CS 365 Project Report

Vision Based Helipad Detection and Estimation of Target Direction

Courtesy : http://ardrone2.parrot.com/usa/

Mentor: Prof. Amitabha Mukerjee

By:

Nitish Sharma (10465)
Anurag Pandey (10137)

Abstract:

In this project, the problem of landing target tracking by an **Unmanned Air Vehicle** –an AR DRONE has been addressed in order to facilitate its autonomous landing. We have used vision so as to make the helicopter detect, recognise and thereby locate the landing target (which would be a helipad of some definite shape) in real-time. The detection and recognition have been done through **Image Processing** techniques and subsequently, the relative location of the Helipad with respect to the helicopter has been calculated and shown. We have also taken the effects of pitch, yaw and roll of the helicopter so as to calculate the target direction precisely.

Motivation and Introduction:

Unmanned Aerial Vehicles have been explored a lot in recent times. They play crucial roles, especially in situations where human intervention becomes inconceivable, highly risky or highly expensive. These situations may include threatful material recovery, disaster relief support, traffic monitoring etc. They are also used in remote sensing applications or they can be developed for geophysical exploration i.e. in oil, gas and mineral exploration and production activities e.g. the *InView Unmanned Aircraft System,* which was developed in 2010 by Barnard Microsystems Limited for the purpose of oil and gas exploration. UAVs are highly useful in penetrating areas which may be too dangerous for a manned craft e.g. UAVs have been developed recently which can fly into a hurricane and communicate near-real-time data directly to the National Hurricane Centre in Florida.

In order to make a UAV function properly, it should have the capability of autonomous landing. This has been an issue which has gained the attention after the US Congressional Research Service in its report on Pilotless Drones dated September 10, 2012 [2] reported that "*despite improvements the accident rate for unmanned aircraft is still far better that that of manned aircraft, many a times the reason being the manual mistake by the operators while making the aircraft to land*." After this report, the importance of an autonomous landing was felt against an operator-controlled landing. This is also very crucial because of the very high instability of the helicopter close to the ground. We therefore are trying to address this very serious issue of autonomous landing of a UAV.

Previous Work

Vision based robot-control have been an active topic of research in the past few years. In [3], a real time computer vision algorithm is presented for tracking a landing target but there was no autonomous landing. Some researchers like in [4] and [5], treated the problem of autonomous landing and vision based tracking in a decoupled way.
A few vision based navigation and landing system were presented in the very early years of this 21st century. Among the popular ones is by Srikanth Saripalli', James E Montgomery and Gaurav S. Sukhatme ([6]) from University of Southern California and California Institute of

Technology who presented an algorithm for vision based autonomous landing of a model helicopter in an unstructured 3D environment.
But all the work that has been done, utilized the measurement systems like that of GPS, IMU etc. But we have not used GPS, (and initially not IMU also) rather implemented the whole algorithm just based on vision.

Overview of the Project:

