

Detection of Obstacles in the Flight Path of an Aircraft

Tarak Gandhi¹, Mau-Tsuen Yang¹, Rangachar Kasturi¹,
Octavia Camps^{1,2}, Lee Coraor¹, and Jeffrey McCandless³

¹Computer Science and Engineering

²Electrical Engineering

Pennsylvania State University

University Park, PA 16802

³Flight Management and Human Factors Division

Mail Stop 262-2

NASA Ames Research Center

Moffett Field, CA 94035

{gandhi,myang,kasturi,camps,coraor}@cse.psu.edu, jmccandless@mail.arc.nasa.gov

Abstract

The National Aeronautics and Space Administration (NASA), along with members of the aircraft industry, recently developed technologies for a new supersonic aircraft. One of the technological areas considered for this aircraft is the use of video cameras and image processing equipment to aid the pilot in detecting other aircraft in the sky. The detection techniques should provide high detection probability for obstacles that can vary from sub-pixel to a few pixels in size, while maintaining a low false alarm probability in the presence of noise and severe background clutter. Furthermore, the detection algorithms must be able to report such obstacles in a timely fashion, imposing severe constraints on their execution time. This paper describes approaches to detect airborne obstacles on collision course and crossing trajectories in video images captured from an airborne aircraft. In both cases the approaches consist of an image processing stage to identify possible obstacles followed by a tracking stage to distinguish between true obstacles and image clutter, based on their behavior. The crossing target detection algorithm was also implemented on a pipelined architecture from DataCube and runs in real time. Both algorithms have been successfully tested on flight tests conducted by NASA.

1 Introduction

Continued advances in the fields of image processing and computer vision have raised interest in their suitability to aid pilots to detect possible obstacles in their flight paths. For the last few years, NASA has been exploring the use of image sequences for detecting obstacles in the flight path

of an aircraft. In the design of a High Speed Civil Transport (HSCT) aircraft with a limited cockpit visibility, NASA has proposed an External Visibility System (XVS) in which high resolution video images would be obtained using cameras mounted on the aircraft. These images can be used to detect obstacles in the flight path to warn the pilots and avoid collisions.

Algorithms for detection of airborne objects from images are abundant in the published literature. Nishiguchi et al. [12] proposed the use of a recursive algorithm to integrate multiple frames while accounting for small object motion. A dynamic programming approach was used by Barniv [4] and Arnold et al. [2] to detect moving objects of small size. The theoretical performance of this approach was characterized by Tonissen and Evans [13].

The above algorithms perform well when the background is uniform. However, in real situations the hazardous object should also be detected against cluttered backgrounds, such as clouds, ground or water. The objects that cross the host aircraft have a significant translation in the image. Hence, subtraction of consecutive images can be used to remove the stationary clutter. If the background clutter also has a significant motion, its motion should be separated from the motion of the target. Irani and Anandan [8] separated the scene motion into planar and parallax motion, and identified independently moving objects which have a significant parallax.

However, objects on a collision course could be nearly stationary in the image. Image differencing is not useful in such a case, since it can remove the object as well. Morphological filtering [6] removes objects of large size, usually corresponding to clutter while retaining the objects of small size. This approach is useful in removing large clutter, such

as clouds. However, it does not remove small-sized clutter.

This paper describes the approaches we have used to detect objects on collision course, as well as those crossing the host aircraft. At present, the detection of collision course and crossing objects are separately implemented. The detection is divided into two stages, the image processing stage and the tracking stage. The image processing stage operates on the entire image, removes most clutter, and isolates features that potentially correspond to targets. The tracking stage tracks these features over a number of frames, measures their properties, and tries to separate the genuine targets from clutter using these properties. Since the first stage has reduced the volume of data to be operated on, more complex tracking algorithms can be implemented without sacrificing the overall time complexity.

For collision course objects, the image processing stage uses morphological filter to separate most clutter. To discriminate the collision course object from remaining clutter, the difference in translation and expansion of corresponding image features is used in the tracking stage. The effectiveness of this approach was demonstrated on a real image sequence captured from an aircraft. For crossing objects, image processing consists of steps such as image differencing and low-stop filtering to remove stationary clutter. This stage was implemented on a pipelined processor system, the DataCube MaxPCI to obtain real time performance. This was followed by tracking of features with a significant and consistent motion, on the associated host machine. Crossing object detection was demonstrated on several image sequences obtained from flight tests conducted by NASA.

