Measuring Horizon Angle from Video on a Small Unmanned Air Vehicle

Terry D. Cornall and G.K. Egan
Dept. of Electrical and Computer Systems Engineering,
Monash University, Melbourne, Australia
terry.cornall@eng.monash.edu.au

Abstract

This article details the work of the authors towards the goal of using video processing for autonomous flight control of small unmanned aircraft (UAVs). The work reports on procedures that were designed by the authors to determine the roll of an aircraft from video imagery of the horizon, using video and computing equipment small and light enough to be carried by the aircraft.

Theory and results of tests using simulated horizon views are given and discussed. Preliminary results from real flights are also given and discussed.

Keywords: UAV, unmanned aircraft, horizon detection, horizon angle, aircraft attitude, atmosphere, sky, ground, image processing

1 Introduction

Vision processing techniques promise to lend themselves well to many autonomous navigation and control tasks, but the usually high amount of processing that needs to be done requires a reasonably powerful computer, large-scale programmable logic or ASICs. Technological advances and increased sales volumes continue to shrink the size, weight and cost of such computers, but still the electrical power and weight constraints of small unmanned air vehicles (UAVs) militate against complex onboard vision processing systems. The risk of loss and damage to the equipment that comes with the nature of the missions that UAVs may be called upon to perform also requires a low-cost approach. Other work is being done in this field, but because the payload capacity of the airborne platforms are so small the vision processing is often done on the ground, using radio telemetry to retrieve the video. [1, 2]. At least one other research group has developed video horizon measurement equipment that is simple enough that it could be implemented in a way that could be airborne, [3], which uses a thermal imaging camera or scanned linear array. Similar devices sensing in the infra-red spectrum, using a small number of discrete infra-red sensors, are used in UAV and aerospace applications, [4] for stabilising aircraft and satellites. There are also devices such as mechanical, solid-state and optical rate gyros that are used with great success in inertial guidance systems in many aircraft. Many of these devices are too large, heavy and require too much power to be useful in a UAV context, but others are well suited to UAV applications and are being used for such. [5].

This article discusses a method that is different from the previously discussed methods in that it uses visible light on a platform light enough to be airborne by a small UAV. It has been shown to work with reasonable accuracy in simulation. It is small, light and requires low computational power. Although it currently relies on a particular video camera, the method is suitable for implementation using other cameras, given a suitable digital interface. It is not necessarily a replacement for infra-red or inertial guidance systems, but could certainly be useful as an adjunct, possibly to overcome the drift problem that inertial devices suffer from. The method is also suitable for use in the infra-red spectrum.

Other researchers are also developing light-weight, specialised vision sensors for use with UAVs. For example [6], discusses a VLSI optic flow sensor for Micro UAVs (MAVS) and many, including [7, 8, 9] discuss systems inspired by insect vision for use in flight stabilisation, obstacle avoidance and terrain following.

This paper describes a method that takes advantage of a small, light, visible light video camera, the CMUcam developed at Carnegie Mellon University. [10, 11] It is a low-cost CMOS digital output camera that has an embedded microprocessor that does the image capture and some primitive vision processing. The author has added another microprocessor, a Microchip PIC16F876 [12], (hereafter referred to as simply PIC), and software to do the higher level vision processing to apply a method of determining aircraft roll angle using the horizon. The PIC is at the lower end of the computational power spectrum for microprocessors available today, and was chosen for that reason, to demonstrate this aspect of the method. The small size, weight and cost of this equipment means that all of the processing can be done aboard the UAV and the results of the vision processing are in a form that can be simply passed on to appropriate control software to be used for maintaining stable flight.

The horizon angle measurement relies on the contrast between the ground and sky brightness in an image taken by a camera aligned to the longitudinal axis of an aircraft. The image pixels are classified into one of two classes, sky or ground, using a simple thresholding criterion, which in real applications would have to be made much more sophisticated, but for the purposes of a proof of concept has been kept simple for this article. A circular mask is applied to the image, for reasons explained below, and the average coordinate (the *centroid*) of each class is calculated. The angle of the horizon is then determined by taking the perpendicular to the line joining the ground and sky centroids. This gives the roll angle of the camera and hence of the aircraft.

