Detection of Obstacles in the Flight Path of an Aircraft

TARAK GANDHI
MAU-TSUEN YANG
RANGACHAR KASTURI, Fellow, IEEE
OCTAVIA CAMPS, Member, IEEE
LEE CORAOR
Pennsylvania State University
JEFFREY McCANDLESS
NASA Ames Research Center

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 subpixel 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. Approaches are described here 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. For collision course object detection, the image-processing stage uses morphological filter to remove large-sized clutter. To remove the remaining small-sized clutter, differences in the behavior of image translation and expansion of the corresponding features is used in the tracking stage. For crossing object detection, the image-processing stage uses low-stop filter and image differencing to separate stationary background clutter. The remaining clutter is removed in the tracking stage by assuming that the genuine object has a large signal strength, as well as a significant and consistent motion over a number of frames. The crossing object detection algorithm was implemented on a pipelined architecture from DataCube and runs in real time. Both algorithms have been successfully tested on flight tests conducted by NASA.

Manuscript received September 20, 2001; revised August 15, 2002; released for publication September 26, 2002.

IEEE Log No. T-AES/39/1/808651.

Refereeing of this contribution was handled by P. M. Willett.

This research was supported in part through Grant NAG2-1152 by NASA Ames Research Center.

Authors' current addresses: T. Gandhi, University of California, San Diego; M-T. Yang, Dept. of Computer Science and Information Engineering, National Dong-Hwa University, Taiwan; R. Kasturi and L. Coraor, Dept. of Computer Science and Engineering, Pennsylvania State University, 220 Pond Labs, University Park, PA 16802-6106; O. Camps, Dept. of Electrical Engineering, Pennsylvania State University, University Park, PA 16802; J. McCandless, Flight Management and Human Factors Division, MS 262-2, NASA Ames Research Center, Moffett Field, CA 94035, E-mail: (kasturi@cse.psu.edu).

I. 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. [17] 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 [5] and Arnold, et al. [2] to detect moving objects of small size. The theoretical performance of this approach was characterized by Tonissen and Evans [19].

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 [11] 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 [7] 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.

We have used the approaches described here to detect objects on collision course, as well as those crossing the host aircraft [9, 10, 13]. At present, the detection of collision course and crossing objects are separately implemented. However, the algorithms can be applied to the same image sequence to detect both kinds of objects simultaneously. 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.

Section II describes the flight maneuvers used to get image sequences for both types of objects. Section III describes the theory used for discriminating collision course objects from background clutter, using difference in the translation and expansion of the corresponding image features. Section IV and V describe the steps used to detect the collision course objects and crossing objects, respectively. Results on real image sequences are shown in the respective sections. The detection of collision course objects was performed off-line on real image sequence obtained from an aircraft. The detection of crossing objects was implemented in real time using DataCube, mounted on an aircraft, and tested during flight tests conducted by NASA. Section VI concludes this work, and explores avenues for future work.

II. 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 Fig. 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,

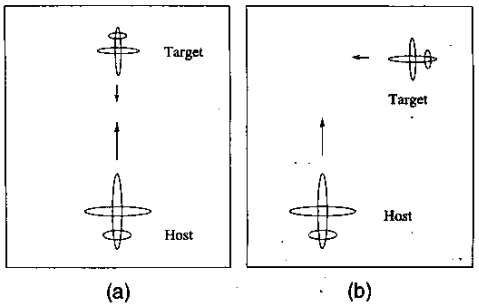


Fig. 1. (a) Collision course object maneuver: target aircraft flies towards host aircraft. (b) Crossing object maneuver: target aircraft flies perpendicular to host aircraft.

whereas in the other, the background was severely cluttered and the elimination of this clutter required the use of feature motion behavior, as described here. Special purpose image-processing hardware required for real-time obstacle detection was not available during these flights and hence the focus was on capturing video data for off-line processing.

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 here.

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. Video was captured using a Kodak Megaplus Model 1.0 digital camera with a 1,024×1,024 pixels resolution. DataCube MaxPCI system with real-time image-processing capabilities was installed on the host aircraft and the collision course object detection and tracking algorithm was tested during these flights. In a final system incorporating these algorithms, both collision course and crossing course object detection algorithms would be running in parallel and both classes of hazards would be presented to the pilots.

