3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Measuring Horizon Angle on a Small Unmanned Air Vehicle using Digital Video Camera and an FPGA

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

Abstract

Describes a method to extract aircraft pose from visible-spectrum video of the horizon. Presents an implementation based on a Field Programmable Gate Array (FPGA). Discusses some methods of numerically calculating measures of confidence for the results. Relates the results of real-flight tests comparing performance to an alternative system that uses infrared light.

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 limited payload capacity of many Unmanned Aircraft applications is at odds with the high amount of processing that is often needed. Technological advances and increased sales volumes continue to shrink the size, weight and cost of such processors, but still the electrical power and weight constraints 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, (especially for PhD students). Work carried out by others in this field has often made use of bidirectional radio links to do the video processing on the ground and send back control signals. [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] [5] [6] 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 some low-cost solid-state inertial measurements systems are well suited to UAV applications and are being used for such. [7].

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 carried by a small UAV. It has been shown to work with reasonable accuracy in simulation and in open-loop flight. It is small, light and requires low computational power and has been through a number of implementations. In 2004 at ICARA version based on a CMUCAM [8] and a PIC microcontroller was presented. [9] The subject of this article is an improved version based on an FPGA coupled with a digital video camera.

Other researchers have also developed light-weight, specialised visible-light sensors for use with UAVs. For example [10], discusses a VLSI optic flow sensor for Micro UAVs (MAVS) and many, including [11, 12, 13] discuss systems inspired by insect vision for use in flight stabilisation, obstacle avoidance and terrain following.

This article briefly discusses a method to measure the horizon angle using the visible light spectrum and a method that has been developed that calculates a measure of confidence in the result. More detail on the background to these issues can be found in previous articles by the authors [14], [15], [9] and the Australian patent document [16], also by one of the authors, Cornall. The method has been proven in simulation, [14], [15] and more recently in real-flight and the real-flight results will be discussed in depth here.

2 Method

For an image of the horizon in a circular viewport that has been segmented into sky and ground classes the line joining the centroids of the sky and ground classes will bisect the horizon at a right angle. This is true, regardless of the roll angle and of the pitch angle, as long as the horizon makes a straight line in the view,

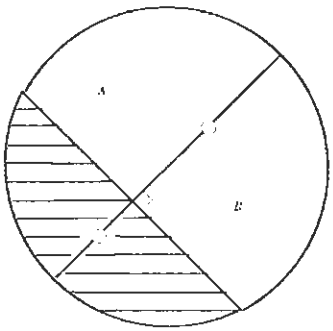

Figure 1: The horizon is perpendicular to the line joining the sky and ground centroids

approximating a chord of the circular viewport. The proof and development of this idea is detailed in [14], [15], [9] and [16]. Figure 1 illustrates the idea, showing the centroids of the classes as small circles joined by a line, the bisector. Because of the circular viewport, the average line of the horizon will cut this bisector at a right angle.

This means that we can find the angle of the horizon simply by measuring the average coordinates of the sky and ground classes to find the centroids. 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 computational 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 using algorithms of O(N) where N is the number of pixels in the image.

3 Segmentation and Validity

It must be emphasised that *any* method that assumes that the pixel values can be classified into two classes may produce false results when applied to an image that does in fact not contain a horizon. Also, there will be situations where even though a horizon is present, other circumstances, such as dark clouds, or bright ground cover, will produce false results. Whatever segmentation method is used, a measure that can be used to reliably detect unreliable segmentation is needed.

3.1 Otsu's Histogram Analysis

Otsu's algorithm [17] works on a one-dimensional histogram to produce a threshold value that segments the histogram values into two classes in a manner that can be shown to be optimal (in a mean squared error sense) for class separation. For each 'bin' in the histogram it calculates the value of a metric σ_b^2 (equation 1) that would be generated if that level was used as a threshold to segment the image into two classes.