2 Flight Maneuvers for Collision Course and Crossing Scenarios

The flight maneuvers were based out of NASA Langley Research Center in Virginia. Two classes of maneuvers were flown, as shown in Figure 1. In the collision course maneuver, the host aircraft was a Boeing 737 and the target aircraft was a Lockheed Martin C-130, both of which were owned by NASA. This maneuver was initiated with the target aircraft climbing directly towards the host aircraft. Before collision could occur, the target aircraft leveled off and flew under the host. Because of safety concerns, the collision course maneuver was conducted only two times. In one of these maneuvers, the background was uniform and it was trivial to separate the target, whereas in the other, the background was severely cluttered and the elimination of this clutter required the use of feature motion behavior, as described in this paper. As an alternative to the collision course maneuvers, flights were conducted with the target aircraft flying directly away from the host aircraft. The images from these maneuvers could be played in reverse to partly simulate a collision condition. The drawback to this

approach is that the background motion and magnitude of relative velocity do not correspond to a collision condition. As a result, such maneuvers are not examined for this paper. In the crossing maneuver, the host aircraft was a modified Convair C-131B (owned by the United States Air Force) and the target aircraft was a Beech King Air B-200 (owned by NASA). This maneuver was safer to perform than the collision course maneuver and was conducted dozens of times.

Planning and running the flight tests was not a trivial task because of safety concerns and the number of people involved. All flights required months of advance planning to pass the Air Force and NASA safety reviews. Each aircraft required pilots, crew, and additional support personnel on the ground for refueling, maintenance, and logistics. A large number of personnel were required because the flights supported not only image processing, but also other aspects of the XVS program. For example, numerous types of displays were tested for their ability to show other aircraft at different distances, and other sensors (such as radar) were tested for their detection ability.

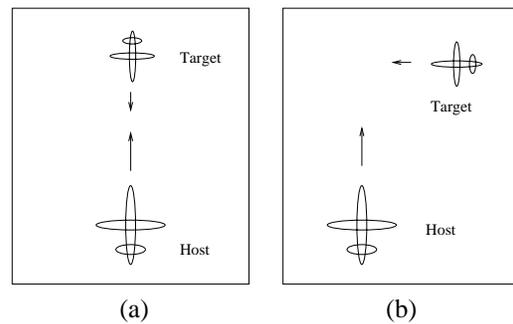


Figure 1. (a) Collision course maneuver: The target aircraft flies towards the host aircraft. (b) Crossing maneuver: The target aircraft flies perpendicular to the host aircraft.

3 Discrimination of Hazard from Clutter

It is well known in the pilots' community, that an object on a collision or near-collision course remains stationary or nearly stationary in its 2-D image view [10]. This property can be used to distinguish hazardous objects from clutter, by measuring the rate of translation of the features in the image. Another useful property is the rate of image expansion, which is approximately inversely proportional to the time to collision. Nelson and Aloimonos [11] use the image expansion in terms of the flow field divergence to estimate the time to collision, for separating obstacles. Francois and Bouthemy [7] separate the image motion into components of divergence, rotation, and deformation. Ancona and Poggio [1] use 1-D correlation to estimate optical

flow for a time-to-crash detector. Baram and Barniv [3] rely on object texture to extract information on local expansion. Instead of estimating a numerical depth value, an object is classified as ‘safe’ or ‘dangerous’ using a pattern recognition approach. Most of these methods are useful for objects of larger sizes. However, in our case, the object sizes can be very small, even sub-pixel, along with very small rates of expansion. Hence, a feature based approach was used in this work, where the rate of expansion was estimated by tracking features over a large number of frames.

Consider an object approaching towards the aircraft with a *relative* velocity of V as shown in Figure 2 (a). Let p be the distance of passage which is the closest distance that the object approaches the camera, and τ denote the time to passage (or collision) which is the time the object takes to reach the distance of passage. The object distance is denoted by r . The corresponding geometry for the background is shown in Figure 2 (b). The declination angle of the line of sight, denoted by θ determines the position of the feature in the image. The velocity of the host aircraft has a magnitude of V_0 and angle of inclination α . The background distance is denoted by r_0 .