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.

III. 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 [14]. 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 [16] use the image expansion in terms of the flow field divergence to estimate the time to collision, for separating obstacles. Ringach and Baram [18] use the normal flow, along with the object boundary information to estimate the flow field divergence. 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 signs of image spatial and temporal derivatives in a feedforward neural network. Baram, Barniv, and Sony [4] improve on the above by using gray level derivatives instead of signs, which reduces the complexity of the neural network. The network learns the probability density functions of the data, corresponding to safe and dangerous categories, which can be used for classification of test data. Francois and Bouthemy [8] separate the image motion into components of divergence, rotation, and deformation. Meyer and Bouthemy [15] obtain maximum likelihood estimation of these components using normal flow measurements. Ancona and Poggio [1] use 1-D correlation to estimate optical flow for a time-to-crash detector. Most of these methods are useful for objects of larger sizes. However, in our case, the object sizes can be very small, even subpixel, 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.

In the following sections, the rates of translation and expansion for a collision course object as well as background clutter are derived. It will be seen that a collision course object usually has a small rate of translation and a large rate of expansion compared with the background clutter. However, the rate of translation also depends on the distance, and far away clutter, lying in upper parts of the image can have less rate of translation than a nearby object. On the other hand, nearby clutter in the lower parts of the image could have large rate of expansion. Hence, the conditions are derived under which the object has a lower translation rate and higher expansion rate compared with the clutter.

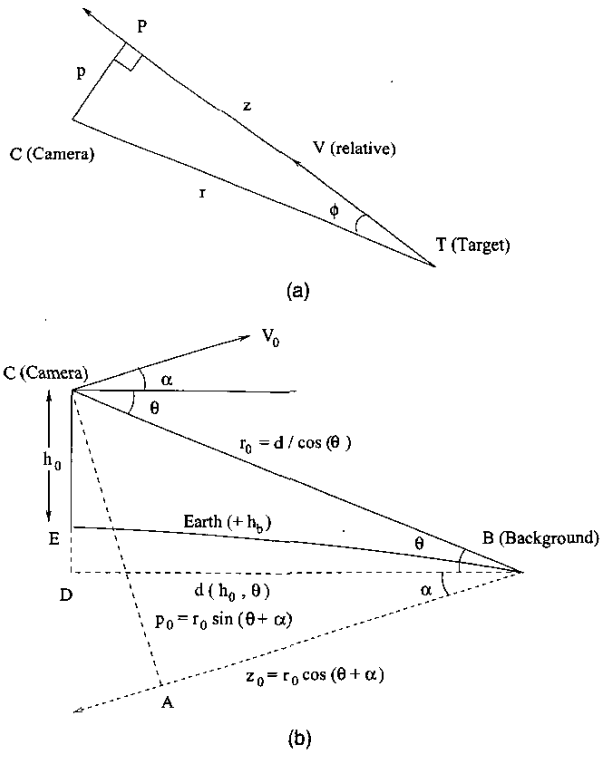


Fig. 2. Geometry of (a) target (b) background moving relative to camera.

Consider an object approaching towards the aircraft with a relative velocity of V as shown in Fig. 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 Fig. 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.

A. Target Translation

As seen from Fig. 2(a), when the object moves, the angle ϕ as well as distances r and z change, but the distance of passage p is constant. The rate of angular translation of an object in the image is $T = \dot{\phi}$. The pixel translation is approximately given by multiplying the angular translation by the focal length. From Fig. 2(a), we have

$$z = p\cot\phi = r\cos\phi. \tag{1}$$

The magnitude of relative velocity V is the rate of decrease of z, given by

$$V = -\dot{z} = p(\csc^2\phi)\dot{\phi} = p(r/p)^2\dot{\phi}. \tag{2}$$

Also, the time to passage is given by

$$\tau = z/V = r\cos\phi/V. \tag{3}$$

From (2) and (3), the rate of target translation is given by

$$T = \dot{\phi} = \frac{pV}{r^2} = \frac{p\cos\phi}{\tau r}. \tag{4}$$

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 rate of translation compared with other objects. However, this rate is also dependent on the target distance, and a nearer target moves faster in the image than a farther target with the same distance of passage. If $S_{\min}$ is the smallest visible dimension that an object can have, the corresponding size s in the image is given by