$$\sigma_b^2 = (u_2 - u_1)^2 (\omega_1 \omega_2) \tag{1}$$

Selecting the threshold value that gives the best value of σ_b^2 is a means of segmenting the image into two classes. For image segmentation, this method can be applied to a value that is dependent upon the pixel values of the image in order to segment them into two classes. In the case of this thesis, segmentation into sky and ground classes is desired. In equation 1 u_1 and u_2 are the mean values of the measured variable for each class where ω_1 and ω_2 are the number of pixels in or the probability of belonging to the respective class. The first term in the righthand side of 1 maximises class separation and the second term tends to equalise the size of the classes because is a maximum when $\omega_1 = \omega_2$. The metric σ_b^2 is a measure of the variance between the classes [17]. Over all, maximising σ_b^2 maximises the difference between the values of the pixels belonging to the different classes [17]. The threshold that generates the greatest value of σ_b^2 is the level that creates classes with the largest interclass variance [17]. In general, the higher the value of σ_b^2, the better the segmentation is for the purposes of discerning sky and ground. (This is not an absolute rule however, as false horizons can also generate a high value.) By observing the segmented video and corresponding values of σ_b^2, the author has determined that a low value of σ_b^2 is a good indicator of unreliable segmentation. [14]

Unfortunately, as observed during testing, and as discussed in [14], there are cases of failed classification that are not well indicated by the σ_b^2. This indicates that the σ_b^2 metric by itself is not an adequate indicator of reliability. This is not surprising, as the metric does not take the spatial grouping of the pixels into account at all, merely their blue value, so in marginal cases misclassification will occur undetected.

3.2 Metric2.5

A new metric has been developed by one of the authors (Cornall) to measure the success of a segmentation of an image into sky and ground. Metric2.5 (or M2p5) is measured using the statistics of the classified pixels in a manner similar to Otsu's σ_b^2 but operating on their spatial coordinates, not their pixel values. It combines the mean *spatial* coordinates u_1 and u_2 of the two classes

(rather than the mean histogram value as per Otsu) with the populations n_1 and n_2 of each class and the radius R of the circular viewport. The formula for metric2.5 is very similar to that used by Otsu's algorithm and the method could be seen as an extension of that work. The major differences are that it is applied to a two-dimensional variable, the spatial coordinate, and that the product of population terms is de-emphasised:

$$m2p5 = \frac{(u_1 - u_2)^2 (n_1 n_2)^{1/3}}{3R^3} \quad (2)$$

In Eqn. 2 the exponent of 1/3 applied to the product of the populations is to reduce its weight compared to the separation of the classes given by $(u_1 - u_2)^2$. The $3R^3$ in the denominator is a normalising factor to bring the value down to near 1. The effect of the population product is to increase the metric in favour of classes that have similar sized populations as the product $n_1 n_2$ is a maximum when the populations are the same, because $n_1 + n_2$ is a constant that depends on the radius of the viewport. The term $(u_1 - u_2)^2$ containing the class average coordinates, or centroids, u_1 and u_2, increases as the class separation increases, which favours segmentations that have the sky and ground classes well separated in space, which is clearly desirable because any segmentation wherein the sky contains ground is likely to be suspect. This term could favor segmentations where there is one large class with a centrally located average coordinate and one small class with its average coordinate right on the circumference, but the term $n_1 n_2$ containing the populations discourages this by decreasing as the class sizes become dissimilar. For a constant viewport size, the $3R^3$ is a constant. Again, it is possible to find frames where misclassified pixels are not indicated by a low value for metric2.5. It can be that the segmented classes are distinct and similarly sized so metric2.5 has a relatively high value, even though the pixels classified as ground are in fact due to dark clouds, for example. Note however that in this case σ_b^2 could have a relatively low value, although this is not guaranteed. The combined use of these two metrics will be better than either alone.

3.3 Other Measures

Both metrics, σ_b^2 and M2p5, can inform about the quality of the partition into sky and ground and the validity of the straight-line horizon model. Thresholds can be set that define acceptable limits for the metrics which trials have shown are adequate for a wide range of lighting conditions.