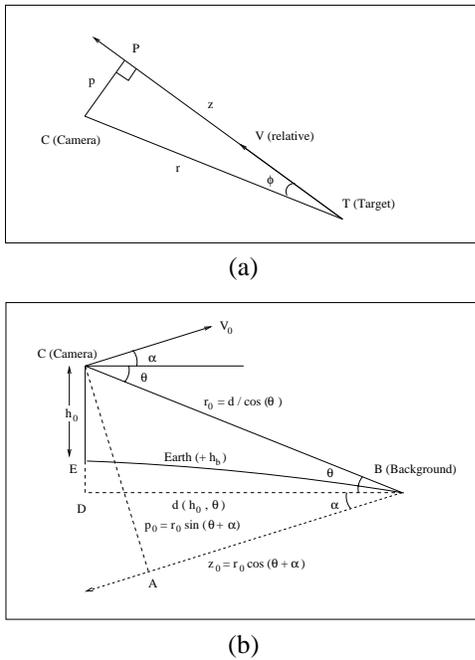


Figure 2. Geometry of (a) target (b) background moving relative to the camera.

3.1 Target Translation

Referring to the scene geometry of Figure 2 (a), the rate of angular translation of an object in the image is given by $T = \dot{\phi}$ – i.e., the rate of change of angle ϕ . The pixel

translation is approximately given by multiplying T by the camera focal length. It can be shown [9] that:

$$T = \dot{\phi} = \frac{pV}{r^2} = \frac{p \cos \phi}{\tau r} \quad (1)$$

Thus, the rate of image translation is proportional to the distance of passage, and the objects on a collision course are likely to have a smaller image motion compared to other objects. However, the rate of translation also depends on the target distance, and a nearer target moves faster in the image than a farther target with the same distance of passage.

It should be noted that the above relationship is valid only if the aircraft does not rotate or vibrate around its own axes. If there is rotation, it should be compensated by using data from the aircraft navigation system. In absence of this data, image features due to clutter should be used to perform the compensation, by modeling their image motion.

3.2 Target Expansion

The rate of image expansion of any object is given by $E = \dot{s}/s$ where s is the size of the object in the image. It is well known that this rate of image expansion is inversely proportional to the time to collision. It can be shown [9] that:

$$E = -\frac{\dot{r}}{r} = \frac{zV}{r^2} = \frac{V \cos \phi}{r} = \frac{\cos^2 \phi}{\tau} \quad (2)$$

$$\tau = \frac{z}{V} = \frac{r \cos \phi}{V} \quad (3)$$

For $\tau = 25 \text{ s} = 750 \text{ frames}$, $E = 0.13\%$ per frame, which is a very small magnitude, measured by tracking over a large number of frames.

3.3 Background Translation

The relationship between the image translation and the distance of passage can be used to remove the clutter which is not on collision course and thus expected to have a large image motion. However, the image motion is inversely proportional to the distance of the object from the camera. Thus, if clutter is at a large distance, it too could have a small image motion. Hence, the image translation of an object should be compared with that of a ground feature in the same line of sight. To compute the rate of background translation, the corresponding background parameters are substituted in equation (1) as:

$$T_0 = \frac{p_0 V_0}{r_0^2} \quad (4)$$

where p_0 , V_0 , and r_0 are shown in the scene geometry of Figure 2 (b). For reliable discrimination between hazard and background, the rate of translation of the hazard

should be much smaller than that of the background – i.e., $T \leq \eta_t^{-1} T_0$ with $\eta_t > 1$, having a larger value for greater discriminating power. In such a case, it can be shown [9] that the following condition should be satisfied:

$$\sin(\theta + \alpha) \sin \theta \geq \eta_t D Q \sqrt{1 - Q^2} f(h_0, \theta) \simeq \eta_t D Q \quad (5)$$

with

$$D = \frac{h_0}{\tau V_0}, \quad Q = \frac{p}{r} = \sin \phi, \quad \cos \phi = \sqrt{1 - Q^2} \simeq 1$$

$$f(h_0, \theta) = 2 \left[1 + \sqrt{1 - 2h_0 / (R_e \tan^2 \theta)} \right]^{-1} \quad (6)$$

where R_e is the radius of the earth, and $f(h_0, \theta)$ accounts for the curvature of the earth. For nearby points, $f \simeq 1$, whereas for points on the horizon, $f = 2$. Since the object distance cannot be greater than the background distance in the line of sight, $r \leq r_0$, one can also write:

$$\sin(\theta + \alpha) \geq \frac{\eta_t p \sqrt{1 - Q^2}}{\tau V_0} \simeq \frac{\eta_t p}{\tau V_0} \quad (7)$$