$$s \geq s_{\min} = S_{\min}/r. \tag{5}$$

Hence, from (4), one can write

$$\frac{T}{s} \leq \frac{p\cos\phi}{\tau S_{\min}} \leq \frac{p}{\tau S_{\min}}. \tag{6}$$

Hence, an object on a near collision course, having sufficient time before imminent collision has the ratio of its image motion to its image size bounded by the above precomputable limit. For example, if the distance of passage of $p = 150$ m (500 ft) is allowed, and an object of smallest size of $S_{\min} = 1.2$ m (4 ft) is to be detected before $\tau = 25$ s (750 frames), then this ratio becomes 1/6, i.e., the image motion per frame is at the most 1/6th of the image size of the object. However, in actual practice, a larger range of velocities should be checked, to have a safety margin.

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 the data from the aircraft navigation system. In the absence of this data, it may be possible to use image features due to clutter (if available) to perform the compensation, by modeling their image motion due to camera rotation.

If this compensation is successful, the velocity-to-size ratio of the object would be bounded. By reducing the image resolution to an appropriate level, the image velocity of the object would also be restricted. Hence, using pyramid construction, target detection can be performed at a number of resolutions, and the suitable resolution selected. This also leads to spatio-temporal integration of the image data and the amplification of signal-to-noise ratio (SNR) which could enable detection of subpixel or low-contrast objects in uniform background, such as clear or overcast sky.

B. Target Expansion

Another discriminating feature between objects on collision course, and objects much farther, is the time to collision. It is well known that the rate of image expansion, i.e., the increase of the image size of an object, is inversely proportional to the time to collision.

In Fig. 2(a), as the object comes closer to the camera along the line of z, its size in the image will become larger. The rate of this expansion of any object is defined as the ratio of the rate of increase in its size to the size at that time, i.e., $E = \dot{s}/s$—where s is the size of the object in the image. Since $s = S/r$ where S is the object size which is assumed constant, we have $\dot{s} = -S\dot{r}/r^2$, and

$$E = -\dot{r}/r. \tag{7}$$

By geometry of Fig. 2(a),

$$r^2 = z^2 + p^2. \tag{8}$$

To find the rate of expansion, this expression is differentiated to yield:

$$2r\dot{r} = 2z\dot{z} = -2zV. \tag{9}$$

Hence, rate of target expansion is given by

$$E = -\frac{\dot{r}}{r} = \frac{zV}{r^2} = \frac{V\cos\phi}{r} = \frac{\cos^2\phi}{\tau} \tag{10}$$

where the time to passage is

$$\tau = z/V = r\cos\phi/V. \tag{11}$$

For $\tau = 25$ s = 750 frames, the ratio is 0.13% per frame, which is a very small magnitude. This small expansion can be measured by tracking it over a large number of frames.

C. Background Translation

The relationship between 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 also decreases with the object distance. Thus, if clutter is at a large distance, it too could have a small image motion. The conditions under which an object on the collision course can be distinguished from ground clutter at the same image position are derived below.

Let r_0 and p_0 denote the background distance, and the minimum distance of approach for the background, respectively, as shown in Fig. 2(b). The relative velocity V_0 between the camera and the background is actually the magnitude of the camera velocity. By substituting these parameters in (4), the rate of background translation can be written as

$$T_0 = \frac{p_0 V_0}{r_0^2}. \tag{12}$$

Let $h_0 = h_c - h_b$ denote the difference between the camera altitude h_c and the background altitude h_b.