Another two measurements that can be of use in deciding if an image or part of an image is suitable for the purpose of detecting the horizon. These are the average brightness of the image, and the average contrast of the image. It would also be possible to use the average brightness of the ground class and the average brightness of the sky class, as well as contrast measures applied to these parts of the image. In a way, the σ_b^2 and M2p5 metrics do take these aspects into account, but the author sees merit in considering overall image brightness and contrast explicitly as a means of being able to accept or reject the results of the image segmentation.

The difficulty arises when trying to decide what the thresholds for acceptability should be. An absolute threshold can be reasonably applied to σ_b^2 and to M2p5 on the basis that below a certain absolute level these measures indicate that the image is unsuitable for segmentation into two distinct classes based on Otsu's histogram analysis method, or that the result of that partitioning does not produce a binary image that conforms to the straight-line model of the horizon.

Brightness, on the other hand, depend strongly on the lighting conditions and the camera settings and orientation that prevail at the time the image is captured. A level of brightness measured directly from the image will be affected by the camera auto-gain and auto-exposure settings, for example, if they are enabled. Disabling these adaptive aspects of the camera is not feasible, because it restricts the ability of the camera to cope with the conditions of lighting that will routinely be met, resulting in the image being too dark, or too over-exposed at times. This means that we need a measure of brightness that takes the camera settings into account. The brightness metric is measured by reading the camera's Automatic Exposure Control (AEC) and Automatic Gain Control (AGC) settings. It was found that a useable measure of brightness of the image can be formulated by multiplying the average brightness $3B^2/(R+G+B)$ as measured from the image by $1/(AGC*2 + AEC + 1)$.

If this measure of brightness is very low, it indicates that the amount of ambient light is insufficient for the image to be useful. Furthermore, by deciding appropriate ranges of brightness, a crude determination of whether the ground only is in view, or whether the image is bright enough to contain at least part of the sky. This is very useful in being able to reject image partitions that can produce false horizons from features on the ground. The brightness measure may also be suitable to determine if the camera is pointing only at the sky, but given the propensity for the high clear sky to drop in brightness it may be difficult to tell the difference between ground and sky in some cases.

4 Horizon Sensor Hardware

An ALTERA Cyclone I EP1C6T144C6 field programmable gate array (FPGA) was used to implement a 'soft' embedded NIOS II microcontroller from Altera and an interface to the digital camera. See figure 2 for a photo of the circuit-boards developed by

the authors. (The side shown carries the FPGA and the other side carries a video I/O circuit). The 'firmware' design is a mixture of VHDL and C software. The size and weight of the FPGA version is compatible with small UAVs, as required. (Weight of $100g$ including camera, size of $100x50x30mm$.) The horizon angle measurements are carried out in software written for the microcontroller (CPU) in the FPGA. The CPU passes the results of the measurements via a serial port to telemetry systems in flight or to software running on a PC during simulations. During closed-loop simulation tests software running on the PC can also implement a PID controller to send control adjustments to a flight simulator. The reason for using the FPGA to interface to the camera rather than the CMUCAM was primarily to make it possible to access the processed video so that it can be transmitted during real flight by a video telemetry link, which was something that the CMUCAM based system couldn't easily be modified to do at a fast enough frame rate. This feature is deemed imperative by the authors as a means to monitor the success or otherwise of the algorithm. The output video can be displayed as a binary image showing the ground and sky classes, or the raw image can be left and the determined horizon angle indicated. Even real-time advice can be given in the form of directional arrows showing the pilot the direction to move the control sticks in to restore horizontal flight. (See figure 4 for example.) The output video makes it possible to tell when the classification method is not successful and to provide evidence for why such failures occurred and what effect parameter adjustments have on the success rate. Statistics and frame number can also be superimposed on the video, making synchronisation of data to video simple. The use of the FPGA also enables us to increase the frame transfer and processing rate and give access to raw image data rather than relying on the built-in functionality of the CMUCAM to do the image processing. The use of the FPGA meant that the camera interface and image masking can be done in hardware implemented outside the CPU (but still in the FPGA) and so doesn't have any negative impact on the processing. Also the frame transfer to the CPU memory space is implemented using DMA techniques in real-time and so there is shorter delay before the frame processing takes place. Eventually the time taken by the CPU for frame processing was reduced to 200 ms giving a rate of five frames per second. This is still not completely satisfactory and the authors have plans to improve it to the point where the processing can run at 20 ms per frame, which is as fast as the camera can deliver them. This will be done by doing the image processing using a pipelined series of hardware processing stages described using VHDL and implemented on a larger FPGA, allowing image processing stages to run concurrently in hardware instead of sequentially in software. The current version of the FPGA used does not have capacity enough to implement both the CPU and the pipelined image processing hardware, which is why this faster implementation has not already been carried out. (90% of the Cyclone FPGA logic resources and 70% of its memory elements were used in implementing the CPU plus camera interface.) Use of a CPU in conjunction with the image processing pipeline is still considered the most flexible way to act on the results of the image processing, as well as for supervisory and configurational tasks, such as setting up the digital video camera, communicating results or control signals to other parts of the system and so on. Work described in [18] explains how the image processing pipeline can be implemented.

