

The transform is implemented by quantizing the Hough parameter space into finite intervals or accumulator cells. As the algorithm runs, each (xi, yi) is transformed into a discretized (r, Θ) curve and the accumulator cells which lie along this curve are incremented. Resulting peaks in the accumulator array represent strong evidence that a corresponding straight line exists in the image. Hence Hough Transform is used to detect lines in input image.

3.1.3. ROI EXTRACTION:

Hough Transform will detect all possible straight lines in the image. This may include lines, which are outside our ROI. For example in our case, horizontal lines are not in ROI as lanes to be tracked appear vertical in the view of AR-Drone camera. So after Hough Transform, next task is to extract ROI. Output of Hough Transform contains r and ϴ of all detected lines. On the basis of values of ϴ of detected lines and known geometry of environment, lines which are outside ROI are discarded. Result of Hough Transform and ROI Extraction can be seen in Figure 5.

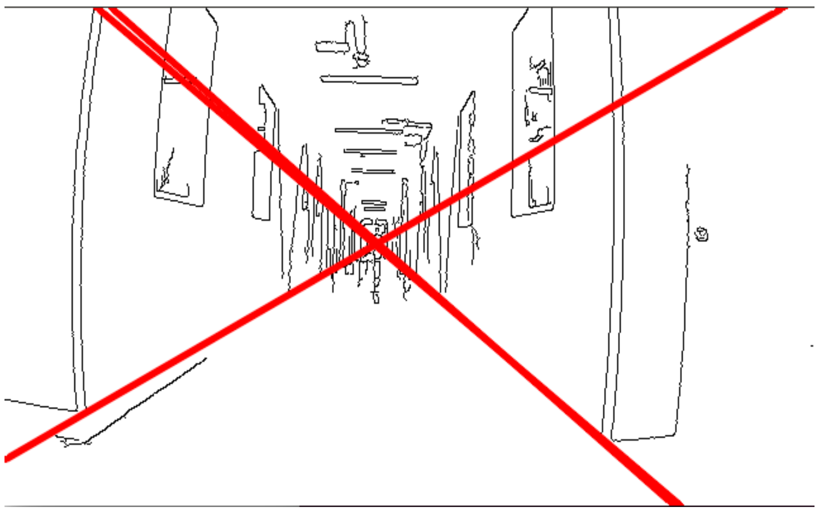


Figure 5: Hough Transform + ROI Extraction

3.1.4. Horizon Detection:

After successfully detection of required lines in the image, next task is to detect point of intersection of these lines. This point of intersection is also known as horizon point and in this case it is used to calculate center point of lanes detected. As r and ϴ of lines are known, so point of intersection can be calculated easily by solving these two equations simultaneously:

$$r1 = xcos\Theta 1 + \mathrm{y}\sin\Theta 1 \text{ .... } (1)$$

$$r2 = xcos\Theta 2 + \mathrm{y}\sin\Theta 2 \text{ .... } (2)$$

Result of this horizon detection part can be seen in Figure 6.

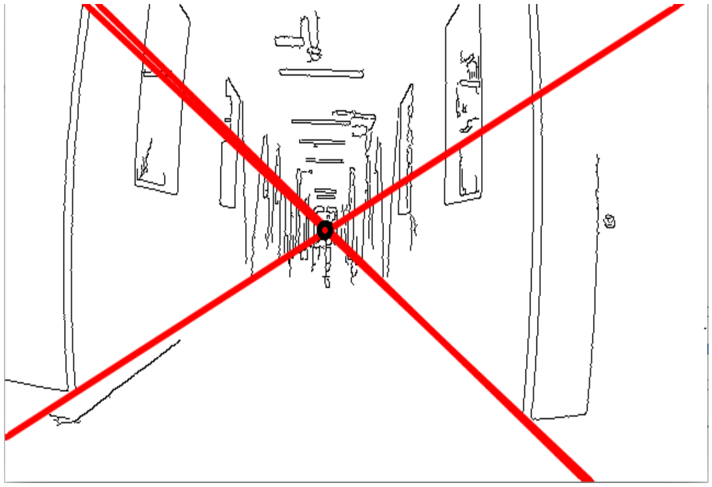


Figure 6: Horizon Detection

At this point, as point of intersection of both lines and their slopes are known so angle between these two lines ‘ θ_{center}’ can be computed using following equation.

$$\tan(\Theta_{center}) = (m1 - m2)/(1 - m1m2) \quad \text{.......................................} \quad (3)$$

As both θ_{center} and point of intersection of lines are known, so by using this information a line in center of detected lanes can be drawn. Result of this part can be shown in Figure 7.