Trials using simulated views have shown that an accuracy of better than 1% is achievable in the measured roll angle, and that the method is inherently able to ignore movements of the camera platform that perturb the position of the horizon in the view, as long as it remains in the view.

2 Theorem, proof and derivation of equations

To analyse the use of centroids for roll and angle derivations, we will consider the horizon image in a circular view. Consider:

Definition 1 *The* sky *class is defined as those pixels that belong to the part if the image that is formed from light coming from the sky.*

Definition 2 *The* ground *class is defined as those pixels that belong to the part if the image that is formed from light coming from the ground.*

Definition 3 *The* centroid *of a class is defined as the average coordinate of those pixels that belong to the class, determined for each axis by taking the sum of the coordinates of each pixel in the class and dividing by the number of pixels in the class.*

Theorem 1 *For a circular viewport, the line joining the centroids of the sky and ground classes will bisect the horizon at a right angle, regardless of the roll angle and of the pitch angle, as long as the horizon makes a straight line in the view.*

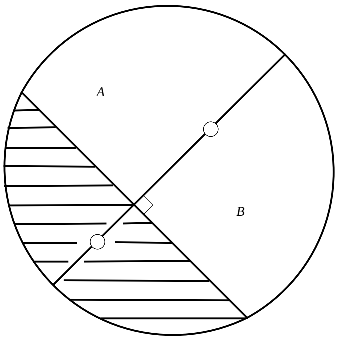


Figure 1: The horizon is perpendicular to the line joining the sky and ground centroids

Consider Proof 1 and figure 1. Arguments of symmetry and an appeal to Euclid's *Elements* regarding the properties of a chord of a circle are used to prove theorem 1.

Proof 1 1. *If the horizon is a straight line, then it forms a* chord *of the circle.*

2. *A line bisecting the chord formed by the horizon and perpendicular to it will pass through the centre of the circle. ([13] Euclid's* Elements *Book III Proposition 3) Call this line the* bisector.

3. *Because it passes through the centre of the circle and is perpendicular to the horizon chord, the bisector divides the sky class into two equal and symmetric areas, with area B being the reflection of area A about the bisector. Similarly with the ground class.*

4. *The average coordinate of area B is therefore in a position that is the reflection of the average coordinate of area A, the reflection being around the line of the bisector.*

5. *The average coordinate of the sky is equal to half the sum of the average coordinates of A and B. This falls halfway on the vector joining them, which is a point on the line of symmetry between them, which is on the bisector. Similarly for the ground class.*

6. *As both the sky and ground centroids fall on the bisector, and it has been shown to be perpendicular to the horizon chord, Theorem 1 is proven.*

The utility of Theorem 1 is that we can use it for finding the angle of the horizon simply by measuring the average coordinates of the sky and ground classes. The method has no dependence on the position of the horizon within the view, only on its angle. The measured angle will not change as the horizon moves with perturbations of the camera platform that do not cause a change in the relative angle between the camera horizontal axis and the horizon angle. In other words, disturbances to the pitch and yaw that are not so extreme as to move the horizon out of the view do not have to be explicitly compensated for in the measurement. This simplifies the implementation of the method considerably.

The measurement task imposes a relatively low computation burden on the vision processing system and, most importantly, does not require a frame buffer as all the operations of classifying the pixels and accumulating the average coordinates are local operations. Even the task of applying a circular mask to the image can be done with no frame buffer. The advantage of this is that a relatively simple vision architecture can achieve the task, and it can be done at a fast rate.

It does of course imply a few things that will not always be so. The horizon will not always be a straight line, nor will it always be in the view. (Note however that the method should work even at altitudes where the curvature of the horizon is noticeable, because the line that joins the points on the horizon where it leaves the view is also a chord of the view circle and the symmetry argument still holds.) The process of classifying pixels into sky and ground classes is not so straightforward as the authors have implied thus far.

