Department of Electrical
and
Computer Systems Engineering

Technical Report
MECSE-28-2003

Aircraft Approach Angle Estimation: Vision Based Landing

D. Tung, D. Suter and A. Bab-Hadiashar

MONASH
UNIVERSITY

Aircraft Approach Angle Estimation: Vision Based Landing

Abstract

An algorithm for estimating the approach angle of an Unmanned Aerial Vehicle (UAV) is presented. The Algorithm involves extracting the horizon and the focus-of-expansion (from opticflow employing robust statistics). Experimental results are presented to validate this approach.

1. Introduction

In recent years, there has been a lot of interest in the development of an Unmanned Air Vehicle (UAV). Various methods have been proposed to enable UAV's to carry out different tasks autonomously [1-5]. As landing currently requires the services of an experienced pilot (all other phases usually being unmanned), the primary interest of our research is to provide an autonomous landing capability for a UAV using vision alone. Within this context, the current focus of our work has been to find the approach angle.

In general, the landing phase of a human driven aircraft can be categorized as a two-stage process. During the first stage, the pilot picks an *aim point* on the runway as such that theoretically the aircraft will impact on the aim point if no subsequent action is taken. Personal communication with an experienced pilot confirmed that this aim point is a critical parameter for landing. The *approach angle* in this context is defined as the angle between the trajectory of the aircraft and the ground surface. Instruments, such as a glidescope, provide clues to pilots relating to the approach angle margins.

When the descent of aircraft reaches a certain height sufficiently close to the ground, a new aim point is set (usually the end of runway). This brings the landing phrase into the second stage. The position of the aircraft (relative to the runway) at which this change of aim point occurs is known as the *round-out* point. This *round-out* point is also a significant parameter for landing.

2. Computer Vision and UAV's

In the past decade, there have been great advancements in the application of computer vision to UAV's and associated aerial visual surveillance. For example, [6] demonstrated how a high precision ego-motion estimate of a camera can be incorporated to achieve video annotations and insertion to reference imagery.

Many UAV implementations incorporate a Global Satellite Positioning System (GPS) or Inertial Navigation System (INS) for aircraft position and displacement measurements. The feasibility of using vision alone to extract aircraft position is demonstrated in [3]. By matching the real-time video feed with a set of reference IRS images or a Digital Evaluation Map (DEM), absolute position was estimated.

In contrast to GPS-based system, vision-based navigation solely relies on cameras - the aircraft is not guided by external signals and, therefore, is not subject to interference or jamming of signals. Carrying cameras for mission purposes, other than landing, has a large number of applications. Thus, having an on-board camera for landing is no extra cost if that camera and video processing system is already of use for the mission objectives.

3. Problem Formulation

Consider a camera in motion as shown in Fig. 1. The dotted line represents the trajectory of the camera. The bottom line is the ground plane, the camera image plane is shown (extended) in a diagonal direction, the upper horizontal line represents a plane parallel to the ground plane and going through the focal point. At time t_i (i=0,1,2…n), the camera is at position p_i (the focal point of camera). At time t_n,: *FOVC* is the field-of-view-center (also known as the principle point, it is the center position of an image, except for possibly small misalignment in manufacture), *FOE* is the focus-of-expansion (it can be extracted by opticflow calculations), *Horizon* is the vanishing line of the ground plane (in current experiments, it is found by

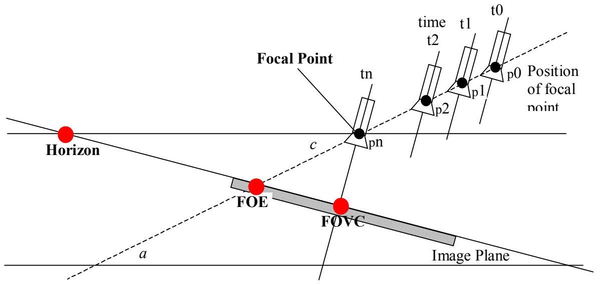

Fig. 1 Geometry of the Approach Angle

extending the projection of parallel lines, lying on the ground plane, on the image plane, until they intersect. Other methods of extraction are being considered). Note that the approach angle (*a)* is equivalent to angle *c*. An expanded version is depicted in Fig. 2, where f_p is the focal length, H is the distance from horizon (in the image) to FOVC, and L is the distance from FOE to FOVC.

