COVER SHEET

Computer-Vision Based Collision Avoidance for UAVs

Ryan J. Carnie[1, 2], Rodney A. Walker[1] and Peter I. Corke[2]

[1] *Cooperative Research Centre for Satellite Systems, Queensland University of Technology, GPO Box 2434, Brisbane, Queensland, 4001, Australia*
[2] *CSIRO ICT Centre, PO Box 883, Kenmore, Queensland, 4069, Australia*

Summary: This research is investigating the feasibility of using computer vision to provide robust sensing capabilities suitable for the purpose of UAV collision avoidance. Presented in this paper is a preliminary strategy for detecting collision-course aircraft from image sequences and a discussion on its performance in processing a real-life data set.

Initial trials were conducted on image streams featuring real collision-course aircraft against a variety of daytime backgrounds. A morphological filtering approach was implemented and used to extract target features from background clutter. Detection performance in images with low signal to noise ratios was improved by averaging image features over multiple frames, using dynamic programming to account for target motion.

Preliminary analysis of the initial data set has yielded encouraging results, demonstrating the ability of the algorithm to detect targets even in situations where visibility to the human eye was poor.

Keywords: collision avoidance, UAV, computer vision, target detection, collision course, morphological filter, track before detect, dynamic programming

Introduction

One of the greatest challenges facing the introduction of UAVs into unrestricted airspace is the development of a collision avoidance capability that meets the standards provided by a human pilot. Current research in this field has experimented with a variety of sensor technologies, such as radar, computer vision, transponders and data-link information exchange (e.g. ADS-B). While future technology, particularly the introduction of ADS-B, will facilitate high performance collision avoidance systems (such as TCAS IV), the reliance on factors such as equipment onboard neighbouring aircraft and the integrity of the GNSS constellation make such systems undesirable as a sole-means of collision avoidance for a UAV. Computer vision offers an alternative, fully self-contained approach or "backup", which emulates the "see and avoid" mechanism of a human pilot. This research is investigating the feasibility of using computer vision to provide robust sensing capabilities suitable for the purpose of UAV collision avoidance. Such an approach must provide a level of performance which is at least equivalent to that of human "see and avoid".

Much research has been published on the topic of automatic target detection using computer vision. Approaches to this problem include spatial techniques, such as mathematical morphology [1, 2] and Gabor filters [3, 4] and temporal-based methods such as maximum likelihood [5], 3D matched filtering [6-8] and dynamic programming [9-12]. In this paper, we implement a combination of morphological filtering and dynamic programming techniques.

In this phase of research, it was desired to investigate the performance of existing target detection algorithms under collision course scenarios in order to gain an understanding of their strengths and limitations. To this end, we have implemented a morphological filter to extract small, point-like targets from large-scale clutter such as clouds. The output of this filter was then passed through a dynamic programming algorithm, which enhanced detection performance in images with poor signal to noise ratios. A description of this preliminary detection strategy, and its performance in processing a sequence of real-life data, is presented.

Target Detection Algorithm

Morphological Filtering

Greyscale morphological filtering for the purposes of target detection generally involves two morphological operations known as *opening* and *closing*. The morphological opening procedure can be generally described as the darkening of small, bright areas (which are too small to accommodate the given morphological *structuring element*) to the values of their neighbouring pixels. Conversely, morphological closing is used to brighten small, dark areas to match the values of their neighbours. These procedures are described graphically in Fig. 1.

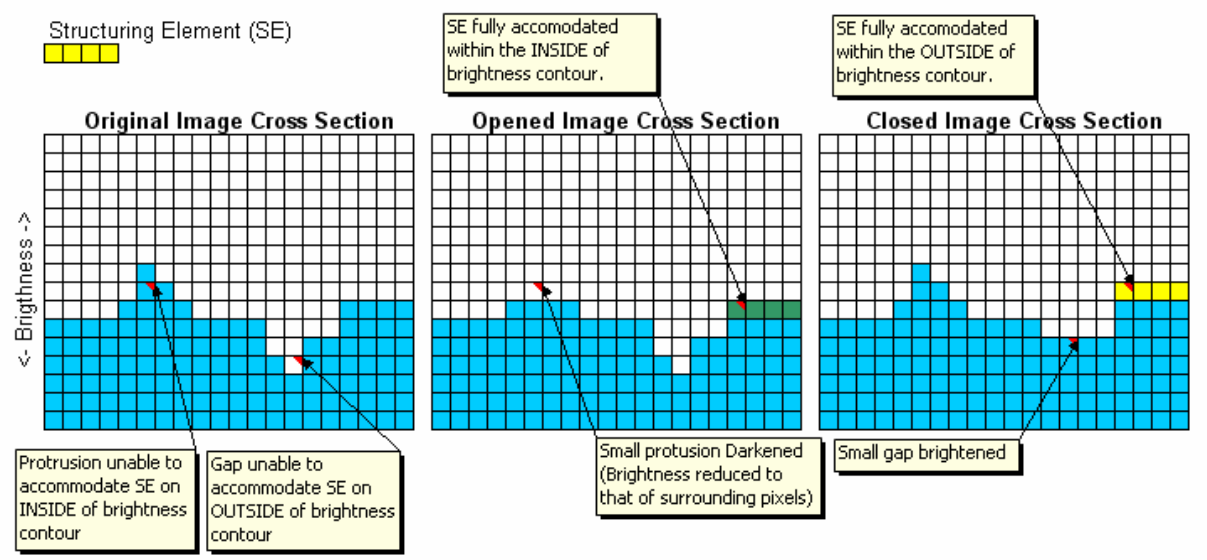

Fig. 1: Opening and Closing Example

Given these basic definitions, it is clear that the difference between an image and its morphological opening is useful for identifying positive (brighter than neighbouring pixels) targets. Accordingly, the difference between a closed image and its original may be used to identify negative (darker) targets. A Close-Minus-Open (CMO) algorithm [1] outputs targets of both positive and negative nature.