Also, the angle of the camera velocity above the horizontal (not horizon) is α. In Fig. 2(b), $d = \overline{DB}$ is a function of the relative height $h_0 = \overline{EC}$ and the angle $\theta = \angle DBC$. If the Earth were flat (or θ is large), and the terrain is smooth, the dotted line corresponding to d would coincide with the surface of the Earth, and $d = h_0 \cot\theta$. However, if we express

$$d(h_0,\theta) = h_0 \cot\theta f(h_0,\theta) \tag{13}$$

then the effects of the Earth's curvature would be incorporated in the function f derived in the Appendix as

$$f(h_0,\theta) = \frac{2}{1+\sqrt{1-2h_0/(R\tan^2\theta)}}. \tag{14}$$

If the Earth's curvature can be neglected, then $f \simeq 1$. On the other hand, if the object is on the horizon, it can be shown that $f = 2$.

Using Fig. 2(b) geometry with (12), the rate of background translation T_0 is given by

$$T_0 = \frac{V_0 \sin(\theta+\alpha)}{r_0} = \frac{V_0 \sin(\theta+\alpha)\sin\theta}{h_0 f(h_0,\theta)} \tag{15}$$

with

$$r_0 = h_0 \csc\theta f(h_0,\theta). \tag{16}$$

If the hazard is to be discriminated from the background in the same line of sight, the rate of translation of the hazard must be much smaller than that of the background, i.e., $T \le \eta_t^{-1} T_0$ with $\eta_t > 1$, having a larger value for greater discriminating power. Using (4) and (15), we have

$$\frac{p\cos\phi}{\tau r} \le \eta_t^{-1} \frac{V_0 \sin(\theta+\alpha)\sin\theta}{h_0 f(h_0,\theta)}. \tag{17}$$

Hence, the object distance r should be larger than the following expression:

$$r \ge \frac{\eta_t p h_0 f(h_0,\theta)\cos\phi}{\tau V_0 \sin(\theta+\alpha)\sin\theta} = \frac{\eta_t p D f(h_0,\theta)\sqrt{1-Q^2}}{\sin(\theta+\alpha)\sin\theta} \tag{18}$$

with

$$D = \frac{h_0}{\tau V_0}, \qquad Q = \frac{p}{r} = \sin\phi, \tag{19}$$

$$\cos\phi = \sqrt{1-Q^2} \simeq 1 \qquad (\text{for } p \ll r).$$

Hence, θ should satisfy

$$\sin(\theta+\alpha)\sin\theta \ge \eta_t D Q \sqrt{1-Q^2} f(h_0,\theta). \tag{20}$$

Also, using $T \le \eta_t^{-1} T_0$, with equations (4) and (15), one can write

$$\frac{p\cos\phi}{\tau r} \le \eta_t^{-1} \frac{V_0 \sin(\theta+\alpha)}{r_0}. \tag{21}$$

Since the object distance cannot be greater than the background distance in the line of sight, $r \le r_0$. Hence, one can also write

$$\sin(\theta+\alpha) \ge \frac{\eta_t p\cos\phi}{\tau V_0}\frac{r_0}{r} \ge \frac{\eta_t p\sqrt{1-Q^2}}{\tau V_0}. \tag{22}$$

For $p \ll r$ or $Q \ll 1$, this condition is approximately independent of r. 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, provided it is sufficiently large.

If the curvature of the Earth can be neglected, then $f \simeq 1$. The necessary condition in (22) does not simplify. However, (20) reduces to

$$\sin(\theta+\alpha)\sin\theta \ge \eta_t D Q \sqrt{1-Q^2}. \tag{23}$$

On solving for θ, this yields

$$\theta \ge \tfrac{1}{2}[\cos^{-1}(-2\eta_t D Q\sqrt{1-Q^2} + \cos\alpha) - \alpha]. \tag{24}$$

For example, if we have

$$\begin{aligned} &p = 150 \text{ m}, \quad \tau = 25 \text{ s}, \quad V_0 = 150 \text{ m/s}, \\ &h_0 = 1 \text{ km}, \quad \alpha = 0, \quad \eta_t = 2.5 \end{aligned} \tag{25}$$

for these values, $D = 0.267$, and from (22) the necessary condition is $\theta \ge 5.7°$. This condition corresponds to the target being at the same position as the background, which is $r = r_0 = 10$ km $\simeq 5.4$ nmi or $Q = 0.015$, using (16). However, if the target is nearer, the condition on θ is determined by (20) or (23). For example, if a hazard should be detected at $r = 5$ km $\simeq$ 2.7 nmi or $Q = 0.03$, one would really need $\theta \ge 8.1°$. The required θ increases as r decreases.