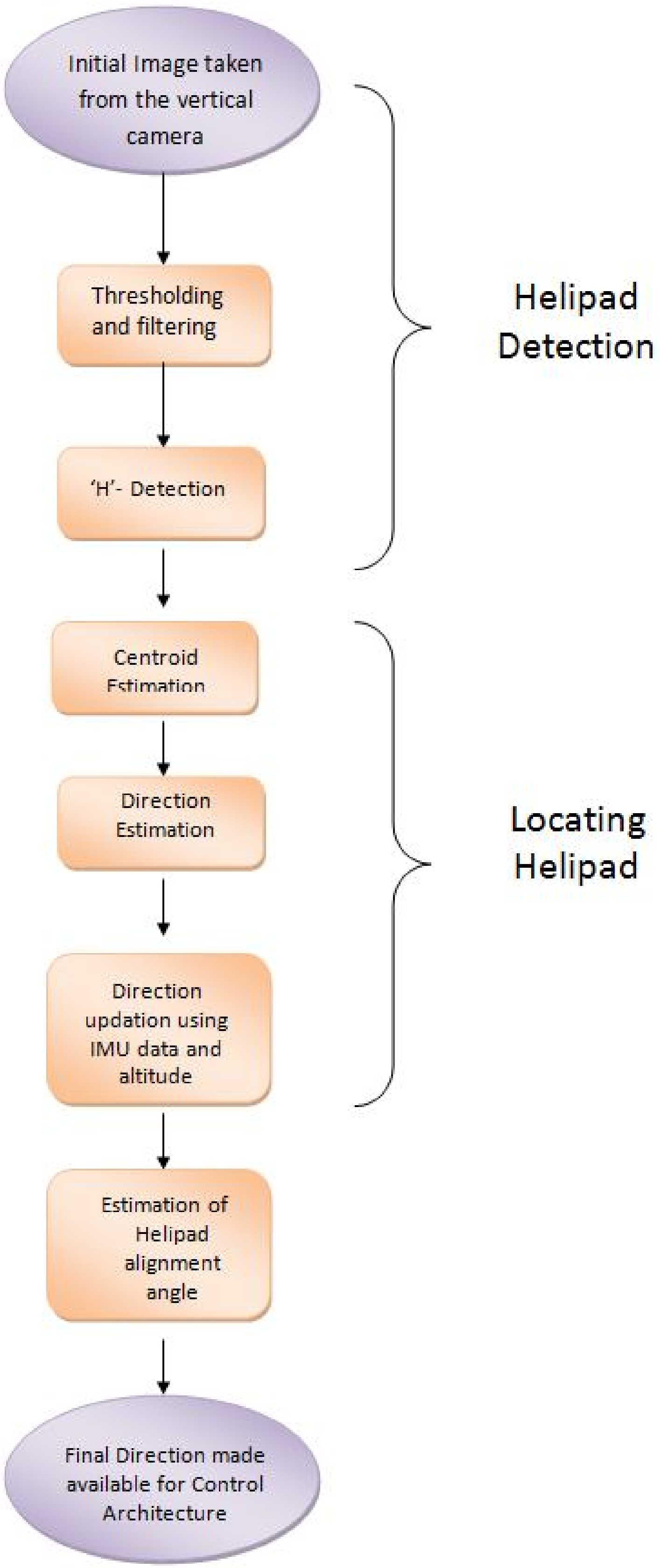
The UAV for which we are designing this whole landing target locating algorithm is the **Parrot AR Drone 2.0.** It is a radio controlled flying quadrotor helicopter [7].
It has the following major features which:

- 30 fps front camera and 60 fps vertical camera
- 3-axis gyroscope
- 3-axis accelerometer
- Ultrasound sensors for ground altitude measurement

We are using the images from the vertical camera for the helipad recognition and the values from the 3-axis gyroscope and the altitude measurement so as to precisely locate the helipad taking into account the effect of pitch, yaw and roll. The following flow chart shows the overall Helipad Locating Strategy used.
It consists of two major parts:

a) Helipad Detection: Done using several image processing techniques
b) Locating Helipad: Direction of the Helipad calculation using the location of H in the image, the IMU data and the altitude value obtained from the ultrasound sensors in the AR DRONE.

OVERALL LANDING TARGET TRACKING STRATEGY



Helipad Detection Using Image Processing:

Our Vision algorithm comprises of the following steps:

1. **Grey Scale Conversion**
   Images were converted from RBG to a Grey Scale using standard function.

2. **Thresholding:**
   Thresholding is used to convert a colour image into a binary one. This is used to minimise the computational cost and increase the efficiency because working on the noisy image obtained from the camera would result in huge computational cost. We chose the threshold level to be at 75% of the maximum grey level so as to get a robust implementation[n].

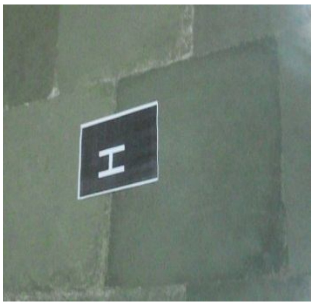
3. **Filtering:**
   For noise removal and edge details preservation, we applied a 7x7 Median Filter to the obtained binary images. Moreover these filters are low pass and remove white noise effectively. This was best for our purpose because, they preserve edge-sharpness and hence apt for detecting and recognising objects with fixed geometrical shape like our Helipad.

4. **Contour formation:**
   After filtering, contour formation of all the connected objects in the image were drawn.