The target detection strategy presented in this paper uses the morphological filtering approach suggested by Casasent [1]. This approach takes the minimum CMO response of a pair of morphological filters, using horizontal and vertical 1D slits as structuring elements. This dual-filter approach reduces the probability of false detections due to jagged boundaries on larger clutter as demonstrated in Fig. 2.

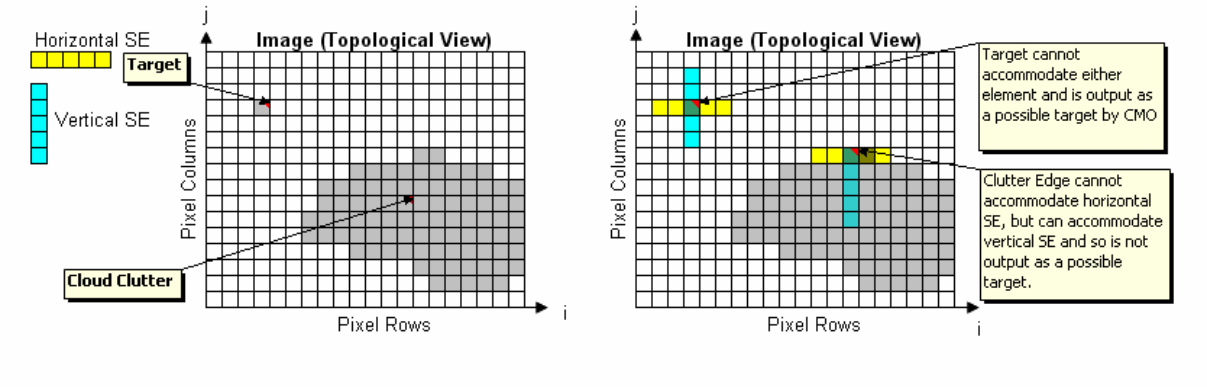

Fig. 2: Demonstration of Dual Morphological Filter Approach

While the CMO algorithm is effective at detecting small target signals, it is also susceptible to false detections due to random noise on individual pixels. The output of the CMO algorithm is thus passed through a dynamic programming algorithm, which desensitises the algorithm to the effects of random noise as discussed in the following section.

Dynamic Programming

The dynamic programming algorithm averages the image sequence of CMO outputs along possible target trajectories, with a decision on the presence of targets being made only after the summation of multiple frames. The number of possible target trajectories can be reduced by considering the possible target state transitions between consecutive frames.

A target signal which is present in an image frame may be represented by a state (i, j, u, v), consisting of a position that resides in the 2D image position space (i, j) and a velocity residing in the 2D velocity space (u, v). The velocity space is discretized and limited to within the range of possible target velocities, with separate branches in the dynamic programming algorithm used to process each (u, v). For the problem of airborne collision avoidance, the near-stationary nature of the target signal [13] allows us to limit the discrete velocity space to $-1 \le u, v \le 1$ pixels per frame. This corresponds to a continuos target velocity of anywhere between 0 and ±1 pixels per frame. The discrete position space (i, j) corresponds to the row-column index of pixels in each frame.

Assuming velocity is constant, it can been shown [10] that for each discrete target state (i, j, u, v) at frame *k*, there are 4 possible state transformations corresponding to frame *k+1*. Given the velocity space for this problem, four velocity branches are sufficient to accommodate possible target motion. Their ranges of valid state transformations are shown below in Fig. 3.

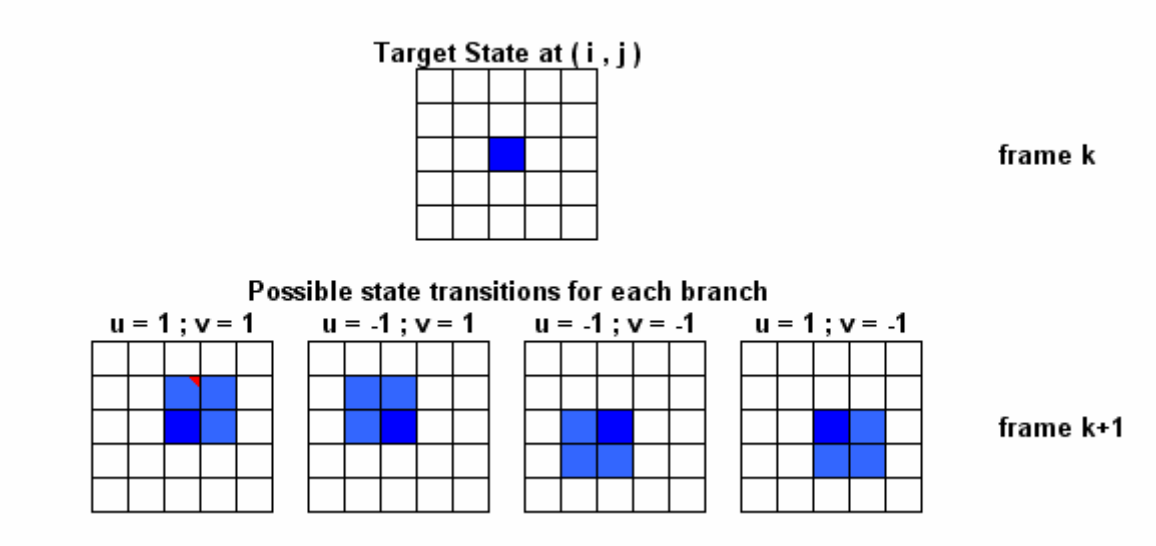

Fig. 3: Possible State Transitions for Dynamic Programming Algorithm

The dynamic programming implementation used in this paper is based on the approach used by Yang [14]. The algorithm is divided into three stages – Initialisation, Recursion, and Decision.

Initialisation

An intermediate image $F_{uv}(i, j, k)$ which recursively tracks possible target states on a frame-to-frame basis is created for each discrete velocity branch.

