Tracking a Line for Automatic Piloting System of Drone

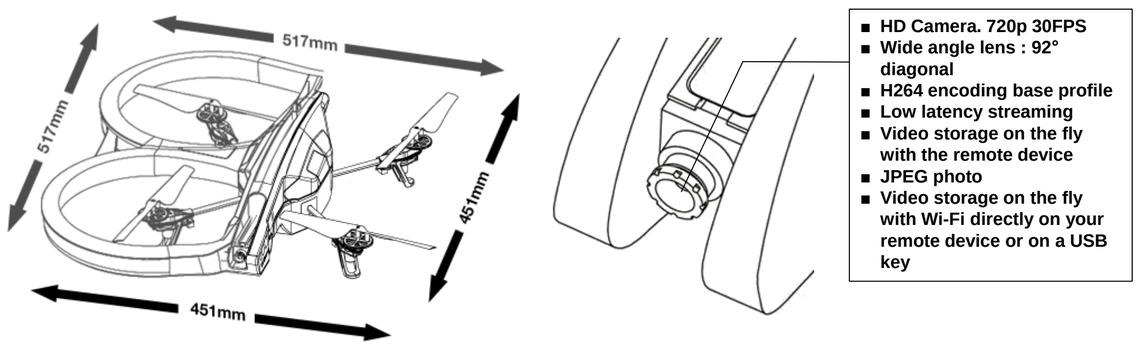
Yawei Yang
Advisor Prof. Jeongkyu Lee
Department of Computer Science and Engineering
University of Bridgeport, Bridgeport, CT

Abstract

Nowadays, the FAA is discussing about open the low area for registered flight, so the Drone for Amazon will be the possible. In addition, Amazon is developing its own drone for future delivery method. In this project, we implement automatic piloting system for AR. Drone by tracking a line with on-board camera. This project uses AR.Drone for our implementation of the line tracker because there exist several available SDKs based on Wi-Fi network and two HD quality cameras. For the SDK, YA.Drone from University in Hamburg, Germany is employed and it is combined with Image Processing Technology in order for a drone to track the line. The preliminary results show that a drone can successfully track various types of lines behind a drone, such as straight, 90 degree, crank, circle and arbitrary lines. Using the automatic piloting system in this project, a drone can send something from one room to other room, and eventually deliver an item outdoor.