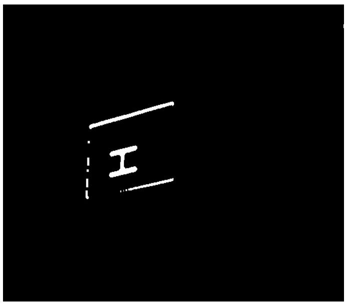
5. **ROI Reduction:**
   Area enclosed by each contour was calculated and based on the sample images, we got a rough range of possible area enclosed by the Helipad Contour and therefore we discarded the objects with contour area beyond that range and thus we could reduce our Region of Interests significantly.

After this, initially we tried to calculate the first, second and third order moments of the objects and by comparing those we were hoping to discard all objects but the Helipad [6].
But eventually, we decided against implementing this because the whole idea of discarding objects on the basis first, second and third order of moments was very much unclear and seemed rather like a hunch. Moreover calculating those was fairly complicated.
And finally we decided to use template matching for the 'H' detection.

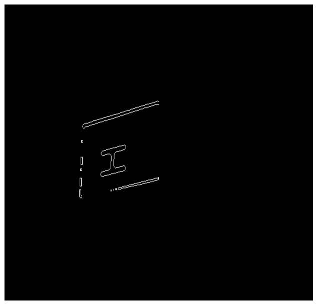
A diagram containing all the images obtained by the step by step process:



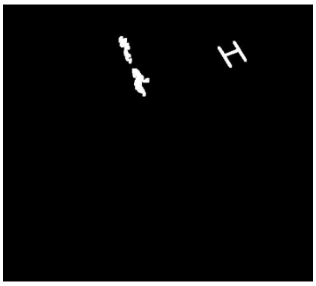
a)Original Image



b) Thresholded and filtered image



c) Images with contours



d) Image with reduced ROI

e) Final Image

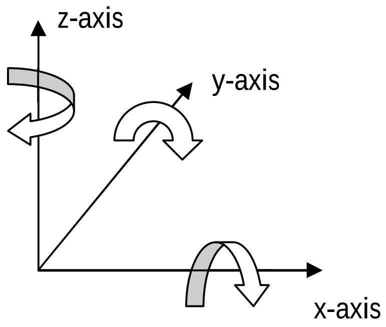
Locating helipad (the landing target) :

1. **Centroid Calculation:**
   Centroid of the image means the centroid of all the objects present in the image. In our case, in most of the frames, the only object left in our final binary image id the symbol "H", so the centroid of the image comes out to be the centroid of the target symbol. The centroid of the image was calculated using the MATLAB function *regionprops().*

2. **Direction Estimation:**
   After calculating the centroid of the object in the image, we need to estimate the direction in which the drone has to move or it has to be given commands to move. For estimating the direction of the target from the present location of the drone or the quadricopter, we have assumed that the plane lies in the centre of the image, as the vertical camera is situated at the centre of the body of the drone.* Now after making this assumption, we have determined the direction in which the drone has to move, as the direction from the centre of the image to the centroid of the object present in the filtered image (which is, in most of the cases, only the symbol "H").

(**Note**:- The '*' assumption was based on the fact that we had not used the IMU (Inertial Measurement Unit) present in the drone, and hence we don't have the information of the tilt angle or the rotation of the plane at the time of estimating the direction. However, if we can use the data provided by the IMU, the changes in the algorithm is stated next).

3. **Direction updating using the data from IMU:**
   For describing the rule of update of the direction, let us first define a coordinate system with reference to our plane i.e. taking the plane as the origin.
   - Coordinate defining- Let us define the coordinate axes as follows:



Here the plane is assumed to be the origin of the system with the plane heading towards positive y-axis. We have defined the rotation angle to be positive along the direction of rotation shown above.

- Updating:-
  Rotation about x-axis is called "*pitch*"
  Rotation about y-axis is called *"roll"*
  Rotation about z-axis is called "*yaw*"

Due to these rotations, the centre of the image shifts and remains no more at the present location of the plane.

Let's denote the pitch angle by θ, the roll angle by ø the yaw angle by α.

If the coordinates of the centre of the image (in image coordinate system i.e. left top corner as (0,0)) be denoted by (x1,y1).

Now if the pitch angle is θ, then the centre of the image shifts down by $\Delta y1$ and the new coordinates of the plane is estimated as:

$$\Delta y1 = h*\tan\theta$$

Therefore,

$$y1' = y1 - \Delta y1$$

y1' denotes the new y coordinate of the plane

h = height of the plane from the ground which can be obtained by using the *ultrasound sensor* provided in the AR Drone

$\Delta y1$ is subtracted because y is calculated from the left top corner.