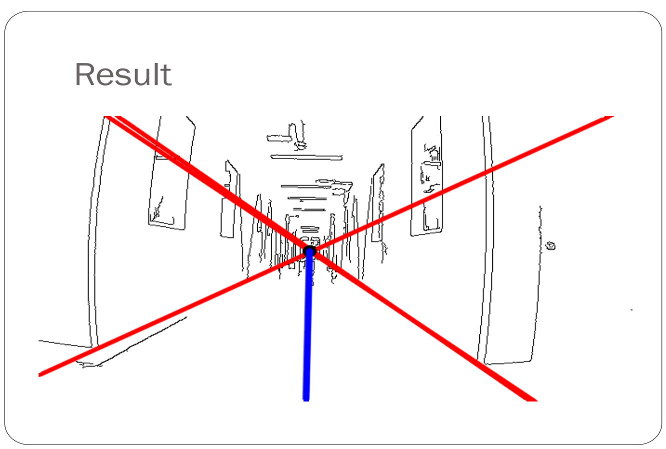


Figure 7: Line in center of detected lanes

3.2. Lane Following:

After successful detection of lanes and horizon point, next task is to design a control algorithm to track center point (Horizon) of these lanes. Center point of image is taken as reference point. Overall control loop can be seen in Figure 8.

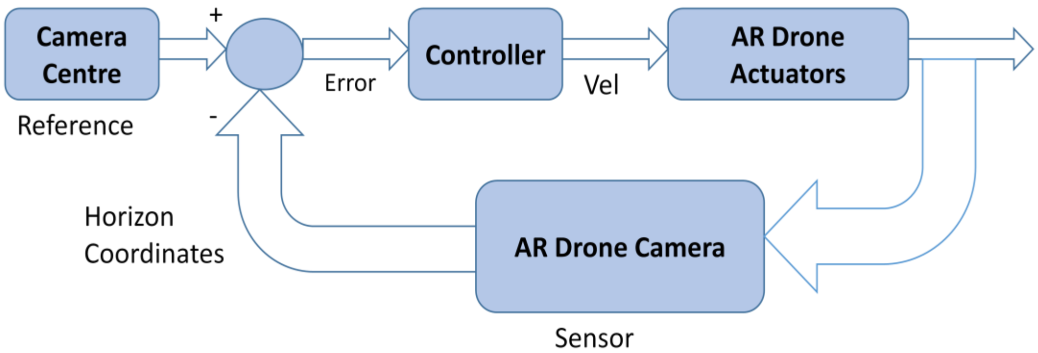


Figure 8: Line in center of detected lanes

Error is difference between image center and horizon point in term of pixels. The task of controller is to minimize error, which is difference between reference point (image center) and horizon coordinates. This is illustrated in Figure 9.

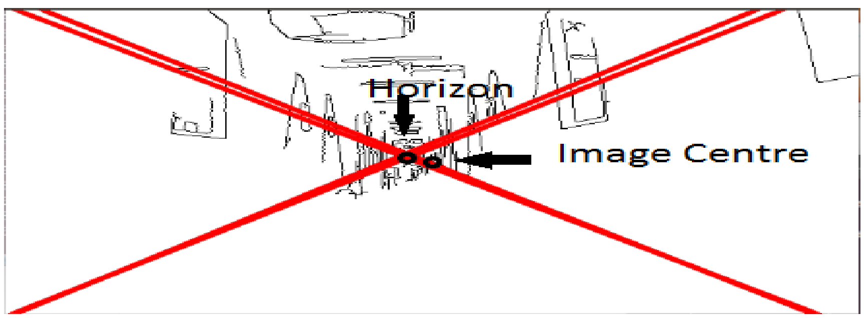


Figure 9: Horizon + Image center

As it can be seen in Figure 9 that image center is not exactly aligned with horizon point, therefore in order to keep AR-Drone in center of lanes detected, controller must try to minimize difference between image center and horizon. As AR-Drone platform has a built-in functionality to stabilize at certain height, therefore task of our controller is to just minimize horizontal distance between reference and horizon. There is no need to take care about vertical displacement between these two points due to built-in functionality of stabilizing at certain height. In this project two control strategies are used.