For all (u, v):

$$F_{uv}(i, j, 0) = f(i, j, 0) \tag{1}$$

Where $f(i, j, k)$ is the image received at frame *k*.

Recursion

For all (u, v):

$$F_{uv}(i, j, k) = [(1-\alpha) f(i, j, k)] + [\alpha \times \max_{(i', j') \in Q(i, j, u, v)} F_{uv}(i', j', (k-1))] \tag{2}$$

Where $0 \le \alpha \le 1$ represents a memory factor and $Q(i, j, u, v)$ represents the four-pixel window of valid *rearward* transitions for target state (i, j, u, v). Since this addition is done recursively, $Q(i, j, u, v)$ is equivalent to the reflection around (i, j) of the possible *forward* transitions which are shown in Fig. 3.

Decision

At frame K, the maximum output on a pixel-to-pixel basis is taken from each of the discrete velocity branches.

$$F_{\max}(i, j, K) = \max_{(u, v)} F_{uv}(i, j, K) \tag{3}$$

This output may be converted to a binary image with the threshold τ set to achieve appropriate probabilities of detection and false alarm.

Data Collection

Sensor Hardware

The digital camera used for this series of trials was a PointGrey Research Dragonfly. Designed specifically for industrial machine vision tasks, the Dragonfly communicates via an IEEE1394 interface and is capable of producing a colour (Bayer tiled) or greyscale resolution of up to 1024x768 pixels. The camera was equipped with a Pentax C-Mount lens with a field of view (FOV) of approximately 17° x 13°, an aperture set to f/8 and focus set to ∞.

Camera Calibration

Ideally, the imaging sensor would show a uniform response to brightness on a pixel-to-pixel basis. In reality, this is not the case. This poses a serious problem for the detection algorithm since particularly dark or bright pixels are likely to be falsely detected by the CMO procedure. Furthermore, the static nature of this error means that it will not be averaged out by dynamic programming. Brightness calibration data was therefore recorded as part of the data collection campaign. Information on the relative brightness gain of each pixel was obtained by detaching the lens from the camera and creating a uniform brightness over the entire area of the imaging sensor. The effect of noise was reduced by averaging the received images over multiple frames and the subsequent output was used to create the appropriate pixel gain matrix for the correction of recorded images. The spread of brightness intensities returned by the imaging sensor under homogeneous illumination is shown in Fig. 4 before and after calibration.

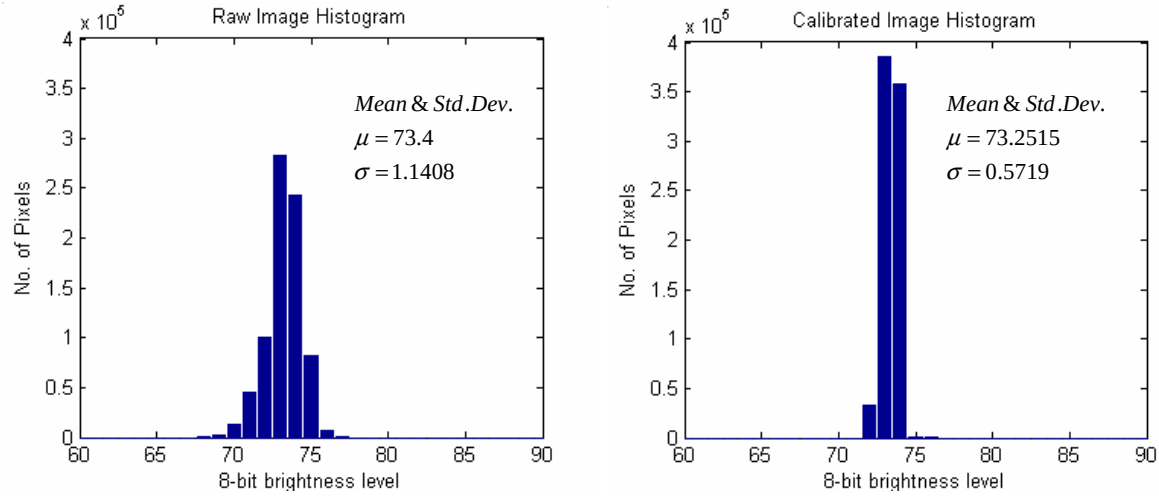

Fig. 4: Image histograms before and after calibration

Note that the standard deviation of the spread has been reduced from 1.1408 to 0.5719, indicating the success of the calibration process.

Field Trials

A preliminary set of daytime data was taken from the top of the Air Traffic Control (ATC) tower at Archerfield Airport, Brisbane. Image streams were recorded of departing aircraft disappearing into the distance and then reversed to simulate an approaching target. However, while this location provided a readily available source of data to assist in the development of the detection algorithm, the flight paths of the target aircraft were not directly aligned with the tower and hence did not directly correspond to a collision situation.

A second set of data was taken from Mary Cairn Cross, a location with an elevation of approximately 1000ft, around 2nmi SE from the township of Maleny. In this set of data, a target aircraft was made to fly directly at the location for a period of time, before gradually pulling away to avoid collision.

In each of these trials, data from the camera was recorded at a frame rate of 7.5 Hz using the Linux-based program Coriander and later processed offline.

Results and Analysis

Data collected from the field trials was processed by the target detection algorithm, with parameters α (forgetting factor) and τ (threshold value) varied to compare differences in performance. It was observed that the target detection algorithm successfully detected and tracked target aircraft throughout the image sequences. Fig. 5 shows the detected path of a target aircraft amongst heavy cloud clutter (taken from the Archerfield data set) as it translates from the lower right portion of the image plane. One frame (#212) of the original image sequence is displayed in Fig. 6(a), showing the nature of the background and the strength of the target signal.