From Theorem 1 the line of the horizon is perpendicular to the line between the ground and the sky centroids and therefore the gradient of the horizon should be the inverse of the gradient of the ground-sky line. This results in an equation for m, the gradient of the horizon:

$$m = \frac{(X_S - X_G)}{(Y_S - Y_G)} \tag{1}$$

From Equation 1, the angle ϕ that the horizon makes to the horizontal is:

$$\phi = \arctan(m) = \arctan\left(\frac{(X_S - X_G)}{(Y_S - Y_G)}\right) \tag{2}$$

A feature not previously mentioned is that the system should be able to determine when a measurement is invalid, using the ratio of sky to ground pixels, which should not go out a certain range. This will be useful to detect when the horizon is out of view.

3 Measurements

The experimental setup used to test Theorem 1 was to place the CMUcam in front of a video monitor and to use Matlab software written by the authors to generate an image of a rotating horizon formed between grey ground and pale-blue sky. The CMUcam was connected via a 115200 Baud asynchronous serial connection to the PIC. In turn the PIC was connected via a second serial port to the desktop computer running the Matlab software, which also collected the output of the PIC software. That output consisted of the measured centroid coordinates and the calculated horizon angle, at a rate of about twice per second. The software on the PIC controlled the CMUcam to configure white balance and to set the range of colors that would be classified as sky pixels. The CMUcam generated a binary image as a result and the software on the PIC polled the CMUcam for the image. It was then processed on the PIC to apply the circular mask, calculate the centroids of the sky and the ground classes and finally to calculate the horizon angle via Equation 2. (The arctan function that this necessitates was implemented by a truncated series approximation. This truncation will contribute slightly to the resulting error values.)

To apply the circular mask, it was necessary to do it during the accumulation of the centroid averages on the PIC, to avoid the need for image storage memory. This was achieved by defining the mask in terms of the first pixel that would be accepted per row, easily precalculated for an approximation to a circular mask, and requiring only a small amount of lookup memory. The last pixel address in the row can be calculated by symmetry. This row mask then can be applied on a row-by-row basis. A smoother mask could be used and would improve accuracy at the cost of increased computation. The amount of time it takes to apply the mask and accumulate the average coordinates for the sky and ground classes is $40\mu s$ per 8 bit chunk on the 20 MHz PIC. This was achieved using assembler language programming for the critical inner loop of the pixel decoder. A faster microprocessor with a good optimising C compiler could do it all in C.

Figures 2 and 3 show the angles and centroid trajectories during a 360° roll manoeuver where a random *jitter* of up to ±100 pixels was applied to the synthetic horizon image to simulate disturbances of up to approximately ±10° of pitch and yaw, which was as much as could be applied without having the horizon leave the view. Figure 3 shows dramatically the amount by which the centroid positions were disturbed. However, figure 2 shows by the small RMS error of 3.5°, just how little effect the disturbances had on the roll angle measurement. Similar measurements with no pitch and yaw perturbations had similar RMS errors. Note that the RMS error of 3.9° is close to 1% of the full range of 360°.

4 Segmentation

4.1 Color of the Sky

A vertical profile through an image of a clear blue sky is shown in Figure 4, along with an image showing the classification of the pixels into sky (white) and ground (black) by a simple threshold on the blue component. Some clear features can be seen from the results. In the original image the demarkation between ground and

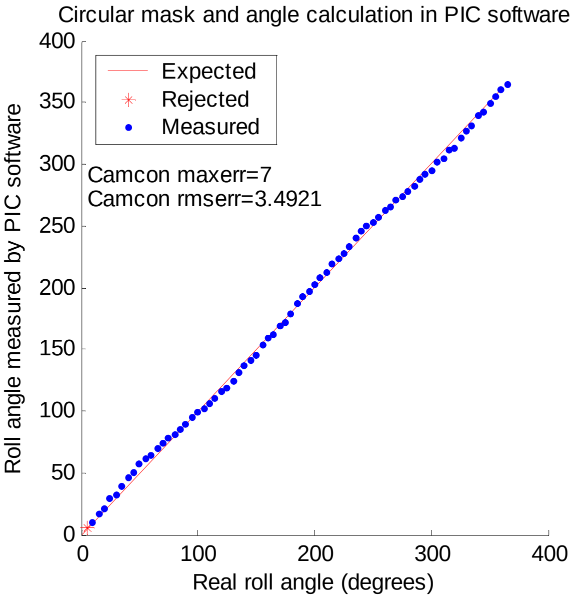


Figure 2: Measured angle versus real angle for roll with turbulence. Circular mask and angle calculated on PIC microcontroller.