The system was also used to compare the video horizon sensor measurements to those of an infrared sensor similar to that described in [6], which has been used successfully by its developer, Professor Greg Egan, for autonomous flight of UAVs at Monash Uni. The results of these measurements are discussed in the section 5.

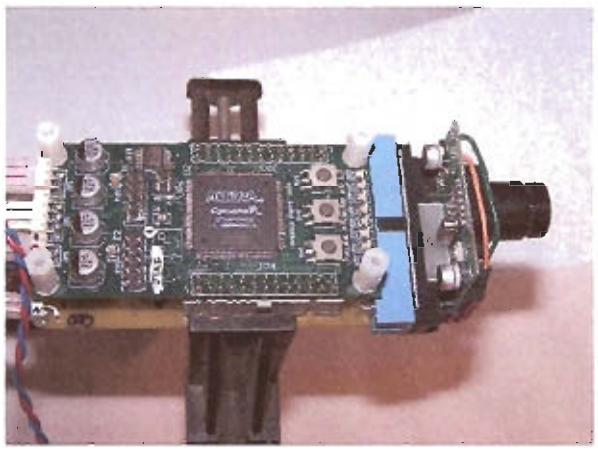

Figure 2: Horizon sensor based on digital video module and FPGA.

5 Flight Measurements

Simulation results have been carried out, and reported in [15], showing that this system is capable of relatively accurate measurements (3% error) of the roll and pitch angle. More recently, tests were conducted during real flight to compare the horizon sensor with a commercially available infrared sensor [4]. Note that both the horizon sensor and the infrared sensor were calculating the angles onboard during flight. The results are not from post-processing or processing of video at the ground-station. The results of one such test are shown in figure 3. It was late in a winter afternoon, with an overcast day, just before a storm, so conditions were not ideal. Note however the high correlation of 0.9 between the two sets of data. Note also that there is

substantial disagreement in places, such as near sample number 629 where the infrared sensor result disagrees with the -46° measurement of the horizon sensor. The data shown are only for when the horizon segmentation metrics indicate a high confidence in the result so the discrepancy is not due to poor view of the horizon.

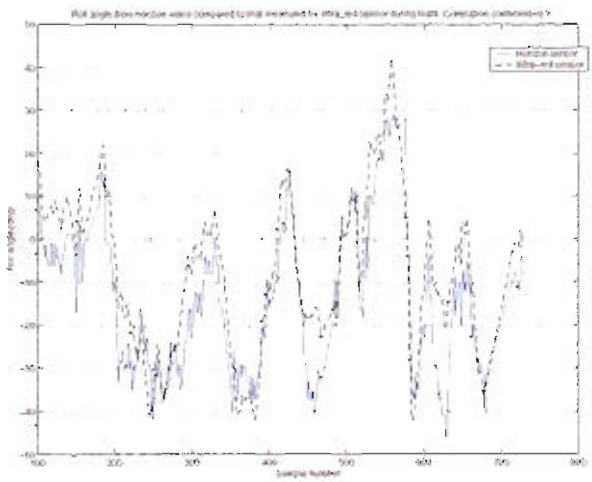

Figure 3: Horizon sensor v infrared sensor for roll angle.