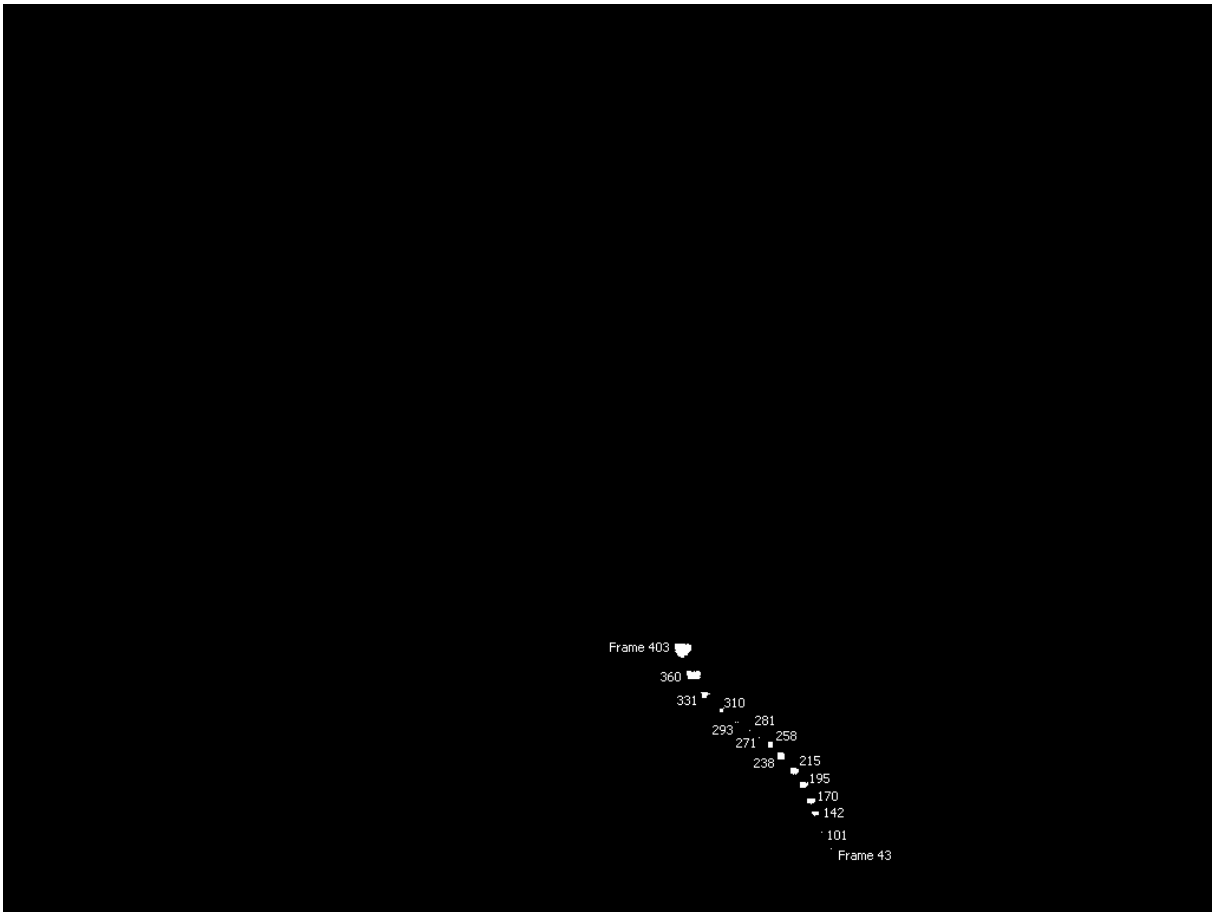

Fig. 5: Path of detected target tracked over multiple frames

Since the output of the target detection algorithm is the binary representation of the dynamic programming response, the sizes of the detected target areas in Fig. 5 correspond to the strength of the target signal rather than target size. This is due to the recursive stage of the dynamic programming algorithm, where peak values are spread to nearby pixels in $Q(i, j, u, v)$ with an attenuation of α. Strong features may be spread multiple times over several frames before their values fall below the binary threshold level, τ. Note that the target signal appears relatively weak between frames 271 and 293, corresponding to a region in the image where there is very poor contrast between the target and the background (Refer to Fig. 6(a)). This is a limitation that is shared with human "see and avoid" capabilities [13] although experimentation with other spectrums (e.g. infrared) may lead to an improvement in detection performance. Aside from this anomaly, it can be observed that the weak target signal is first detected in frame 43, and becomes stronger as the aircraft draws nearer.

The effect of each intermediate stage of the algorithm is demonstrated via the images displayed in Fig. 6. For viewing purposes, the outputs displayed in Fig. 6(c) and (e) have been gamma-corrected with a factor of 0.25 to enhance the detail present in the dark images. Brighter areas in the outputs of the CMO filter and dynamic programming responses indicate the possible presence of targets. The images in Fig. 6(d) and Fig. 6(f) correspond to the binary representations of these images with a fixed threshold of 0.035 (on a 0-1 brightness scale).