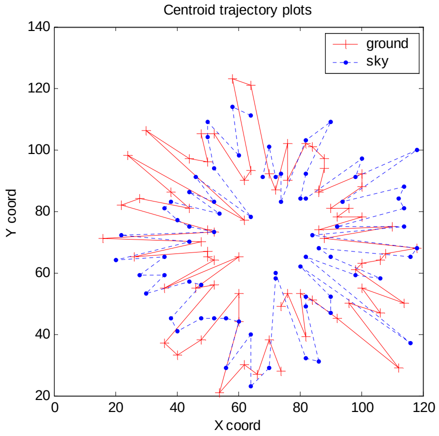


Figure 3: Ground and sky centroids for roll with simulated turbulence. Circular mask applied on PIC microcontroller.

sky is sharp for all the luminance, red, green and blue profiles (only blue is shown for brevity). At the bottom of the image the profile shows a mainly low value, with a lot of texture and a few spikes as sunlit objects on the ground are encountered, then there is a sudden increase as the horizon is crossed. In the clear sky case, the difference between the ground and sky values is more pronounced in the blue profile than they are in the luminance, red or green profiles. This is of course not a surprising observation, given the everyday phenomenon of a clear blue sky. The profile (not shown) for a cloudy image shows the same features as the clear sky image, though the values are lower, the difference between ground and cloudy sky less pronounced, and the downward trend with elevation is also less pronounced. The downward trend with elevation results in the higher elevations of the sky being as low in value as the ground in the luminance red and green profiles in the clear sky cases. In cloudy sky cases the blue profile falls more slowly than the others and remains well above the ground value. Paradoxically, this means that an overcast sky is a better candidate for simple thresholding on the blue component than a clear blue sky is. It is not hard to see that there could be a problem as the zenith is approached in a clear sky. As shown in the thresholded sub-image the zenith has been misclassified as ground. To overcome this it would be necessary to lower the threshold and use other information such as texture as discussed below, to eliminate bright ground pixels.

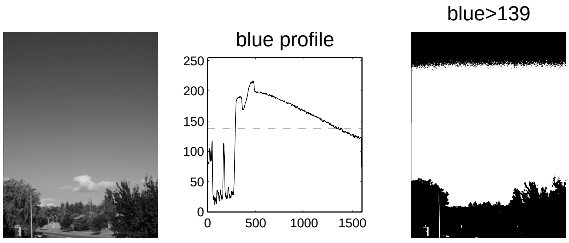


Figure 4: Blue profile and segmentation by threshold on B component of clear blue sky.

There are certainly some circumstances where the demarkation between ground and sky is not so clear, or where it is due to other image features that could be mistaken for a horizon. An example is shown in Figure 5 where the brightness of snow has a confounding effect. Any similar brightly reflecting terrain would cause similar problems.

Based on these observations a simple view is initially taken, that in many cases, the sky is brighter and bluer than the ground. This conclusion is similar to the observations of how some insects use the properties of the light in the environment. For example [9] discusses how the ocelli of a dragonfly might enable the difference in UV light levels between the ground and the sky for to be used for orientation during flight.