which is approximately independent of r . Hence, it can be said that for detection to be possible at all for a particular θ and α , the above condition is necessary irrespective of the target distance r . For example, if we have:

$$p = 150 \text{ m} \simeq 500 \text{ ft}, \quad \tau = 25 \text{ s},$$

$$V_0 = 150 \text{ m/s} \simeq 290 \text{ knots}, \quad h_0 = 1 \text{ km} \simeq 3280 \text{ ft},$$

$$\alpha = 0, \quad \eta_t = 2.5, \quad f \simeq 1 \quad (8)$$

For these values $D = 0.267$ and from equation (7), the necessary condition is $\theta \geq 5.7^\circ$. This condition corresponds to the target being at the same position as the background, which is $r = r_0 = 10 \text{ km} \simeq 5.4 \text{ nmi}$, or $Q = 0.015$. However, if the target is nearer, the condition on θ is determined by equation (5). For example, if a hazard should be detected at $r = 5 \text{ km} \simeq 2.7 \text{ nmi}$ or $Q = 0.03$, one would really need $\theta \geq 8.1^\circ$. The required θ increases as r decreases.

3.4 Background Expansion

For estimating the rate of background expansion, the background parameters are substituted in equation (2) as:

$$E_0 = \frac{z_0 V_0}{r_0^2} \quad (9)$$

If reliable discrimination of the hazard from the background is required, the rate of expansion of the hazard must be much larger than that of the background, – i.e., $E \geq \eta_e E_0$ with $\eta_e > 1$, having a large value for greater discriminating power. Using the geometry of Figure 2 (b), it can be shown [9] that the following condition is required:

$$\cos(\theta + \alpha) \sin \theta \leq \eta_e^{-1} D (1 - Q^2) f(h_0, \theta) \simeq \eta_e^{-1} D \quad (10)$$

where D and Q are given by equation (6). For the conditions stated in equation (8), we need $\theta \leq 6.2^\circ$ for reliable detection using expansion.

3.5 Analysis

The behavior of conditions required for detection using translation and expansion is shown as plots of required angle θ against $D = h_0 / (\tau V_0)$ in Figure 3. The dashed line shows the maximum allowable θ for detection using expansion (independent of the target distance). The other curves show the minimum θ required for detection using translation for various values of target distance r in km , for the distance of passage $p = 150 \text{ m}$. For a wider range of detection, the required minimum θ for translation should be small, whereas the maximum allowable θ using expansion should be large. The minimum θ for translation increases with $D = h_0 / (\tau V_0)$ as well as $Q = p/r$. However, maximum θ for expansion which is independent of Q increases faster with D . Hence, a large value of D – i.e., large aircraft height h_0 , small time to passage τ , and small host aircraft velocity V_0 – would give a greater range of θ for which at least one of the two approaches would be useful for discrimination. However, detection using translation improves for small D and Q – i.e., large target distance r but small distance of passage p .

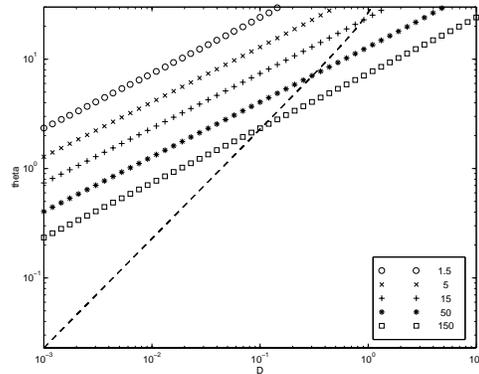


Figure 3. Plots of detection behavior using translation and expansion.

4 Collision Course Object Detection

For detecting objects on a collision course, the image processing stage uses morphological filtering to remove most of the background clutter. The tracking stage tracks the translation and expansion of the features for a number of frames, and using these to separate the objects on collision course, from the remaining background clutter. The approach used for detecting the objects on a collision course

was tested on the image sequence provided by NASA, containing an aircraft on a collision course with severe background clutter. The approach successfully discriminates the object from the clutter. The running time for the current implementation is approximately 7 seconds per frame, with a scope for improvement by optimizing the code. Specialized hardware may be able to improve the performance further to enable real time implementation.