D. Background Expansion

For estimating the rate of expansion of the background, the corresponding parameters for the background are substituted in (10) to give

$$E_0 = \frac{z_0 V_0}{r_0^2}. \tag{26}$$

From Fig. 2(b) and (13), the rate of background expansion can be written as

$$\begin{aligned} E_0 &= \frac{V_0\cos(\theta+\alpha)}{r_0} = \frac{V_0\cos(\theta+\alpha)\cos\theta}{d} \\ &= \frac{V_0\cos(\theta+\alpha)\sin\theta}{h_0 f(h_0,\theta)}. \end{aligned} \tag{27}$$

If reliable discrimination of the hazard from the background in the same line of sight is required, the rate of expansion of the hazard must be much larger than that of the background, i.e., $E \ge \eta_e E_0$ with $\eta_e > 1$, having a large value for greater discriminating power. Using (10) and (27), one needs

$$\frac{\cos^2\phi}{\tau} \ge \eta_e \frac{V_0\cos(\theta+\alpha)\sin\theta}{h_0 f(h_0,\theta)} \tag{28}$$

or

$$\begin{aligned} \cos(\theta+\alpha)\sin\theta &\le \frac{h_0 f(h_0,\theta)\cos^2\phi}{\eta_e \tau V_0} \\ &= \eta_e^{-1} D(1-Q^2) f(h_0,\theta) \end{aligned} \tag{29}$$

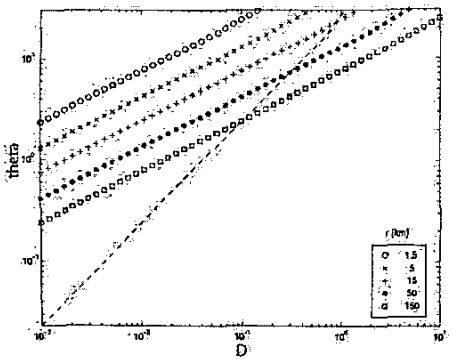


Fig. 3. Plots for detection using translation and expansion. Plots of required θ in degrees against $D = h_0/(\tau V_0)$ in meters. Maximum θ for detection using expansion is shown by dashed line, and is independent of target distance. Other curves show minimum θ required for detection using translation for various values of target distance r in km, for the distance of passage $p = 150$ m.

where D and Q are given by (19). For the case of $f \simeq 1$, the (29) reduces to

$$\cos(\theta + \alpha)\sin\theta \leq \eta_e^{-1} D(1 - Q^2). \tag{30}$$

Explicit solution for θ is then given by

$$\theta \leq \tfrac{1}{2}[\sin^{-1}(2\eta_e^{-1} D(1 - Q^2) + \sin\alpha) - \alpha]. \tag{31}$$

For the conditions in (25), with $\eta_e = 2.5$ and $Q \ll 1$, we need $\theta \leq 6.2°$ for reliable detection using expansion.

E. 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 Fig. 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 $p = 150$ 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.

IV. 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 uses 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 a lot of background clutter. The approach successfully discriminates the object from the clutter. The running time for the current implementation is approximately 7 s 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.

A. Image Processing Stage

A morphological filter [7] 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 nonnegative, 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 [7]. 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

$$F_+ = F - \max\{F \circ m_x, F \circ m_y\} \tag{32}$$

$$F_- = -F + \min\{F \bullet m_x, F \bullet m_y\} \tag{33}$$

where F is the original image, and $\circ$ and $\bullet$ denote morphological opening and closing operations, respectively. Nonmaximal suppression was performed on the outputs F_+ and F_- of the filters, and pixels exceeding a threshold were sent as features to the tracking stage.

B. 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

(a)

(b)

Fig. 4. (a) First and (b) last frames in image sequence used for target detection. Target aircraft, not visible in (a), is marked by a rectangle in (b).