If the roll angle is ø, then the centre of the image shifts left by $\Delta x1$ and the new coordinates of the plane is estimated as:

$$\Delta x1 = h*\tan ø$$

Therefore,

$$x1' = x1 + \Delta x1$$

$\Delta x1$ is added because x is calculated from the left top corner.

If both pitch and roll is present, then the new coordinates of the plane are given by

$$x1' = x1 + \Delta x1$$

and

$$y1' = y1 - \Delta y1$$

Hence the final direction in which the plane has to go is given by the direction from (x1',y1') to the centroid of the symbolic object "H".

4. **Estimation of alignment angle:**

The alignment angle means the angle by how much the plane should yaw in order to align itself with the "H" symbol of the helipad. This angle is estimated by detecting the edge of the symbol "H", and then taking the slope of the edge. This angle is measured with reference to the positive x-axis and is denoted by ψ. The actual alignment angle is obtained by subtracting 90^0 from the angle ψ.

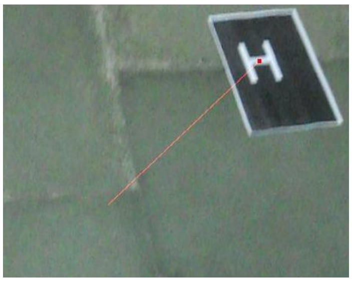
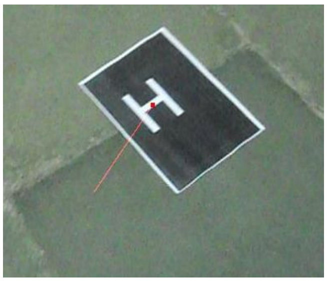
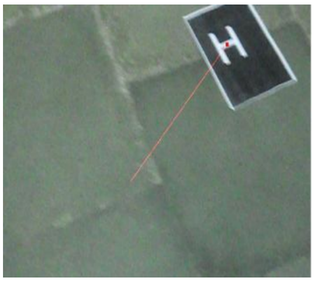
This alignment angle does not change with pitch and roll, but it does change with yaw motion of the plane.

Updated alignment angle = $\psi-\alpha-90^0$

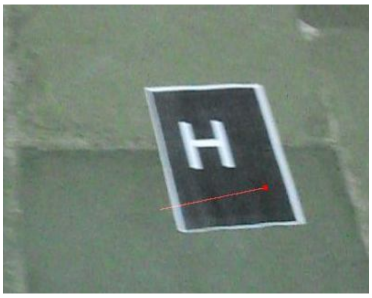
Where ψ is anti-clockwise from the positive x-axis and α is calculated according the coordinate system defined earlier.

RESULTS:-

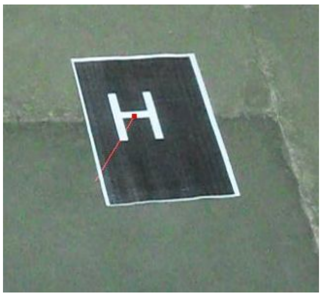
The following are the snaps of the frames of the final processed output video showing the direction of the Helipad with respect to the Helicopter.

It can be seen that Helipad is located precisely despite of rotation and translation of the Helipad with respect ti the aircraft.