Applying simple trigonometry:

$$a = \tan^{-1}\frac{H}{f_p} - \tan^{-1}\frac{L}{f_p} \qquad (1)$$

$$= \tan^{-1}\frac{fp(H-L)}{fp^2 + HL} \qquad (2)$$

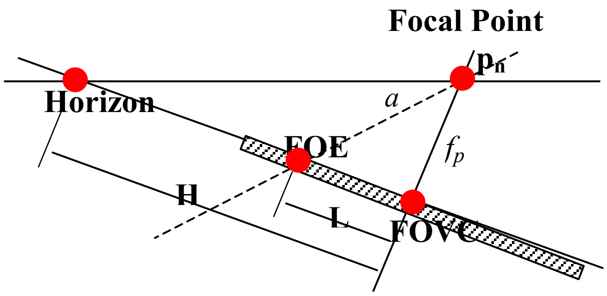

Fig. 2 Approach angle *c* measurements

4. Extracting Parameters

This section details our current implementation to extract the required parameters (f_p, *H* and *L*) for generating an approach angle estimate.

4.1. Extracting parameter: f_p

The focal length value, f_p, in units of pixels can be recovered by camera calibration [7]. A rough approximation can also be found by using the manufacturer's specifications for the physical height of sensor chip H_m (in mm), and focal length F_m (in mm). We are making our measurements in units of pixels so we can convert by simply using the number of pixels from top to bottom of the image (this assumes that the image is projected exactly with the area of the sensor). Specifically, let the image height be S_p (in pixel units), then,

$$\frac{H_m}{F_m} = \frac{S_p}{f_p} \qquad (3)$$

Note: Subscript *p* denotes units of pixel and subscript m represents quantities measured in mm.

4.2. Extracting FOE

The focus-of-expansion (FOE) represents the direction of translational motion and can be derived from the image velocity field - essentially [8], the FOE is the heading point, within the visual field.

If UAV is flying parallel to the ground plane, the FOE should lie on the horizon. If the direction of the motion is not parallel to the ground plane, the FOE will not lie on the vanishing line. This is the key idea behind out method - the displacement of FOE, away from the horizon, is utilized to find the approach angle.

The current implementation extracts FOE using a normalized block based correlation method as described below.

Firstly, two regions-of-interest (ROI) are extracted in each frame. The ROI's consist of nine 8x8 pixels blocks. Because we essentially triangulate from each ROI, an separation (ie: half image frame width) is required. It is also important to ensure that both ROI's are placed on image area that constitutes ground plane movement. Opticflow calculations, based on normalized Sum of Squared Differences (SSD), are carried out on each of these nine blocks inside both ROI. This is shown pictorially in Fig. 3.

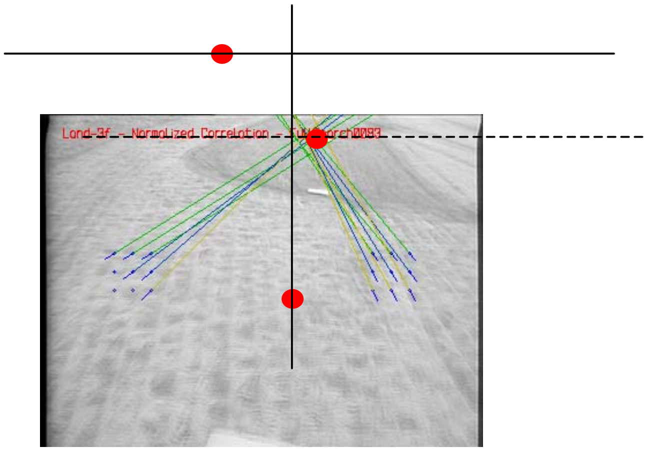

Fig. 3 Extraction of FOE: Red dot at top – point on the horizon, Red dot at bottom – FOVC. Red dot in between is the FOE found by robust estimation of optic flow vector instersections