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 subimage corresponding to the window was thresholded, and the connected component containing the center of the window was found. All the pixels in the subimage that did not belong to the component were set to zero. The subimage was convolved with a number of smoothing masks. These masks perform matched filtering with object templates corresponding to 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.

C. 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 $\tau \simeq 25$ s, (19) gives $D = 0.329$. If $p = 150$ m $\simeq 500$ ft is allowed, and $\eta_t = \eta_e = 2.5$, then the necessary condition for reliable detection using translation by (22) is $\theta \geq 10.5°$. Reliable detection using expansion requires $\theta \leq 7.6°$ using (29). 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, $\tau = 100$ s gives $D = 0.0822$, and the conditions for

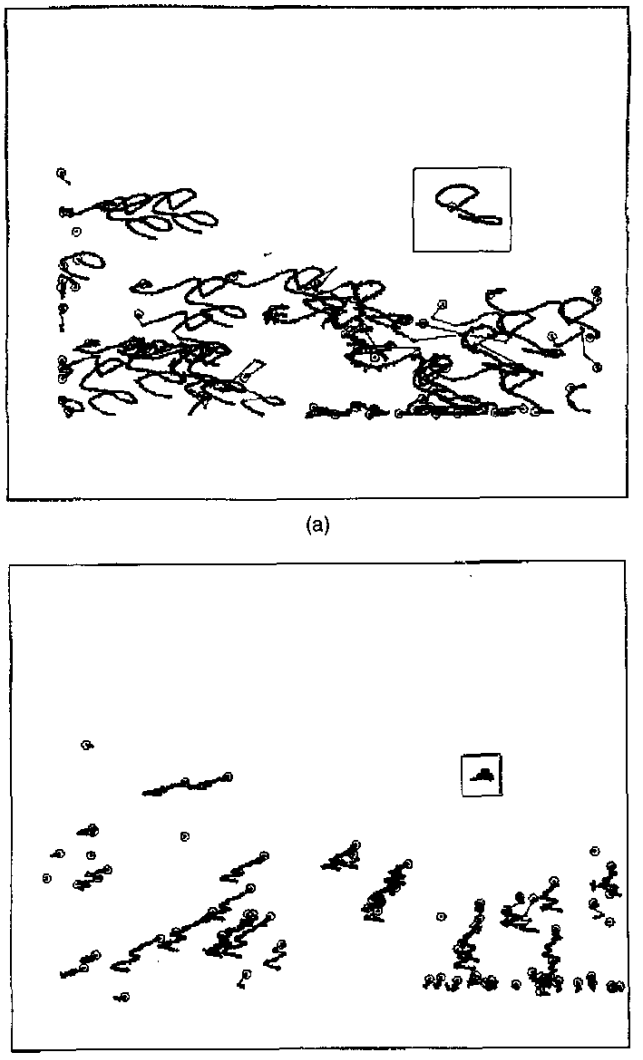


Fig. 5. Feature tracks (a) before (b) after compensation. Compensation refers to the rotation compensation using navigation. Target aircraft is marked by a rectangle.

reliable detection with translation and expansion would change to $\theta \geq 2.6°$ and $\theta \leq 1.9°$, respectively. In practice, it was observed in this case that both translation and expansion could separate the collision course aircraft from most clutter.

Fig. 4(a) and (b) show the first and the last image frames in the 482 frame (16 s) sequence used for target detection. Fig. 5(a) and (b) show all the target tracks before and after motion compensation, respectively. The target track is shown separately in Fig. 6(a) and (b). Fig. 6(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 Figs. 6(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 Fig. 7. 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). This ground based feature has a small time to passage but large distance of passage, and therefore does not pose danger to the aircraft.