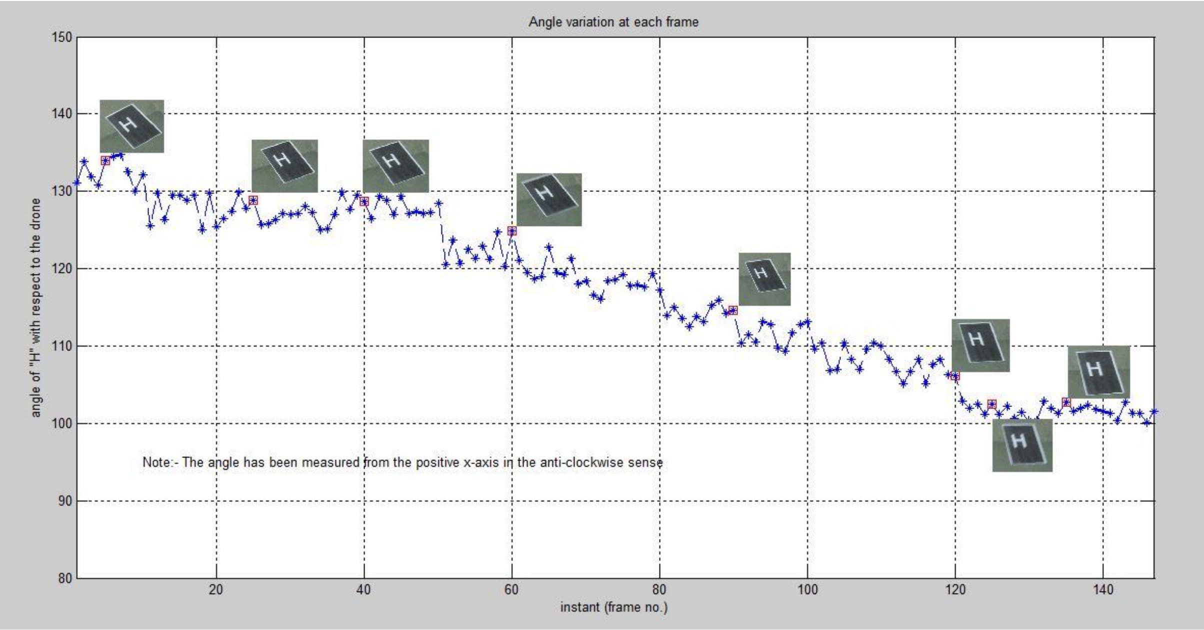
Helipad located incorrectly momentarily



Helipad located correctly again

There were certain instances (though very few), when the Helipad centroid was located incorrectly. But that fault was always momentary and was corrected by the algorithm in the very net moment. Now when one feeds these directions of Helipad to the helicopter in the real-time, the helicopter actually doesn’t get time to responsd to that incorrect direction that quickly and eventually the helicopter ends up going in the right direction.

The following plot shows the variation of Helipad orientation with respect to x-axis at various instances. This gives the alignment angle to the helicopter i.e. the angle by which the helicopter needs to rotate itself so as to perfectly align itself with the helipad.



Alignment angle at every instant

CONCLUSION AND FUTURE IMPROVEMENTS:

We have implemented the basic algorithm and have achieved good results with some small amount of error (approx. 15%-20%) in detection of the helipad. But this error boils down even smaller when the control architecture is implemented, because the fault was only momentary and was corrected in the very next frame. So in the overall process of landing, the efficiency increases significantly.

Apart from this, there are some improvements possible in this work. Some of these are:

- Estimation of actual distance of the target from the drone using relative scaling method being applied on some sample images.
- Passing the information collected to the controller of the drone to implement the motion of the drone according to the estimated target direction and orientation.

If these improvements can be done, then the autonomous landing will be completed successfully.

References:-

0. http://en.wikipedia.org/wiki/Unmanned_aerial_vehicle

1. *Pushing the wrong button: Bad button placement leads to drone crashes* http://arstechnica.com/information-technology/2013/03/pushing-the-wrong-button-bad-button-placement-leads-to-drone-crashes/

2. *Pilotless Drones: Background and Considerations for Congress Regarding Unmanned Aircraft Operations in the National Airspace System* http://www.fas.org/sgp/crs/natsec/R42718.pdf

3. *Courtney* ***S.*** *Sharp, Omid Shakemia, and S.Shankar Sas****try,****"A vision system for landing an unmanned aerial vehicle," in* ***In Proceedings of IEEE International Conference on Robotics and Automation,*** *2001, pp. 1720-1728.*

4. *O.Amidi,* ***An Autonornous Vision-Guided Helicopter,*** *Ph.D. thesis, Robotics Institute, Camegie Mellon University, 1996.*

5. *Ryan Miller, Bemard Mettler, and Omead Amidi,"camegie mellon university's 1997 international aerial Robotics competition entry".*

6. *Vision-based Autonomous Landing of an Unmanned Aerial Vehicle by* ***Srikanth Saripalli', James E Montgomery and Gaurav S. Sukhatme in*** *Proceedings of the 2002 IEEE International Conference on Robotics* ***&*** *Automation Washington, DC May 2002*

7. http://en.wikipedia.org/wiki/Parrot_AR.Drone

8. http://ardrone2.parrot.com/usa/