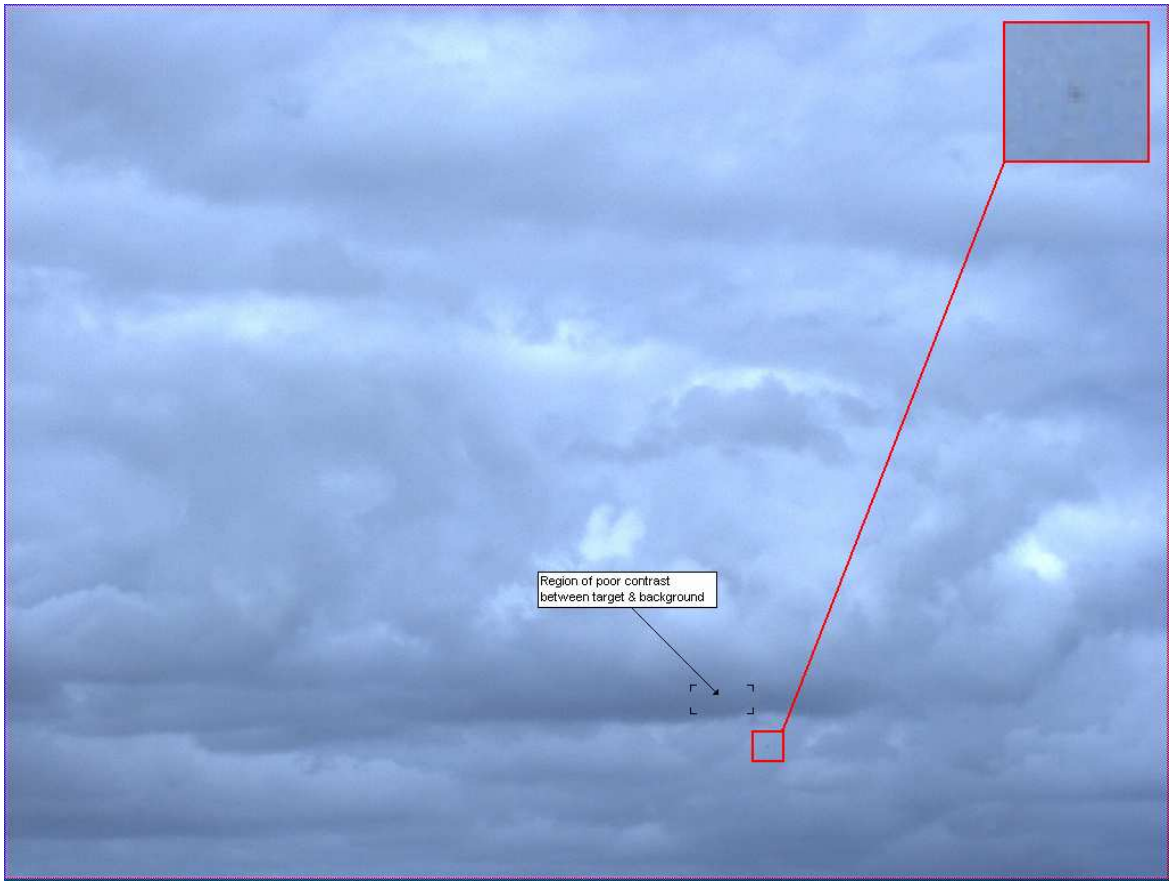

Fig. 6(a): Original image frame (#212) showing location of airborne target

Fig. 6(b): Grayscale version of original image

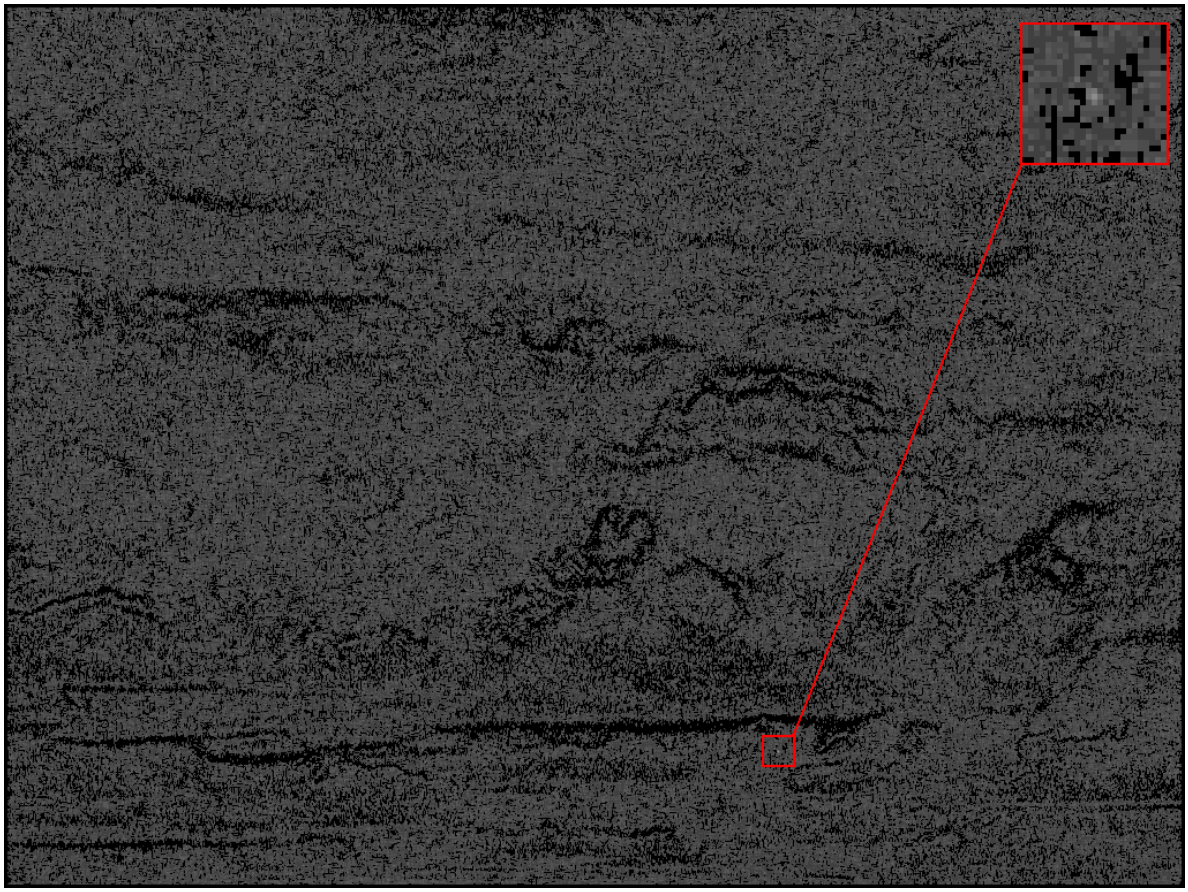