V. CROSSING OBJECT DETECTION

In addition to the detection of objects on a collision course, it is useful to monitor the objects

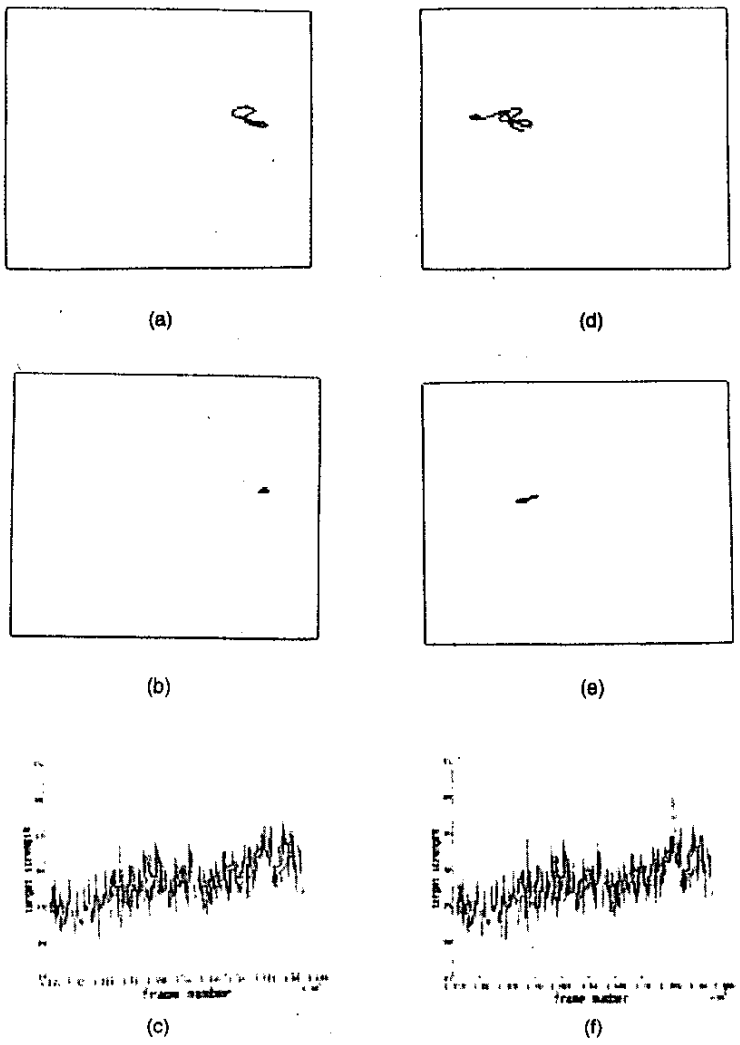


Fig. 6. Translation and expansion for typical target and clutter tracks. (a) Target track. (b) Target track after compensation. (c) Plot of expansion of target track against frame number. (d)–(f) Corresponding plots for clutter track. Target track has smaller rate of translation and larger rate of expansion.

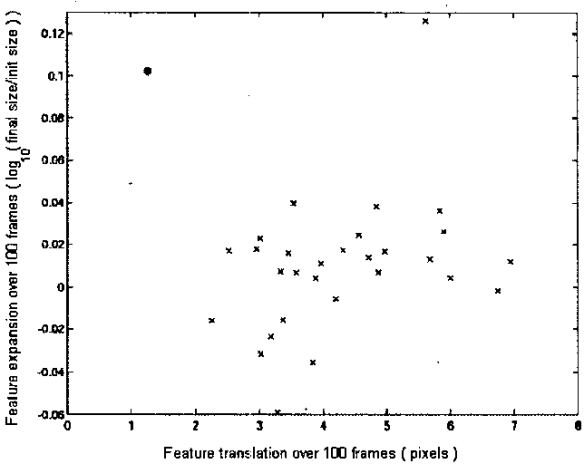


Fig. 7. Scatter plot of feature expansion against translation. Target marked as encircled asterisk, and is in upper left corner, having small rate of translation and large rate of expansion.

which are crossing the aircraft. For this purpose, a system was designed to specifically detect objects having a translational motion in the image. To distinguish translating objects from ground or cloud clutter, the following criteria were used.

1) The object should have sufficient signal strength.

2) The object should have an image velocity greater than a threshold.

3) The object should have a consistent motion, i.e., its velocity must not change abruptly.

Note that the clutter is expected to have smaller motion than a crossing object but larger motion and smaller expansion than collision course object. Hence, different algorithms are needed to separate clutter from crossing and collision course objects.