3.2.1. On-Off Control:

On-off control is the simplest form of feedback control. In this case, algorithm used can be illustrated by following flow chart

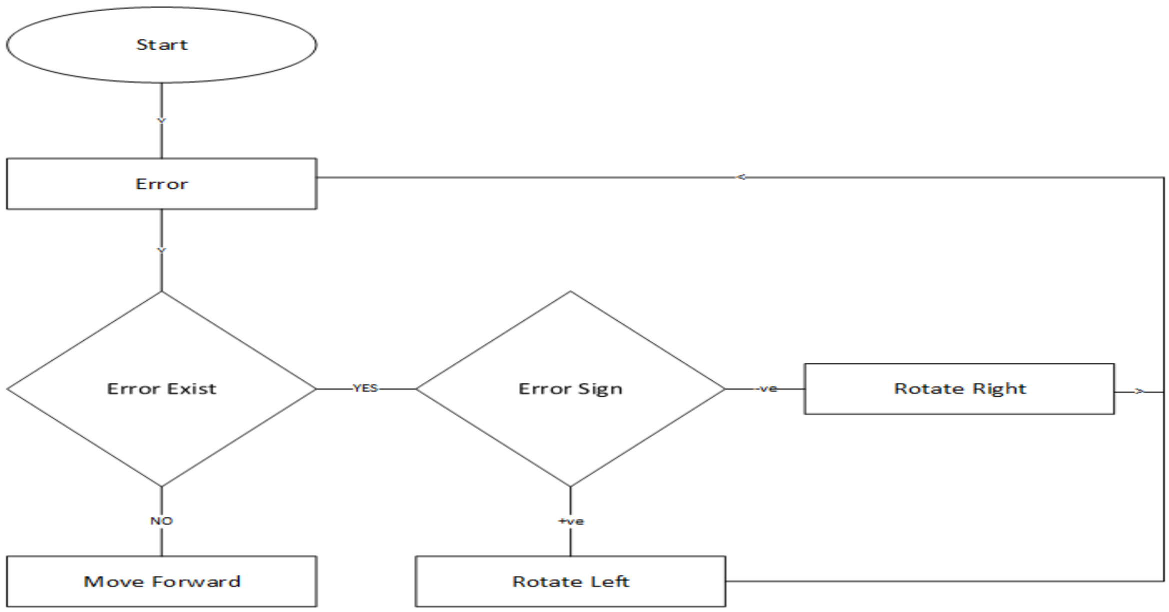


Figure 10: Flow Chart for On-Off Control

It can be seen from above flow chart that first of all error is calculated. If error is very small or close to zero then it means that AR-Drone is approximately in center of lanes, therefore a move forward command will be given to AR-Drone. But if error is not close to zero then it means that AR-Drone is not in center of lanes and also its orientation is not perfectly aligned, therefore in this case there is a need to correct error first. This error is corrected by giving fixed values of positive or negative angular velocities to AR-Drone depending upon signs of error. Once error is corrected, move forward command will be issued again.

3.2.2. Hybrid Control:

On-off control is not an optimal controller choice for practical systems due to oscillations produced by it in the system. Therefore an optimal controller is needed in order to successfully track lanes. After trying different approaches, we implemented a hybrid kind of controller which uses two different techniques to control linear and angular velocities of AR-Drone. In order to control linear velocity of AR-Drone, we used a control methodology which is basically inspired from Stanley control algorithm with some modifications. Linear velocity which was fixed in case of on-off control is now varied using following equation:

$$vel_{linear} = \left(1 - \left|\tan^{-1}\frac{err}{err\_max}\right|\right) * vel\_max \qquad \text{.. (4)}$$

Where 'err' is the value of error which is difference between horizon and image center. 'err_max' is maximum value of error and 'vel_max' is maximum linear velocity. So by using this equation, linear velocity increases as error decreases and vice versa. This gives us a better control on linear velocity of AR-Drone depending upon error value.