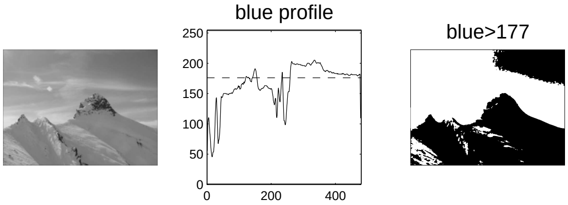
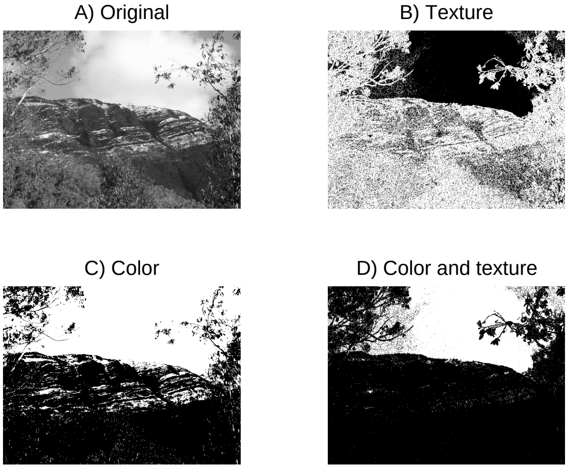


Figure 5: Blue profile and segmentation by threshold on B component sky and snowy ground. Photo with permission from Loren Campbell.

The authors are aware that problems will be caused by reflections of the sky from water which will not be able to be discriminated against on the basis of color, nor on the basis of brightness and there are also the case of fog or dust (high turbidity) where light is scattered by larger particles and visibility can be affected to the point that all directions appear of equal brightness.

4.2 Texture of the ground

It is observed that the features that can be found in the sky are usually low in texture, by comparison to the features on the ground, at least. In [14] the authors of that article discuss their work on processing video images sent by a radio link from a UAV. They give an approach to modelling the sky and ground using texture as well as color. The authors of this present article are investigating a related, albeit simpler approach, constrained as they are by the need to do the processing onboard the UAV. If it were possible to calculate the differences between pixels in the video scan-line without adding delay, this would give a measure of the texture in the image. With that in mind, a simple test was carried out to see if this could be used, along with color, as a discriminant between the sky and the ground parts of an image. Figure 6 shows the result of a preliminary test. In this test, the luminance of each pixel of A), the original image, is subtracted from its predecessor in the scan line, and the pixel positions with a difference greater than some threshold are tentatively classified as ground (B). The original image is also tested for pixels whose blue component is greater than a threshold and these pixel positions are tentatively classified as sky (C.) The logical combination of these two results is used to form a segmentation of the image into sky and ground parts, (D) where the sky is indicated by white and the ground by dark. Comparison of D) and C) shows convincingly that the use of color and texture is superior to the use of color alone, at least in the case of snow-strewn rocks.


A) Original

B) Texture

C) Color

D) Color and texture

Figure 6: Color and texture used for segmentation.

In order to carry out this texture processing task aboard the UAV, a new camera will be used. The second version of the CMUcam has the ability to calculate the pixel differences on the fly, and as soon as it can be integrated into the test vehicle and the appropriate software written, this method will be tested in flight.

5 Flight

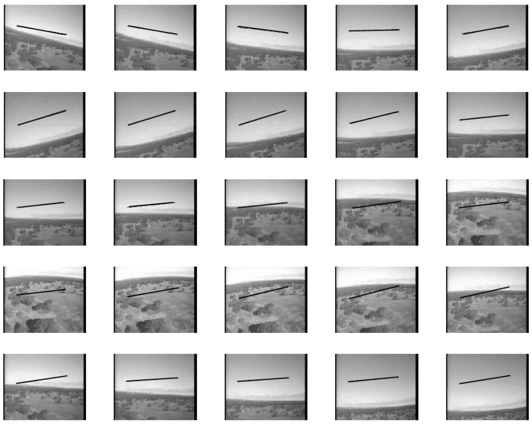


Figure 7: Measured horizon angles.

