Department of Electrical
and
Computer Systems Engineering

Technical Report
MECSE-4-2004

A Low Computation Method to Determine Horizon Angle from Video - Preliminary version

Terry Cornall

MONASH
UNIVERSITY

A Low Computation Method to Determine Horizon Angle from Video-Preliminary version.

Terry Cornall Centre for Telecommunications and Information Engineering
Dept. Electrical and Computer Systems
Monash University
Melbourne, Australia
terry.cornall@eng.monash.edu.au

NOTE:This is a version of a paper that has been submitted for publication which has been abbreviated for reasons of copyright and patent considerations.

***Abstract*— This article details the work of the author 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 author 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.**

Results of trials using simulated horizon views are given and discussed.

Key words. UAV, unmanned aircraft, horizon detection, horizon angle, aircraft attitude, atmosphere, sky, ground, image processing

I. 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 programable 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. [EIN02], [ENIW02]. 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, [LF91], 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, [BT03] for stabilising aircraft and satellites. In fact the International Space Station has three such systems in the Zvezda module [ESA00]. 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. [Hud04].

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.

This paper describes a method that takes advantage of a small, light, visible light video camera, the CMUcam developed at Carnegie Mellon University. [RRN02], [RI], [Omn00] 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 [Mic03], (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. *...Details of exactly how the classified image areas are used to calculate the roll angle of the camera have been removed pending Patent application...*

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.

II. 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. (In the case of a rectangular sensor the asymmetry causes errors, that, whilst they can be compensated for to a certain degree, can be eliminated completely by simply ignoring, or *masking out* the pixels that fall outside of the circle.) First we will define some terms, then we will propose and prove a new theorem. This is followed by the derivation of an equation for the angle that the horizon makes to the horizontal.

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: ...Details removed pending Patent application...

The utility of Theorem 1 is that we can use it for finding the angle of the horizon simply *...Details removed pending Patent application...*. 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 *...Details removed pending Patent application...* 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. *...Details removed pending Patent application...* The process of classifying pixels into sky and ground classes is not so straightforward as the author has implied thus far. The ground is not always darker than the sky in the visible spectrum, and so on. These are topics for further research and for the moment these exceptions will be ignored.

Fig. 1. *...Figure removed pending Patent application...*

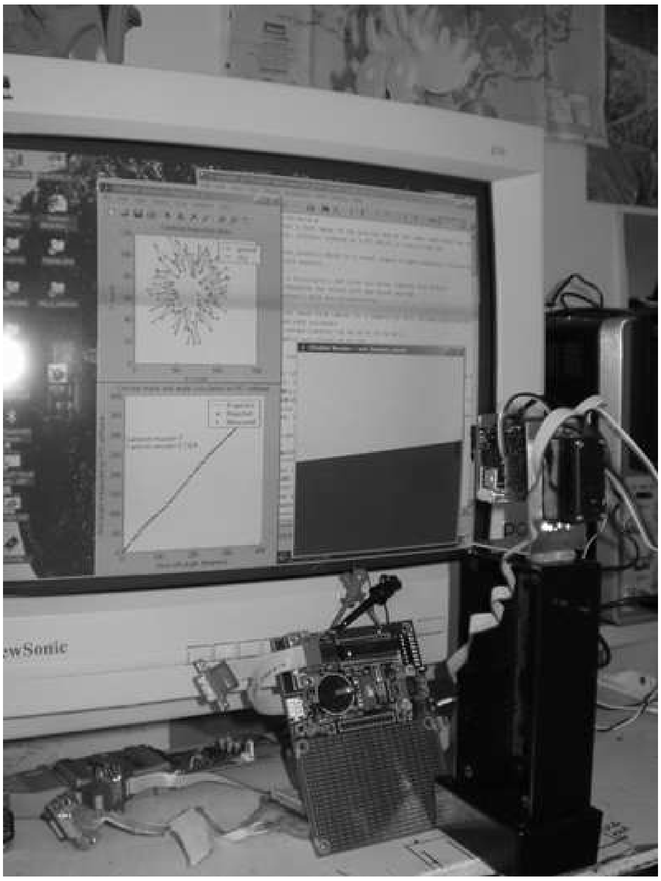

Fig. 2. Experimental setup.

From Theorem 1 *...Details removed pending Patent application...* 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:

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

III. 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 author 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. See figure 2.

That output consisted of the calculated horizon angle, at a rate of about twice per second. The software on the

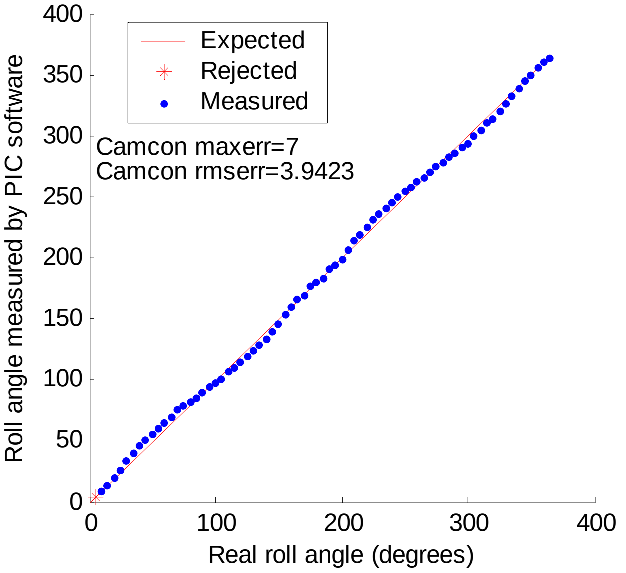

Fig. 3. Measured angle versus real angle during smooth 360° roll. Circular mask and angle calculated on PIC microcontroller.