Yaw motion of AR-Drone is controlled using a PID controller. Error between reference point and horizon is reduced by varying angular velocity of AR-Drone. Values of angular velocities are decided by using a well known PID controller.

4. Experiments & Analysis

All experiments are conducted in corridor outside controls lab. Black lines on the bottom of sidewalls are taken as lanes to be tracked. A low cost AR-Drone available in lab is used as an experimental platform for testing of algorithm developed. AR-Drone has limited computational complexity; however image processing with computer vision algorithms need pretty high resources. Due to this reason, all computations will be conducted in a laptop connected to AR-Drone wirelessly. This communication will take place through a ROS platform running on laptop. This communication between AR-Drone and computer can be seen in the Figure11.

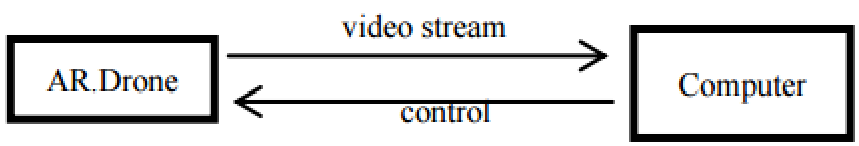
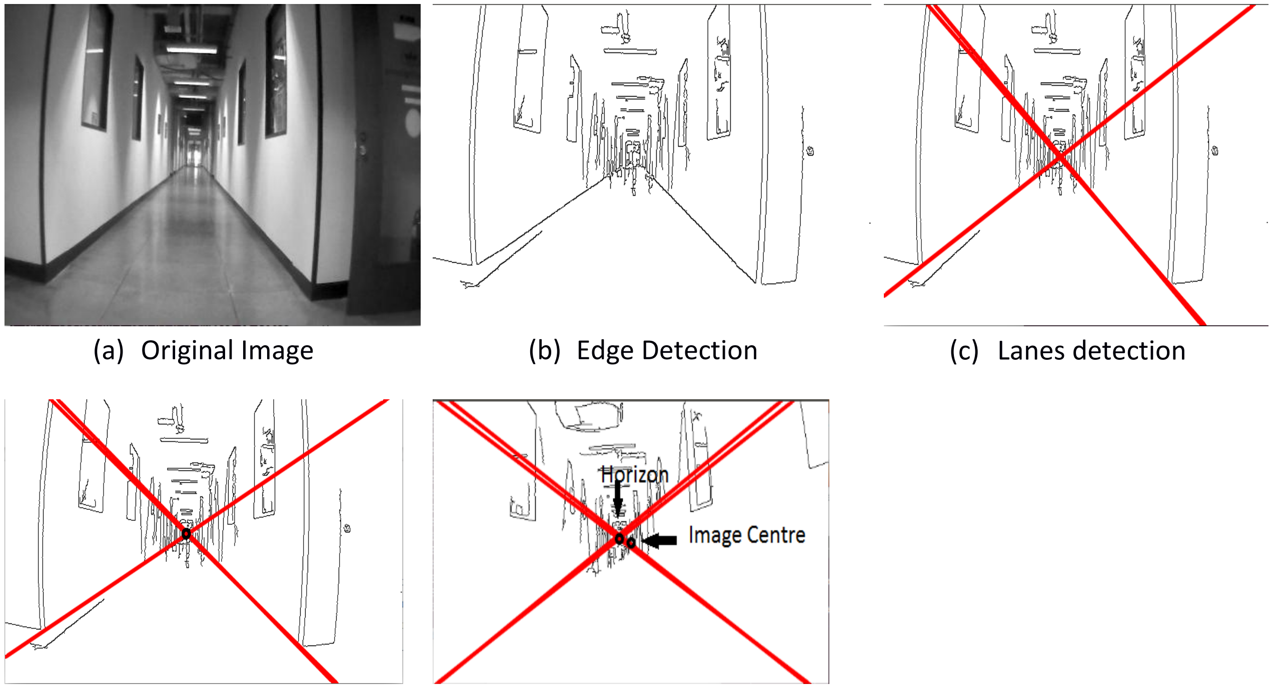


Figure 11:Communication process between AR-Drone and computer

Main challenging task in lane detection part was input camera noise. In order to deal with this noise, a moving averaging filter is implemented. This filter smoothed up output of camera to some extent. There is also an upper and a lower bound defined on the error in order to keep error bounded. If in any case error increases from its bounds, then system goes into a fail-safe state. Results of lane detection part are summarized in Figure 12.