Fig. 6(c): CMO filter output highlighting detected target feature

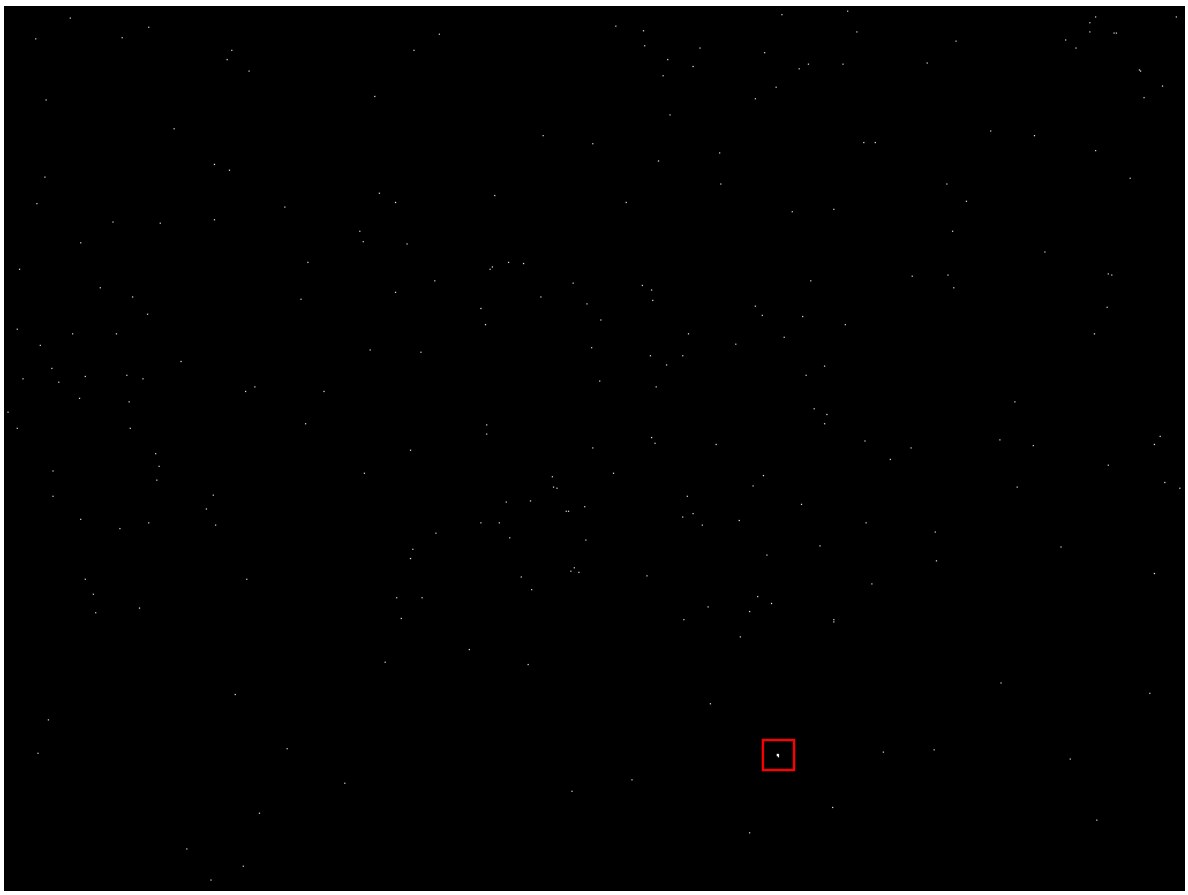

Fig. 6(d): Binary representation of CMO filter output showing detected target & false alarms