4.1 Image Processing Stage

A morphological filter [6] can remove large sized features (usually clutter), while retaining small sized features (usually targets). A difference between the original image and its morphological opening (top-hat transform) outputs small-sized positive targets (bright targets in dark background). On the other hand, the difference between the morphological closing and the original image (bottom-hat transform) outputs negative targets (dark targets in bright background). Both these images are non-negative, and can be separately used to detect targets.

A single mask for these morphological operations gives undesirable outputs for jagged boundaries of large features. Hence, a horizontal mask m_x and a vertical mask m_y were used separately as proposed by [6]. These masks are of length 5 with origin at the center of the mask, with all the pixels having the default value of zero. The outputs are given by:

$$F_+ = F - \max\{F \circ m_x, F \circ m_y\} \quad (11)$$

$$F_- = -F + \min\{F \bullet m_x, F \bullet m_y\} \quad (12)$$

where \circ and \bullet denote morphological opening and closing operations, respectively. Non-maximal suppression was performed on the outputs of the filters, and pixels exceeding a threshold were sent as features to the tracking stage.

4.2 Tracking Stage

To estimate the translation of the features, they were tracked over a large number of frames. Since the navigation system data was available, the position of the features were compensated using this data. The strongest feature in a window around the predicted position was taken as the corresponding feature in the next frame, and the smoothed estimates of the feature position and velocity in each frame were obtained using Kalman filter approach.

For detecting expansion, a 15×15 window around each feature was explored. The sub-image corresponding to the window was thresholded, and the connected component containing the center of the window was found. All the pixels in the sub-image that did not belong to the component were set to zero. The sub-image was convolved with a number of smoothing masks. These masks perform matched

filtering with a object templates corresponding a number of different sizes. The maximum output from all these masks was considered as the measure of target strength. The expansion was measured in terms of increase of the target strength, tracked over a number of frames. The target strength was plotted against the frame number, and the mean rate of expansion was estimated by applying least squares to the logarithm of the target strength.

4.3 Results

The estimation of translation and expansion was performed on a sequence of images captured from an analog camera in which the target aircraft is approaching the camera. The aircraft was flying at a barometric height of around 3200 *ft* (975 *m*), with the airspeed around 160 *knots* (82 *m/s*). The inclination angle α was less than 3° . For $\tau = 25$ seconds, equation (6) gives $D = 0.329$. If $p = 150 \text{ m} \simeq 500 \text{ ft}$ is allowed, and $\eta_t = \eta_e = 2.5$, then the necessary condition for reliable detection using translation by equation (7) is $\theta \geq 10.5^\circ$. Reliable detection using expansion requires $\theta \leq 7.6^\circ$ using equation (10). Since the FOV of the camera was small (9.75°), expansion would be more suitable for most parts of the image under the above conditions. However, for larger values of τ , translation would be more favorable than expansion. For example, $\tau = 100$ seconds gives $D = 0.0822$, and the conditions for reliable detection with translation and expansion would change to $\theta \geq 2.6^\circ$ and $\theta \leq 1.9^\circ$, respectively. In practice, it was observed that both translation and expansion could separate the collision course aircraft from most clutter.

Figure 4 (a) and (b) show the first and the last image frames in the 482 frame (16 second) sequence used for target detection. Figure 4 (c) and (d) show all the target tracks before and after motion compensation, respectively. The target track is shown separately in Figure 5 (a) and (b). Figure 5 (c) shows the plot of the estimated target strength against the frame number, to measure the target expansion. Corresponding plots for a clutter track are shown in Figures 5 (d)-(f). It can be seen that the target track has a smaller rate of translation but a larger rate of expansion than the clutter track. A scatter plot of the feature expansion against translation for these tracks, including the target track is shown in Figure 6. The rate of translation is measured in terms of the displacement magnitude of the compensated features in 100 frames, whereas the expansion is measured in terms of the increase in the logarithm (to base 10) of the target strength in 100 frames. The target having large rate of expansion and small rate of translation is located in the upper left corner of the plot. A clutter feature in the upper right corner having large rates of both translation and expansion is from lower part of the image (large θ).