Opticflow calculations produce a set of motion vectors. Ideally, all lines should intersect on a single point. In practice though, due to noise in the image and false-matches, these extended lines do not intersect at a single point. Thus, the goal is to robustly find a statistical intersection point, starting with a set of line equations. We utilized M-estimator to provide an intra-frame FOE. (In detail, we take the parameters for each line defined by an optical flow vector, and feed these parameters into MATLAB's robustfit function choosing the "fair" kernel parameter.)

In order to further limit the possible effect of false-matches, an inter-frame FOE estimate is developed to augment the intra-frame FOE. The idea is to apply a binary weighted mean (0-weight to those judged to be outliers and 1-weight to those jusdged to be inliers) to a number of consecutive intra-frame FOEs so that if indeed a false-match does corrupt the intra-frame FOE, it could be detected immediately. The procedure, in more detail, is to find the median of the estimates, use the Median Absolute Deviation formula [9] for estimating the scale (standard deviation) of the noise, and then classify as outliers those values that are more than 2.5 standard deviations away from the median. Of course, such a procedure assumes that the FOE is relatively stable over several frames.

For diagnostic purposes it is also useful to return an estimate of the intra-frame reliability. The intra-frame FOE estimate has a patch of eighty optic flow line intersections surrounding it. The current implementations utilize this characteristic to indicate the underlying accuracy of FOE estimation. A small the patch area represents a more accurate underlying FOE estimate.

4.3. Extracting the horizon

In a typical landing scenario, the orientation of an aircraft is aligned with the runway. The aircraft descends slowly until its wheels touch the ground and the horizon is usually visible within the image area if the angle of camera is chosen appropriately. If this is the case, segmentation-based horizon detection methods [10, 11] can be utilized to locate the horizon.

Another method to extract the horizon involves extending two pairs of parallel lines on the ground plane, within the image space, to infinity. For this paper, we adopt such an approach. In our experiments, several objects were placed on the ground to generate parallel lines in the image – in a field we could lay down plastic sheets or take advantage of runway markers and buildings etc.

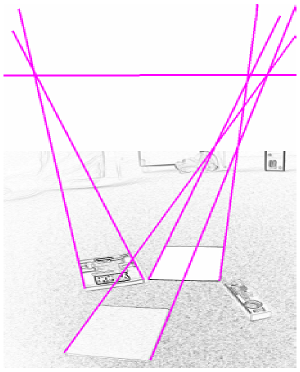

Fig. 4 Horizon detection by parallel lines

An edge image was generated and parallel lines extracted (Hough transform).

Joining various extended lines together formed a series of intersections, which enable us to approximate the horizon position on the image plane. An example of this is shown on Fig. 4.

5. Experimental Setup

In the laboratory, the camera was firmly fixed on the sliding platform and it is allowed to slide along the rails incrementally (point-and-shoot). This simulates a camera on an aircraft in descent. The mounting platform was pre-adjusted to a fixed angle so that the camera was sliding towards the ground at a known specific angle (similar to the dotted line in Fig. 5). This descent angle was measured carefully with a fine protractor and it represented the ground truth for the approach angle.