Investigation of the video telemetry, one frame of which is shown in figure 4 showed that the horizon angle does in fact reach -46° at sample number 629 which indicates that the horizon sensor is in fact substantially correct for that section. It appears that the infrared sensor is giving a low roll reading at that point in the flight. It is thought that some environmental feature such as a dark cloud might have caused the anomalous infrared reading. Anecdotal reports from other users of the infrared sensor indicate that this sort of effect has been noticed before.

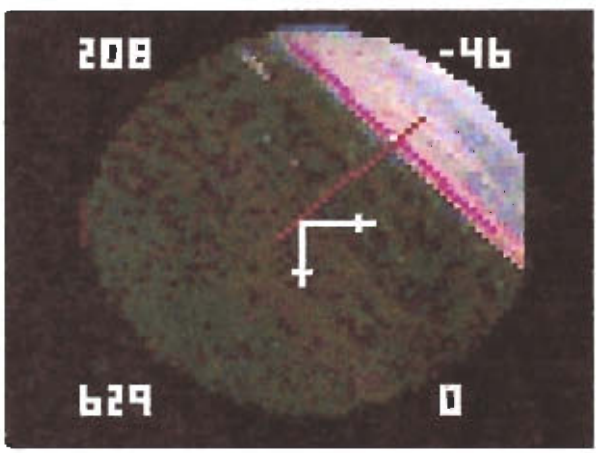

Figure 4: Snapshot of frame 629 showing correct horizon sensor reading.

Another observation came from the flight tests that is encouraging, and that is that the M2P5 confidence metric worked well to eliminate false readings due to objects on the ground that were 'sky-coloured'. In particular, there was a medium-sized freshwater dam in the view at a number of times, and it was reflecting the sky. During those frames, M2P5 was below the acceptable threshold of 200 because the segmentation resulted in a ground class with a large area of 'sky' embedded within it and these frames were consistently rejected.

Yet another observation, not so encouraging, was that in the launch or landing approaches during some trials when the horizon was out of view and only ground was visible, 'false positive' results were given when horizons were erroneously detected in ground features. The efforts, using M2P5 and σ_b^2 and brightness and contrast measures to defeat this problem were not completely sufficient. Although this did only happen in 2 out of the approximately 1000 frames of the trial that produced figure 5 for example, nonetheless, control actions based on these erroneous results would probably have resulted in crashes. This emphasises the notion that the output of the horizon sensor must be further processed to check it for conformity with previous measurements before action is taken.

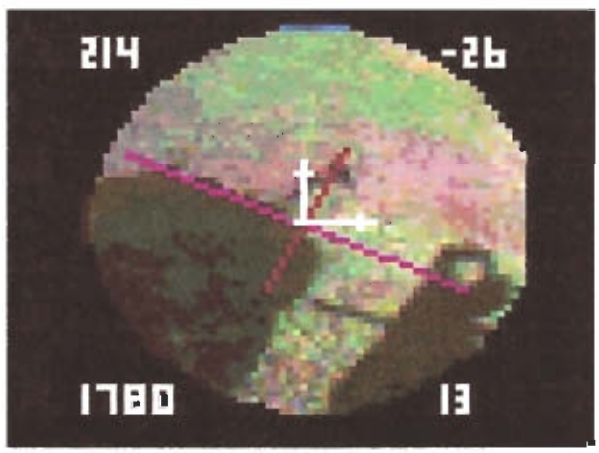

Figure 5: False positive on launch.

The discussion has focussed on roll measurements, partly because of space concerns, but also because during tests it became noticeable that the pitch angle varies considerably faster than the roll angle. It also seems that the 200ms sampling rate is too slow for the pitch variation as in many cases there are only as few as five measurements between peaks in the pitch. A rule of thumb mentioned in [19] is that the sampling frequency for a digital control system should be 30 times faster or more than the highest frequency present in the output of the controlled plant. It is clear that 200ms sampling rate is not adequate for the longitudinal (pitch) characteristics of this aircraft. Nonetheless, measurements of the pitch angle correlates very well in more slowly varying tests done on the ground and in simulation and it is expected that a 20ms frame rate will enable adequate performance in

both the the pitch and roll control of future closed-loop experiments.