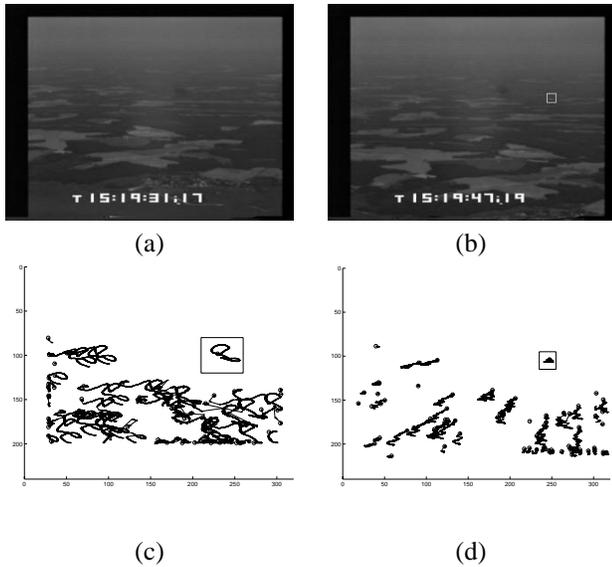


Figure 4. (a) First (b) last frames in the image sequence used for target detection. Feature tracks (c) before (d) after compensation. The target, not visible in (a), is marked by a rectangle in (b)-(d).

5 Crossing Object Detection

The image processing stage of the detection system performs image differencing and low-stop filtering to remove stationary and uniform background. The tracking stage tracks the feature positions, integrates the feature strength over time, and checks for significant and consistent motion in the features to distinguish target from clutter. The image processing stage was implemented on the pipelined image processing system, the DataCube MaxPCI, whereas the tracking stage was implemented on the associated host machine. For a successful real-time implementation, the output rate of image processing stage was matched to the input rate of the tracking stage by dynamically selecting the threshold for feature extraction. This is known as the rate constraint criterion [5]. The system was mounted on the host aircraft, and flight tests were conducted by NASA with another aircraft flying in front of this aircraft. The detection and tracking of the target aircraft were demonstrated during the flight test.

It should be noted that simple algorithms were used for the system to get real time performance. Other approaches requiring optical flow computations were tried [9], but were found to be unsuitable for real time implementation with the particular hardware. The image processing stage and tracking stage for crossing object detection are briefly described below, and in more detail in [9].

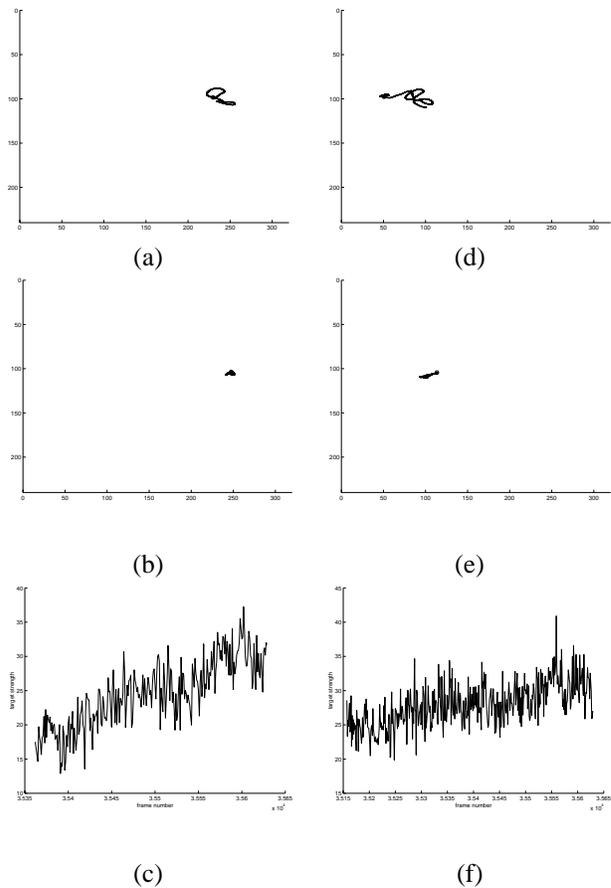


Figure 5. Translation and expansion for typical target and clutter tracks: Target track (a) before (b) after compensation. (c) Plot of expansion of target track against frame number. (d)-(f) Corresponding plots for a clutter track. The target track has a smaller rate of translation and a larger rate of expansion.

5.1 Image Processing Stage

This stage performs the basic image processing steps to suppress clutter and extract features which could potentially be crossing targets. The resolution of the image was reduced by performing a smoothing and down-sampling operation, so that real time operation becomes feasible with the current hardware. A low-stop filter was applied to the reduced image to suppress background clutter. Image differencing was then performed by subtracting consecutive frames, which is equivalent to low-stop filtering in temporal direction. Since the object is assumed to be translating, image differencing would suppress stationary objects corresponding to background clutter.