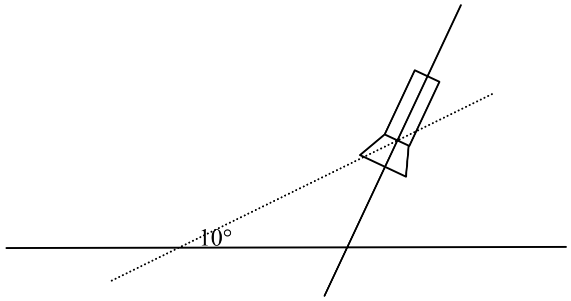

Fig. 5 motion of camera for the experiment

The ground surface was composed of textured carpet. The experiment was carried out inside our lab during the day with sufficient sunlight. In addition, two extra incandescent light sources were utilized to ensure good quality pictures. The sequences were processed offline to extract the parameters required by equation 1 (optimized code should make this real time).

The estimated focal length of the camera used in the experiments, recovered using Zhang camera calibration routines [7] , f_p = 933.5 (pixels)

The measured ground truth and estimated approach angle are illustrated in Table 1.

Ground Truth	Estimated Angle
10	11.1
18	18.0
24	22.0

Table 1 Experimental results

In comparing the approach angle generated by our algorithm with the ground truth, the numerical results revealed that we can recover the approach angle iwth about one or two degrees error - this compares with up to 9 degrees lerror with a naive implementation that does not used robust statistics, as outlined above, to eliminate the effects of outliers. Clearly some form of outlier rejection is necessary.

6. Concluding Remarks

A new method to extract the approach angle of an UAV based solely on vision is presented. The described method extracts three parameters: FOE and Horizon are extracted from clues within the image space, and the focal point, in units of pixels, is derived from camera calibration or approximated from the manufacturer's specs. These three parameters enable us to generate an approach angle estimate.

There are several issues that we plan to address in future work. Though we have analysed actual footage from a UAV landing, as opposed to the lab footage mainly employed here; we do not have ground truth to compare against. We are designing instrumentation to collect such ground truth data. In regards to horizon detection, a possible solution is to utilize an infrared sensor to locate the discontinuity of radiation level between the sky and the ground. Future work will also involves ground speed estimation and round-out point estimation is in progress.

7. References

1. Tierno, J.E., *Distributed autonomous control of concurrent combat tasks.* Proceedings of the American control conference, 2001. **Vol. 1**: p. 37-42.
2. Beard, R.W., T.W. Mclain, and M. Goodrich, *Coordinated target assignment and intercept for unmanned air vehicles.* International conference on Robotics and Automation (ICRA), 2002. **Vol. 3**: p. 2581-2586.
3. Sim, D.G., et al., *Integrated Position Estimation Using Aerial Image Sequences.* IEEE Trans. PAMI, 2002. **Vol. 24**(1): p. 1-18.
4. Hall, C.E., *A Real-Time Linux system for autonomous navigation and flight attitude control of an uninhabited aerial vehicle.* Digital Avionics Systems, 2001. **Vol. 1**: p. 1A1/1 -1A1/9.
5. Shim, D.H., et al., *Multi-functional autopilot design and experiments for rotorcraft-based unmanned aerial vehicles.* Digital Avionics Systems, 2001. **Vol.1**: p. 3C4/1 -3C4/8.
6. Harpreet, K.J., et al., *Annotations of Video by Alignment to Reference Imagery.* Vision Algorithms: theory and practice, 1999: p. 253-263.
7. Zhang, Z., *A Flexible New Technique for Camera Calibration.* 1998, Microsoft Inc: Redmond.
8. Guissin, R. and S. Ullman, *Direct Computation of the Focus of Expansion From Velocity Field Measurements.* IEEE Workshop on Visual Motion, 1991: p. 146-155.
9. Rousseeuw, P.J.a.A.L., *1987.* Robust Regression and outlier detection, New York: John Wiley & Sons.
10. Cozman, F. and E. Krotkov, *Position estimation from outdoor visual landmarks for teleoperation of lunar rovers.* Workshop on Applications of Computer Vision, 1996: p. 156 -161.
11. Scott M, E., et al., *Towards Flight Autonomy: Vision-Based Horizon Detection for Micro Air Vehicles.* Florida Conference on Recent Advances in Robotics (http://www.mil.ufl.edu/publications/)