Between July and October 2004 a number of trial flights took place with the horizon sensor equipment, consisting of the CMUcam and the PIC microcontroller mounted in a small glider. The measured angles were transmitted to the ground station by a data telemetry radio link. The glider also carried an analog video transmitter with a separate camera to transmit video to be used to verify the automated measurements. The glider was controlled from the ground by a radio link. Figure 7 shows 5 seconds of video frames from one trial in fair conditions. It has been overlaid with an artificial horizon indicator showing the roll angle that was calculated onboard the aircraft and transmitted to the recorder at the ground station. The horizon displacement measurement is not indicated. As you can see, the measured angle roughly follows the visible angle, with some inaccuracies and some latency in the response of the sensor to the change in the video. The data from these trials is still being evaluated, but they have shown that, on the whole, the experimental equipment functions well, and also allowed the authors a number of important preliminary observations about the reliability and accuracy of the horizon angle measurement method as described. This trial only used color and brightness in the sky/ground segmentation and the authors have noted that in general it works as expected, in that the artificial horizon indicator generally follows the angle of the horizon in the video. However, there were a number of times when dark clouds were misclassified as ground. This wasn't completely unexpected, and the use of texture as described above should allow for an improvement. It was also noticed that it is difficult to synchronise the video and the telemetry data during post-processing. The current method uses two separate radio links and two separate recording devices, one for the data and one for the video. The equipment will be re-designed to overcome this problem. This is an important consideration, because the degree of lag in the angle measurement would affect the performance of a feedback control system greatly, and must be well-known.

6 Conclusions and future work

The sky/ground segmentation used in this method needs to be improved with the use of texture measurements. The telemetry equipment used in the verification trials also needs to be improved to facilitate comparison between the automated angle measurements and those made by a human observer. Comparison with other automated measurement systems such as [14] and [1] will be made. Although the technique has not been described herein, the authors have developed a means to also make pitch measurements using this method, given the altitude, and this will also be tested and verified in a similar manner. Eventually the authors plan to use the onboard measurements of roll and pitch to automatically stabilise the aircraft.

The above challenges and suggested improvements not withstanding, the method described in this article promises to be a useful computer vision tool that can be brought to bear on some of the problems of autonomous flight.

References

[1] S Ettinger, P. Ifju, and M. C. Nechyba. Vision-guided flight for micro air vehicles. URL reference: `http://mil.ufl.edu/~nechyba/mav/index.html#vision1`, September 2002.

[2] S. Ettinger, M. Nechyba, P. Ifju, and M. Waszak. Vision-guided flight stability and control for micro air vehicles. In *International Conference on Intelligent Robots and Systems.*, volume 3, page 2134 to 2140. IEEE/RSJ, Sept 2002.

[3] Luc Fety, Michel Terre, and Xavier Noreve. Image processing for the detection of the horizon and device for the implemention thereof. United States Patent 5,214,720, Thompson TRT Defense, July 1991.

[4] B. Taylor, C. Bil, S. Watkins, and G. Egan. Horizon sensing attitude stabilisation: A vmc autopilot. 18th International UAV Systems Conference, Bristol, UK, 2003.

[5] Trammell Hudson. Autopilot: Uav command and control. http://sourceforge.net/projects/autopilot/, April 2004.

[6] G. Barrows. Mixed-mode vlsi optic flow sensors for micro air vehicles. URL reference: `ftp://ftp.centeye.com/pub/Dissertation.pdf`. PhD thesis.

[7] G. Barrows, J. S. Chahl, and M. V. Srinivasan. Biomimetic visual sensing and flight control. April 2002.

[8] T. Netter and N. Franceschini. A robotic aircraft that follows terrain using a neuromorphic eye. October 2002.

[9] G. Stange, S. Stowe, J.S. Chahl, and A. Massaro. Anisotropic imaging in the dragonfly median ocellus: a matched filter for horizon detection. *Journal of Comparitive Physiology A.*, 188:455 to 467, 2002.

[10] Anthony Rowe, Chuck Rosenberg, and Illah Nourbakhsh. A low cost embedded color vision system. In *Proceedings of IROS 2002*, 2002.

[11] Carnegie Mellon University Robotics Institute. Cmucam vision system. URL reference: `http://www-2.cs.cmu.edu/~cmucam/`.

[12] Microchip. Pic16f87xa data sheet. Technical report, 2003.

[13] Sir Thomas Little Heath. *The Thirteen Books of the Elements (Euclid, Vol. 2–Books III-IX)*, volume 2 of *3*. Dover, New York, second edition, 1956.

[14] S. Todorovic, M. C. Nechyba, and P. G. Ifju. Sky/ground modeling for autonomous mav flight. May 2003.