Directly using the output of the previous step would give rise to a large number of features for an extended target.

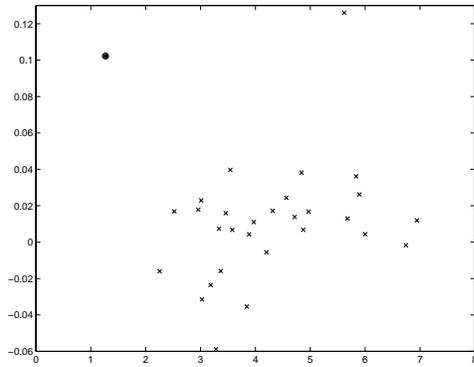


Figure 6. Scatter plot of the feature expansion against translation: The target marked as an encircled asterisk, is in upper left corner, having a small rate of translation and a large rate of expansion.

Non-maximal suppression was performed to get a single feature (or sometimes a small number of features) for the entire target. To extract candidate features, thresholding was performed. The threshold was selected using the image histogram, so that the number of features match the processing rate of the slower tracking stage. The positions and the amplitudes of the resulting features were transmitted to the tracking stage.

5.2 Tracking Stage

This stage maintains a list of tracks containing the position, velocity, amplitude, and the track life. The list is empty in the beginning. For each track in the list of tracks, the strongest feature from the image processing stage in a neighborhood window is selected as the continuation of the track. Kalman filter is used to update the track position and velocity. The feature amplitude is scaled by a forgetting factor, and added to the track amplitude. If no feature satisfying the above conditions is found in the neighborhood of the track, the position and velocity are extrapolated using only the state update.

After all the current tracks are updated, features in the feature list are used to check for new tracks. For every feature, the list of tracks is scanned to see if a track is already there in its neighborhood. If not, a track is created out of the feature. Its position will be the same as feature position whereas velocity initialized to zero. The actual velocity will be computed only in the next frame.

If the number of tracks is too large, the stage can get overloaded and fail to operate in real time. In such a case, the weakest tracks are deleted to prevent overloading. Furthermore, tracks which are very close to each other and have nearly the same velocity are merged, to eliminate multiple

tracks corresponding to same object. Tracks which satisfy the criteria of the object – i.e., having an amplitude larger than a threshold, as well as a significant and consistent track velocity – are output as potential objects.

5.3 Results

The real-time image capturing, recording, and processing system was demonstrated on the flight tests conducted by NASA. During the first set of flight tests, image sequences were captured and recorded successfully at the rate of 30 frames per second. The tracking algorithms were designed and fine-tuned using these image sequences. During the next set of flight tests, in addition to the real-time capturing and recording, the crossing target tracking algorithm was executed concurrently at the rate of 15 frames per second. Several image sequences with the target aircraft crossing the host aircraft were obtained. It was observed that the system successfully detected and tracked the crossing object during the flight tests. Figure 7 shows a trace of the tracking algorithm applied on an image sequence with target aircraft translating from right to left at a distance of 3 nautical miles (5.4 km). The aircraft is located at the end of the track in this image.



Figure 7. Tracking algorithm applied on an image sequence with the target aircraft (marked by a rectangle) translating from right to left at a distance of 3 nmi (5.4 km).

Table 1 summarizes the performance of the crossing target tracking algorithm on a number of image sequences with different distances between the host and the target aircraft. The mis-detection (MD) rate is the ratio of the number of frames in which the target was missed to the total number of frames, whereas the false alarm rate (FA) is the ratio of the total number of false alarms throughout the sequence to the number of image frames in the sequence. The rate of false alarm depends on the amount of clutter in the images, whereas the rate of mis-detection depends on the target size and contrast, and therefore increases with the target distance

Table 1. The performance of the crossing target detection algorithm for image sequences with a number of target distances.

Distance		# Frames	MD rate	FA rate
<i>nmi</i>	<i>km</i>			
1.5	2.78	120	0.061	0.000
1.8	3.33	130	0.113	0.000
2.0	3.70	150	0.394	0.000
2.4	4.44	210	0.059	0.000
3.0	5.55	210	0.056	0.000
4.7	8.70	300	0.335	0.183
5.0	9.26	340	0.803	0.147
5.4	10.00	410	0.643	0.000

in most cases. Since false alarms can be very annoying to the pilots, a low false alarm rate was more desirable than a low mis-detection rate. Hence, the parameters of the algorithms were selected to reduce the false alarm rate, and were same for all the scenarios. It is possible to get a better performance by adjusting parameters according to the characteristics (such as the clutter level) of each scenario.