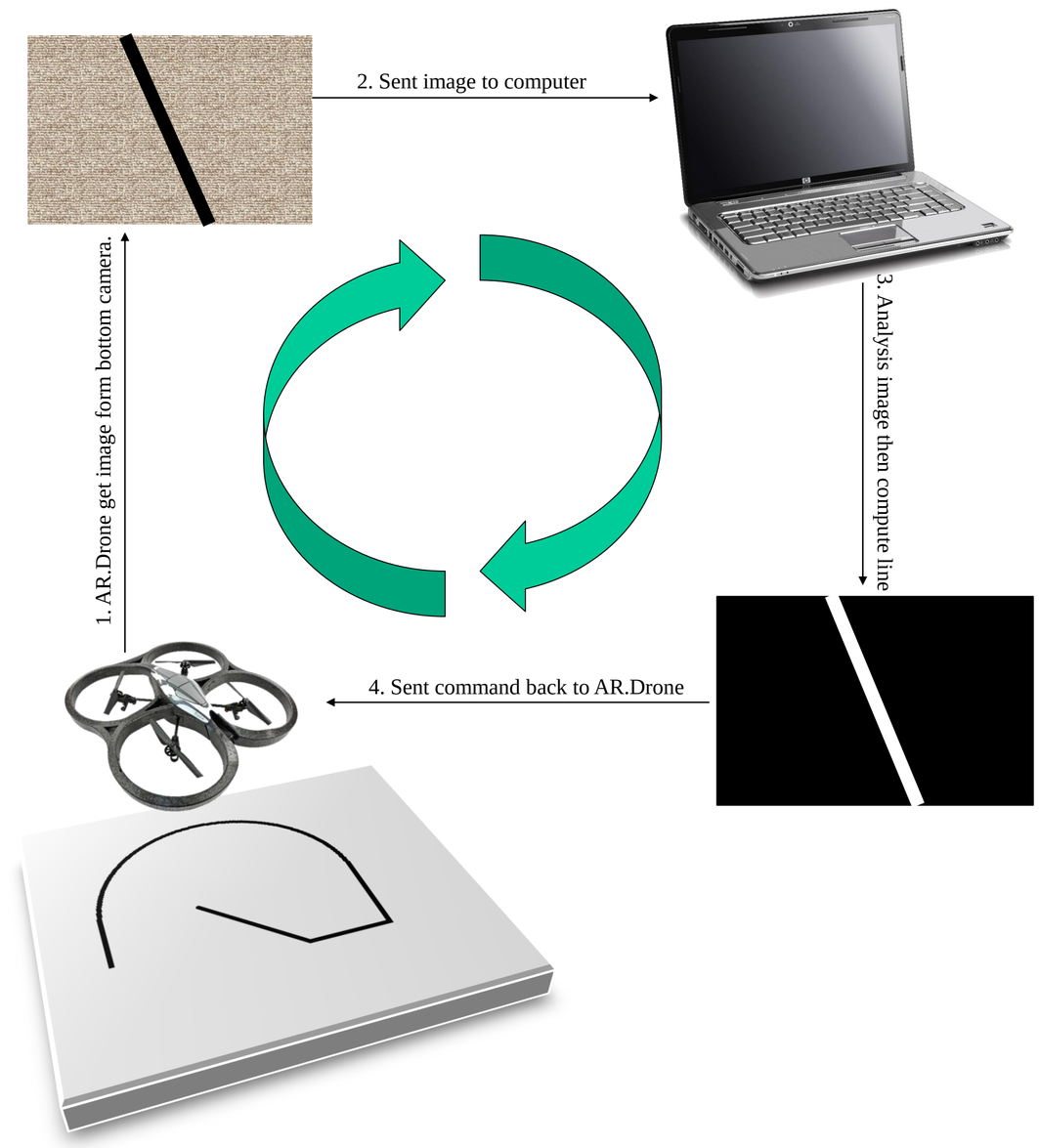
Design and Implementation

AR.Drone, is a small quadcopter from Parrot like this.



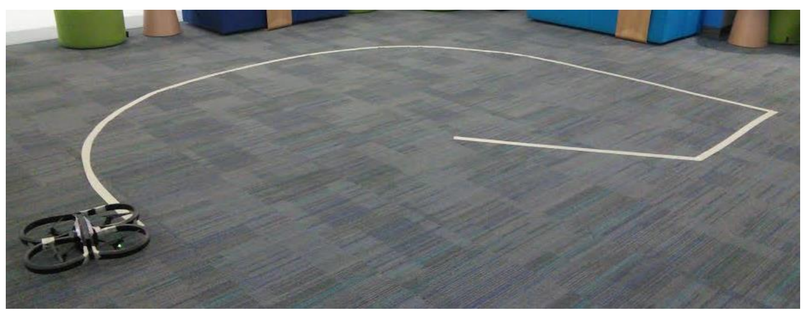
First, we need get image from AR.Drone bottom camera. After that, this program will binary the image base on different RGB range, using OpenCV Library to generate binary image to show which is line. It include default value, however based on different environment, it also allow manually choose the specific color range. analyze the result image, and calculate boundary of the image for four bounders, then sent the command to AR.Drone to turn or go straight.

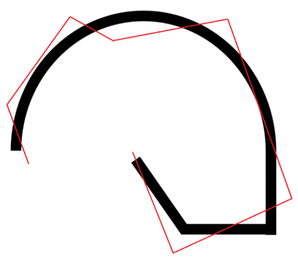
For algorithm, the image we get form AR.Drone will be binary with OpenCV ***inRange*** function, (this will be reset the image to 1 and 0, only two value) . After that, we will detect the first row and last row, first column and last column, to make sure that where is the line. The image will be separate into three part, (Left, Middle, Right). In the left and right part will be turn directly, and based on distance with the center, the turn power will be different. And in the middle part, AR.Drone just go straight to avoid that only 1 pixel can go stragith. If only bottom row detective the line, then land on the ground. So based on different conditions to sent different command.



Final Result

Now the AR.Drone can detect different color line but only work on the non-mirror reflection ground, like rug or curtain. Because if the lightness is too strong that the camera will cannot detect other color. It can finish different type of turn and curve. i.e. straight, 90 degree, crank, circle and arbitrary lines as figures.





Black Bold Line is Target Path
Red Narrow Line is Real Path

Conclusion

The problem is the latency for communicate and analysis. The SDK some time cannot get image form AR.Drone. Also the when analysis the image, still 0.5 seconds late, when sent back to AR.Drone, it already fly away, and maybe lost the line. So the same result when SDK cannot get image.

For algorithm if we have a cross “+” for line, AR.Drone only can go straight, and if it’s “Y” corner, AR.Drone will only choose left side.

References

1. http://opencv.org/
2. http://vsis-www.informatik.uni-hamburg.de /oldServer/teaching//projects/yadrone/
3. http://cdn.ardrone2.parrot.com/

Multimedia Information Group