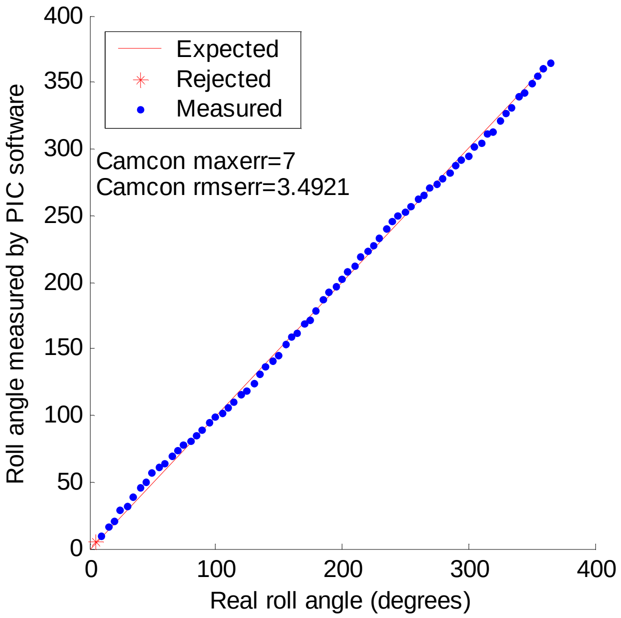

Fig. 4. Measured angle versus real angle for roll with turbulence. Circular mask and angle calculated on PIC microcontroller.

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 *...Details removed pending Patent application...* 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.)

The amount of time it takes process the sky and ground classes and calculate the horizon angle 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.

Figure 3 shows the angle calculated on the PIC during a smooth $360°$ roll. (The first data point is marked *rejected* and not used in the error calculation only as a precaution, as the first measurement was found to often be anomalous due to the nature of the communications between the PIC and the Matlab software on the PC.) Note the RMS error of $3.9°$ which is close to 1% of the full range of $360°$. Contrast this figure to figure 4 which shows the angles during a 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 $\pm 10°$ of pitch and yaw, which was as much as could be applied without having the horizon leave the view. Figure 4 shows, by its remarkable similarity to figure 3 and by the similar RMS error of $3.5°$, just how little effect the disturbances had on the roll angle measurement.

IV. Conclusions and Future Work

So far, all of the development of this method has been carried out using simulated or recorded views of the horizon, and many of the problems that need to be overcome before real flight trials can take place have yet to be addressed. These include adaptive white balance and exposure control to deal with harsh and changeable lighting conditions. Also, this method relies on the assumption that the horizon forms an approximately straight line in the image. In many circumstances this assumption will fail. Methods need to be developed to detect and deal with the situations that will occur when this method is not reliable.

The tested frame-rate of this equipment is not yet fast enough, at about only one frame per second. This is due mainly to factors to do with the current testing environment, not the camera, nor the processing being done on the PIC. The camera is capable of better than 5 frames per second at the current spatial resolution and with appropriate optimisation (and/or a faster microprocessor) the PIC software is capable of keeping up with that rate. A target frame rate of at least 5 frames per second is sought.

The microprocessor being used in trials is not the fastest nor most powerful in its power/size class. Lack of a hardware multiplication facility slows it considerably, (although this is not a real impediment as the nature of the algorithm is such that multiplication is required only once per frame.) The fact that the author had to resort to assembler language programming in order to achieve the required processing speed in the critical sections suggests that a faster microprocessors or DSPs with pipelined architecture and hardware multiplier could be used to improve the speed of the implementation. Further trials *...Details* *removed pending Patent application...* lead the author to expect a reduction in RMS error of about 50%, from 4° to 2°.

The camera being used could also be improved, and there is a second version of the CMUcam now available that has improved resolution and includes a frame buffer (which although not required for this method, certainly promises greater flexibility for other methods.)[RI]. Other cameras could of course be used, with the appropriate provision of a digital interface, although this could require the binarisation of the image to take place in the vision processor. This is not an expensive process and can be done on the fly. Indeed, the author is in the process of developing an interface to allow any analog camera to be connected to a programmable logic device in order to implement the algorithm.

The above challenges and suggested improvements not withstanding, the elegant and resilient 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

[BT03] S. Watkins G Egan B. Taylor, C. Bil. Horizon sensing attitude stabilisation: A vmc autopilot. 18th International UAV Systems Conference, Bristol, UK, 2003.

[EIN02] 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.

[ENIW02] 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.

[ESA00] European Space Agency ESA. Technical description of the zvezda module. http://www.spaceflight.esa.int/file.cfm?filename=mopzvezdamod, July 2000.

[Hud04] Trammell Hudson. Autopilot: Uav command and control. http://sourceforge.net/projects/autopilot/, April 2004.

[LF91] Xavier Noreve Luc Fety, Michel Terre. 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.

[Mic03] Microchip. Pic16f87xa data sheet. Technical report, 2003.

[Omn00] OmniVisionTechnologies. Advanced information preliminary regarding ov6620 single-chip cmos cif color digital camera. Advanced Information Preliminary Version 1.4, OmniVisionTechnologies, Inc, 930 Thompson Place Sunnyvale,California 94086 U.S.A., May 2000.

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

[RRN02] Anthony Rowe, Chuck Rosenberg, and Illah Nourbakhsh. A low cost embedded color vision system. In *Proceedings of IROS 2002*, 2002.