6 Conclusions and Future Work

It is the authors' contention that it will be impossible to reject all false horizons on the basis of the visual appearance of the image, because it will always be possible to get images that look like horizons even when they are not, no matter what segmentation method is used. It is clear, however, from images such as 5 that the relatively simple method of segmentation used in this work can be improved, and in fact more complex methods have been reported that appear to work very well. Nonetheless, there is a place for simple methods that work adequately, especially if there are also measurable values such as the M2P5 and σ_b^2 metrics that can be used to qualify the results. Future directions, as mentioned earlier, will be to increase the frame rate to the maximum possible and to use temporal filtering to reject those false measured angles due to the few false horizons that do get past the confidence metrics.

7 References

[1] S Ettinger, P. Ifju and M. C. Nechyba, "Vision-guided flight for micro air vehicles", http://mil.ufl.edu/ nechyba/mav/index.html#vision1, visited on 27 October 2006.

[2] S. Ettinger, M. Nechyba, P. Ifju and M. Waszak, "Vision-guided flight stability and control for micro air vehicles", *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3, pp 2134–2140 (2002).

[3] L. Fety, M. Terre and X. Noreve, "Image processing for the detection of the horizon and device for the implemention thereof", United States Patent 5214720, Thompson TRT Defense (1991).

[4] FMA Direct website, http://www.fmadirect.com, visited on 27 October 2006.

[5] B. Taylor, C. Bil, S. Watkins and G.K. Egan, "Horizon sensing attitude stabilisation: A VMC autopilot", *Presented at the 18th International UAV Systems Conference*, Bristol, UK (2003).

[6] J.A. Gozdecki, "Aircraft attitude sensor and feedback control system", United States Patent 6181989 (2001).

[7] T. Hudson, "Autopilot: UAV command and control", http://sourceforge.net/projects/autopilot, visited on 27 October 2006.

[8] Carnegie Mellon University Robotics Institute, "Cmucam vision system", http://www-2.cs.cmu.edu/ cmucam, visited on 27 October 2006.

[9] T.D. Cornall and G.K. Egan, "Measuring horizon angle from video on a small unmanned air vehicle", *Presented at the 2nd International Conference on Autonomous Robots and Agents (ICARA 04)*, Palmerston North, NZ (2004).

[10] G. Barrows, PhD Thesis, "Mixed-mode vlsi optic flow sensors for micro air vehicles", ftp://ftp.centeye.com/pub/Dissertation.pdf, visited on 27 October 2006.

[11] G. Barrows, J.S. Chahl and M.V. Srinivasan, "Biomimetic visual sensing and flight control", *Presented at the 17th International UAV Systems Conference*, Bristol, UK (2002).

[12] T. Netter and N. Franceschini, "A robotic aircraft that follows terrain using a neuromorphic eye", *Proceedings IEEE/RSJ Int Conference on Robots and Systems* 1, pp 129–134 (2002).

[13] 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, pp 455–467 (2002).

[14] T.D. Cornall and G.K. Egan, "Heaven and earth and how to tell the difference", *Presented at the 11th Australian International Aerospace Congress*, Melbourne, Australia (2005).

[15] T.D. Cornall, G.K. Egan and A. Price, "Aircraft attitude estimation from horizon video", *IEE Electronics Letters*, 42(13), pp 744–745 (2006).

[16] T.D. Cornall, "Method and apparatus for determining horizon angle and displacement", Standard Complete Australian Patent Number 2004202924 (2004).

[17] N. Otsu, "A threshold selection method from gray level histograms", *IEEE Trans. Systems, Man and Cybernetics*, 9, pp 62–66 (1979).

[18] A. Price, J. Pyke, D. Ashiri and T.D. Cornall, "Real time object detection for an unmanned aerial vehicle using an FPGA based vision system", *Proceedings 2006 IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, FL, pp 2854–2859 (2006).

[19] B. Etkin, *Dynamics of Atmospheric Flight*, John Wiley and Sons (1972).