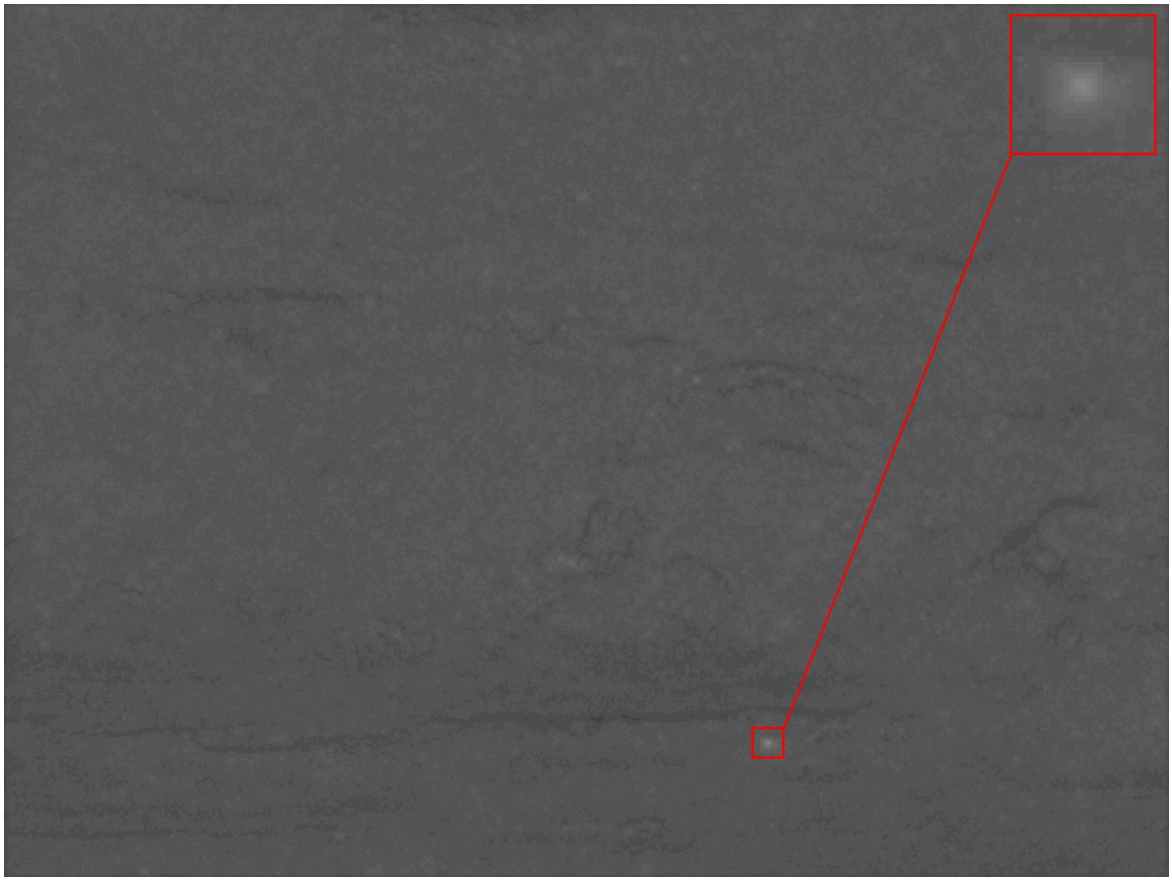

Fig. 6(e): Dynamic programming output highlighting detected target feature.

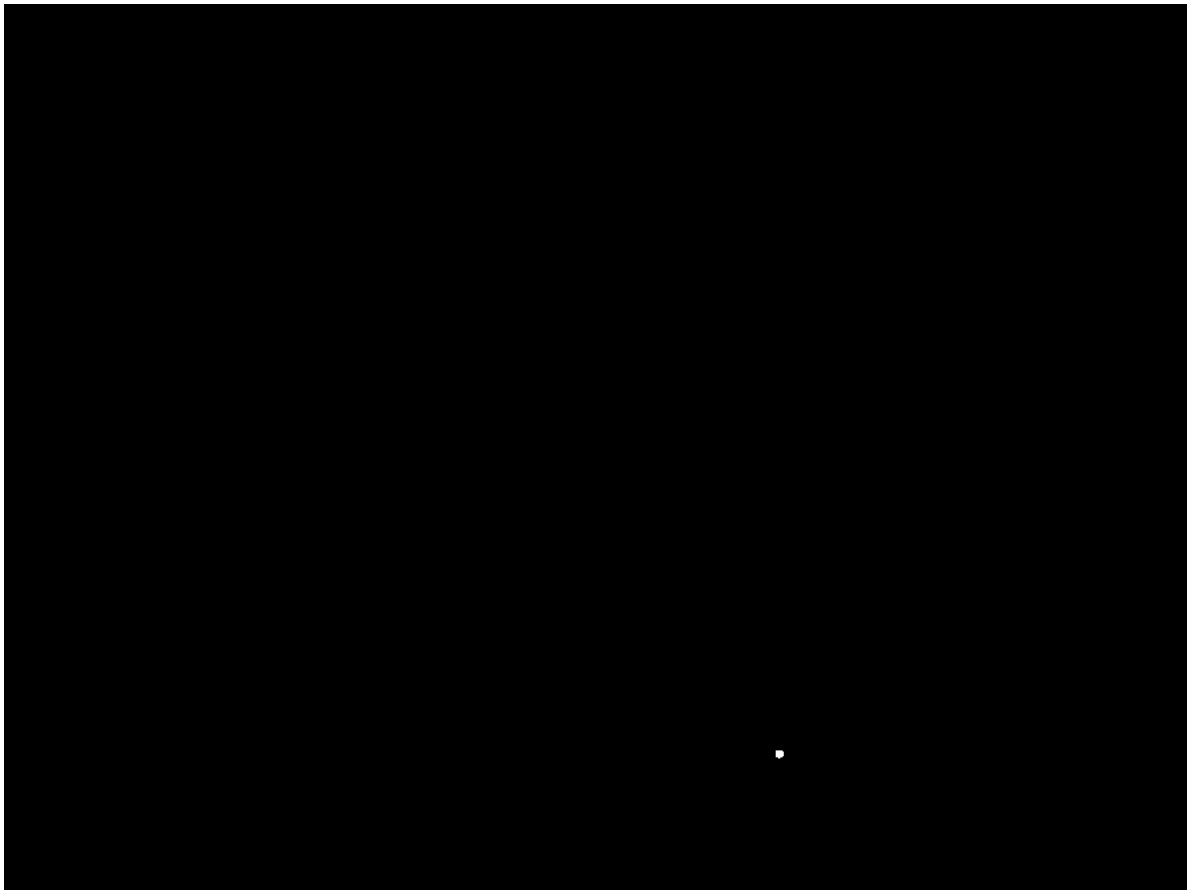

Fig. 6(f): Binary representation of dynamic programming output.

As can be seen, the output of the morphological CMO filter shown in Fig. 6(c) has extracted point-like features from the greyscale image frame. There is a particularly strong feature, surrounded by the red box, which corresponds to the target signal. However, there are also numerous features of varying strengths that correspond to noise on individual pixels. The binary representation of this CMO response shown in Fig. 6 (d) demonstrates that a threshold decision made at this point in the algorithm would be overly susceptible to false alarms due to such noise. Hence, the output of the CMO filter as displayed in Fig. 6(c) is subject to dynamic programming before a decision is made on target presence.

Fig. 6(e) shows that the dynamic programming algorithm has averaged out the effects of noise while maintaining the strength of the target signal. Note that the edges of the target appear less defined; this is a result of the recursive phase of the dynamic programming algorithm. Non-maxima suppression could be used if a single point detection is desired, however it was not necessary for this experiment. The binary version of the dynamic programming output serves as the final decision on the presence of a target, and is observed in Fig. 6(f) to have suppressed the false alarms due to noise while retaining detection of the genuine target.

The variation of algorithm parameters and the corresponding changes in detection performance are summarised in Table 1. In this case, detection performance is measured by the time to first detection in addition to the rates of missed detection and false alarm. The image sequence that was used to generate this data features a distant aircraft which grows from sub-pixel size to a target of four pixels in diameter over 400 frames.