6 Summary and Future Work

This paper described approaches for detecting obstacles in the flight path of an aircraft in presence of background clutter. Algorithms for detecting objects on a collision course, as well as those crossing the host aircraft were developed. To distinguish collision course objects from background clutter, their translation and expansion in the image were used, whereas for crossing objects, their strength, motion, and motion consistency were used. Detection of crossing objects was implemented on a real time system, and successfully tested on flight tests. The following avenues of future work can be explored.

To estimate image translation in case of collision course objects, compensation for the rotational or vibrational motion of the camera was performed. If the navigation data is unavailable, the image features due to the stationary clutter should be used to perform the compensation, by modeling their image motion.

Image expansion can also be caused by rotation of the target aircraft causing a 'false' expansion in the direction perpendicular to the rotation axis, deforming its shape in the image. On the other hand, the expansion due to a collision course would take place uniformly in all directions without deformation. Measurement of deformation components [7] could be useful for distinguishing between the false expansion and the genuine expansion due to a collision course.

The performance of the crossing object detection system was relatively poor in the cases where the host aircraft rotated about its own axes. To improve the performance, the

image motion due to aircraft rotation should be compensated using navigation data. Alternatively, the background motion should be modeled to separate independent object motion. This could be done using the approach of Irani and Anandan [8] which separates the scene motion into planar and parallax components. However, since the DataCube architecture can perform only simple image processing operations, the procedure would have to be performed on the host machine, using feature based approach.

Finally, the approaches developed here for obstacle detection using visible light images could be combined with those using radar and other sources for design of a complete collision avoidance system.

References

- [1] N. Ancona and T. Poggio. Optical flow from 1-D correlation: Application to a simple time-to-crash detector. *International Journal of Computer Vision*, 14:131–146, 1995.
- [2] J. Arnold, S. Shaw, and H. Pasternack. Efficient target tracking using dynamic programming. *IEEE Trans. on Aerospace and Electronic Systems*, 29(1):44–56, January 1993.
- [3] Y. Baram and Y. Barniv. Obstacle detection by recognizing binary expansion patterns. *IEEE Trans. on Aerospace and Electronic Systems*, 32(1):191–197, January 1996.
- [4] Y. Barniv. Dynamic programming solution for detecting dim moving targets. *IEEE Trans. on Aerospace and Electronic Systems*, 21(1):144–156, January 1985.
- [5] J. S. Bird and M. M. Goulding. Rate-constrained target detection. *IEEE Trans. on Aerospace and Electronic Systems*, 28(2):491–503, April 1992.
- [6] D. Casasent and A. Ye. Detection filters and algorithm fusion for ATR. *IEEE Trans. on Image Processing*, 6(1):114–125, January 1997.
- [7] E. Francois and P. Bouthemy. Derivation of qualitative information in motion analysis. *Image and Vision Computing*, 8(4):279–288, November 1990.
- [8] M. Irani and P. Anandan. A unified approach to moving object detection in 2D and 3D scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(6):577–589, June 1998.
- [9] R. Kasturi, O. Camps, L. Coraor, T. Gandhi, M.-T. Yang, and R. Kasturi. Obstacle detection algorithms for aircraft navigation. Technical Report CSE-00-002, Department of Computer Science and Engineering, Penn State University, January 2000.
- [10] S. S. Krause. *Avoiding Mid-Air Collisions*. TAB books, Mc Graw Hill Inc., Blue Ridge Summit, PA, 1995.
- [11] R. C. Nelson and J. Y. Aloimonos. Obstacle avoidance using flow field divergence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(10):1102–1106, 1989.
- [12] K. Nishiguchi, M. Kobayashi, and A. Ichikawa. Small target detection from image sequences using recursive max filter. In *Proc. SPIE*, volume 2561, pages 153–166, July 1995.
- [13] S. M. Tonissen and R. J. Evans. Performance of dynamic programming techniques for track-before-detect. *IEEE Trans. on Aerospace and Electronic Systems*, 32(4):1440–1451, October 1996.