(a) Original Image (b) Edge Detection (c) Lanes detection

(d) Horizon detection Horizon + Image center

Figure 12:Results of lane detection part

As explained earlier, that in lane following part first of all, a simple on-off control is implemented. But simple on-off controller is not good enough to precisely control motion of AR-Drone. With simple on-off control, AR-Drone followed a zigzag trajectory due to nature of controller. It also went into fail-safe state most of the times due to fact that error is increased from bounded limits due to oscillatory nature of on-off controller.

In order to deal with issues of on-off control, a hybrid control strategy is used as explained earlier. This controller performed much better than on-off control strategy. It has removed oscillations from the system produced due to on-off control and AR-Drone followed a much smoothed trajectory as compared to on-off control. These results can be seen in practical demo. AR-Drone can seen in Figure 13.

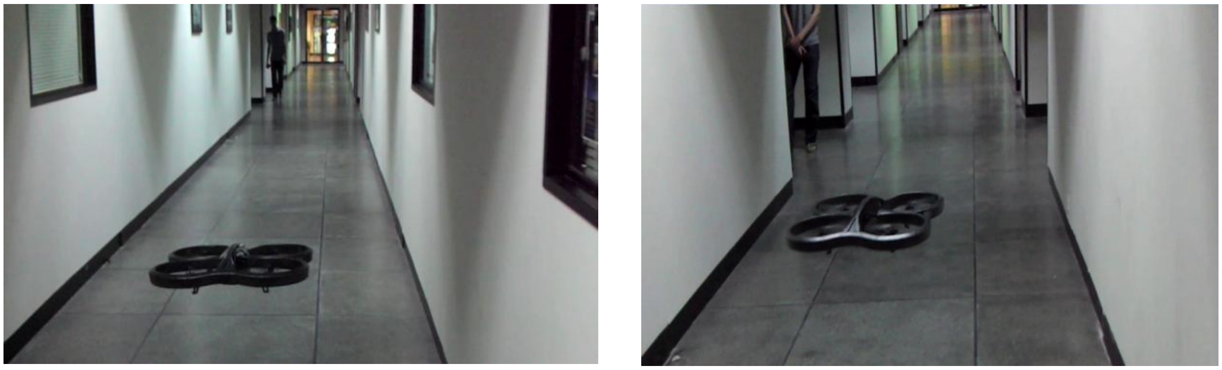


Figure 13: AR-Drone tracking Lanes

5. Future Work & Conclusion

In this project a simple, real time and effective approach is used to detect lanes in the image. Our developed algorithm is able to successfully detect required lines in the image. After lane detection part, two different control algorithms are used to control AR-Drone. Experiments showed that hybrid control technique performed much better than on-off control.

In future, this system can be modified to track a canal instead of lanes in indoor environment. Resultant trajectory can also be refined further by tuning gain values of PID.

References

[1] Inkyu Sa, H. H. (2013). Monocular Vision based Autonomous Navigation for a Cost-Effective Open-Source MAVs in GPS-denied Environments. Preprint submitted to 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.

[2] *Krajnık, T., Vonasek, V., Fiser, D., Faigl, J. (2011). AR-Drone as a Platform for Robotic Research and Education.In: Research and Education in Robotics Heidelberg, Springer.*

[3] M. Muller, S. Lupashin, and R. D'Andrea, "Quad copter ball juggling," in Proc. IEEE Intl. Conf. on Intelligent Robots and Systems (IROS), 2011.

[4] Wang, Y., et al. (2004) "Lane detection and tracking using B-Snake," in Proc. Image and Vision Computing (269-280).

[5] A. Broggi, Robust real-time lane and road detection in critical shadow conditions, Proceedings IEEE International Symposium on Computer Vision, Coral Gables, Florida, November 19–21 (1995).

[6] A. Kaske, R. Husson, D. Wolf, Chi-square fitting of deformable templates for lane boundary detection, IAR Annual Meeting '95, November 1995 (1995) Grenoble France.