Table 1: Variation in Algorithm Parameters

	$\tau = 0.025$			$\tau = 0.030$			$\tau = 0.035$		
α	**FD**	**MD**	**FA**	**FD**	**MD**	**FA**	**FD**	**MD**	**FA**
0.7	19	0.068	40.135	21	0.174	3.565	101	0.060	0.423
0.75	20	0.047	22.638	29	0.162	1.923	103	0.061	0.303
0.8	21	0.058	13.108	29	0.170	1.230	103	0.061	0.218
0.85	21	0.074	8.158	29	0.170	0.780	105	0.041	0.165
0.9	29	0.059	4.888	40	0.156	0.468	106	0.011	0.130

Where:
FD - The first frame in which the target is successfully detected
MD - Rate of missed detections per frame (after the first detection is made)
FA - Rate of false alarms per frame

Note that the variation of the forgetting factor α, which dictates the strength of the dynamic programming algorithm, has minimal effect on the rates of missed detection presented in Table 1. Lowering the threshold parameter τ is the only reliable means of reducing missed detections. The main effect of increasing the value of α is the suppression of false alarms, which is strongly evidenced in results displayed for all three values of τ. Hence, the dynamic programming algorithm allows the lowering of the threshold parameter τ to minimise the rate of missed detections without generating an otherwise impractical rate of false alarm.

The main disadvantage of increasing α is the resulting increase in the number of frames required before first detection. This is consistent throughout the first detection results for each value of τ and is due to the emerging target having reduced influence (1-α) on the dynamic programming output.

It may be observed that the missed detection rates for τ = 0.030 are relatively high compared to other values of τ. This was due to the intermittent rising and falling of the target signal above and below the threshold value 0.030 between frames 30 and 95. Note that this is consistent with low missed detection rates for τ =0.025 and a first detected frame of 101 for τ =0.035.

Conclusions and Future Work

This paper has presented a computer-vision based aircraft detection algorithm with a view to developing UAV collision avoidance capabilities. Target features were extracted via a morphological filtering approach and dynamic programming was used to improve performance in images with low signal to noise ratios. Preliminary results, which demonstrated an ability to detect distant aircraft even in the presence of heavy cloud clutter, are encouraging.

In future research, we will explore further techniques that will allow the algorithm to accommodate varying lighting conditions and more complex backgrounds with features such as heavy terrain clutter. Of particular interest is research by Gandhi [15], who suggests the distinguishing of targets from clutter based on the properties of low translation and large expansion over time. Additionally, future work with cameras onboard moving platforms will require the development of a strategy to compensate for ego-motion effects due to camera rotations and translations, possibly through integration with inertial sensors.

Ultimately, this research will endeavour to identify the amount of target information which can be reliably extracted from images and how this information can be used for the purposes of UAV collision avoidance. Consideration will also be given to what, if any, supporting sensors (e.g. radar) may be required in order to achieve a sufficiently robust solution.

References

1. D. Casasent and A. Ye, "Detection filters and algorithm fusion for ATR," *Image Processing, IEEE Transactions on*, vol. 6, pp. 114-125, 1997.

2. N. Sang, T. Zang, and G. Wang, "Gray scale morphology for small object detection," *SPIE 2759 Signal and data procession of small targets*, pp. 123-129, 1996.

3. N. Braithwaite and B. Bhanu, “Hierarchical Gabor filters for object detection in infrared images,” *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pp. 628-631, Seattle, Washington, June 20-23, 1994.

4. A. K. Jain, N. K. Ratha, and S. Lakshmanan, "Object detection using gabor filters," *Pattern Recognition*, vol. 30, pp. 295-309, 1997.

5. S. D. Blostien and T. H. Huang, "Detecting small moving objects in image sequences using sequential hypothesis testing," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* pp. 1611-1629, 1991.

6. I. S. Reed, R. M. Gagliardi, and L. B. Stotts, "Optical moving target detection with 3-D matched filtering," *Aerospace and Electronic Systems, IEEE Transactions on,* vol. 24, pp. 327-336, 1988.

7. B. Porat and B. Friedlander, "A frequency domain algorithm for multiframe detection and estimation of dim targets," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 12, pp. 398-401, 1990.

8. M. Diani, G. Corsini, and A. Baldacci, "Space-time processing for the detection of airborne targets in IR image sequences," *Vision, Image and Signal Processing, IEE Proceedings,* vol. 148, pp. 151-157, 2001.

9. Y. Barniv, "Dynamic programming solution for detecting dim moving targets," *Aerospace and Electronic Systems, IEEE Transactions on,* vol. AES-21, 1, 1985.

10. S. M. Tonissen and R. J. Evans, "Target tracking using dynamic programming: algorithm and performance," *Proceedings, IEEE Conference on Decision and Control, (CDC '96),* pp 2741-2746, Kobe, Japan, 1996.

11. S. M. Tonissen and R. J. Evans, "Peformance of dynamic programming techniques for Track-Before-Detect," *Aerospace and Electronic Systems, IEEE Transactions on,* vol. 32, pp. 1440-1451, 1996.

12. J. Arnold, S. W. Shaw, and H. Pasternack, "Efficient target tracking using dynamic programming," *Aerospace and Electronic Systems, IEEE Transactions on,* vol. 29, pp. 44-56, 1993.

13. Bureau of Air Safety Investigation, "Limitations of the See-and-Avoid Principle," 1991.

14. M.-T. Yang, T. Gandhi, R. Kasturi, L. Coraor, O. Camps, and J. McCandless, "Real-Time Implementations of Obstacle Detection Algorithms on a Datacube MaxPCI Architecture," *Real-Time Imaging 8,* pp. 157-172, 2002.

15. T. Gandhi, M.-T. Yang, R. Kasturi, O. Camps, L. Coraor, and J. McCandless, "Detection of obstacles in the flight path of an aircraft," *Aerospace and Electronic Systems, IEEE Transactions on,* vol. 39, pp. 176-191, 2003.