The system to detect translating objects has been implemented on the pipelined image-processing system, the DataCube MaxPCI described in [9, 13] to obtain real time performance. The system was mounted on the Air Force Total In-Flight Simulator (TIFS/NC1314) aircraft, and flight tests were conducted by NASA with another aircraft flying in front of it. The detection and tracking of the target aircraft were demonstrated during the flight test.

This system is divided into two stages, an image-processing stage and a tracking stage. The image-processing stage removes most of the clutter, and isolates potential features which could be translating objects. This stage involves repetitive image operations such as convolution, pointwise operations, histograms, etc. which are suitable for a pipelined architecture, and can be performed in integer format. Hence, these steps are implemented on the DataCube machine. The output of this stage is a list of image features which are likely to contain the target objects, including their positions and the signal strengths. However, the list may also contain features corresponding to background clutter, which are not separated by the simple image-processing steps. The tracking stage tracks these features to distinguish the genuine translating objects from background clutter using the criteria mentioned above. Since the image-processing stage has reduced the volume of data to be operated on, more complicated target tracking algorithms can be implemented even on the host PC associated with the DataCube. The threshold used in the image-processing stage is adjusted dynamically to give a nearly constant number of features for the tracking stage so that they can be processed in real time using the slower host. This matching of the output rate of one stage to the input rate of the next stage is known as the rate constraint criterion [6].

A. Image-Processing Stage

This stage performs the basic image-processing steps to suppress clutter and extract features which could potentially be crossing targets.

1) *Resolution Reduction*: The resolution of the image is reduced so that the system is capable of operation in real time. The image is convolved with a 5×5 low-pass filter mask and then down-sampled by two in both horizontal and vertical directions. Low-pass filtering suppresses high frequencies, which would otherwise have been aliased to low frequencies by the down-sampler. Although the image resolution is reduced, the SNR is actually enhanced. This is because the target size is usually greater than 2 pixels, leading to spatial integration of the target contrast.

2) *Low-Stop Filtering*: A low-stop filter is applied to the reduced image to suppress background clutter. The filter is formed by subtracting a low-pass filter of larger size (9×7) from a low-pass filter of smaller size (3×5). This step suppresses uniform background intensity and weak clutter corresponding to low frequencies, and also performs spatial integration for larger objects. A rectangular mask is used since the targets are expected to have a greater width than height.

3) *Image Differencing*: Image differencing is performed on the low-stop filtered images by subtracting consecutive frames. This is equivalent to a low-stop filter in temporal direction. Since the object is assumed to be translating, image differencing suppresses stationary objects corresponding to background clutter. It should be noted that steps 1 to 3 are theoretically interchangeable, since they are all linear filters. However, since these operations are performed with integer arithmetic of limited precision, the particular order of the steps is used to reduce the truncation error.

4) *Nonmaximal Suppression*: Directly using the output of the previous step would give rise to a large number of features for an extended target. Nonmaximal suppression is performed to get a single feature (or sometimes a small number of features) for the entire target. Pixels can have both positive or negative values corresponding to bright and dark targets, respectively. Hence, an absolute value image is first formed, and every pixel which is not a local maximum in its 3×3 neighborhood is marked. The marked pixels are set to zero in the original image, i.e., the image before taking the absolute values.

5) *Histogram Formation*: To extract candidate features, the output from the above steps should be thresholded. Furthermore, the threshold should be chosen so that the number of features does not overload the tracking stage. Hence, the threshold is selected so that the number of pixels exceeding the threshold is less than or equal to a fixed rate which matches the operation speed of the tracking stage. For this purpose, a histogram of the values of

the nonmaximal suppressed image is constructed. The threshold then is determined as the smallest pixel value for which the number of elements in the histogram bins above this value does not exceed the fixed rate. Applying this value as the threshold would then ensure that the number of features remains bounded.

6) *Thresholding and Feature Output*: Pixels in the image with the output value greater than the threshold are separated as features. The feature positions and amplitudes are transmitted to the tracking stage.

B. Tracking Stage

This stage maintains a list of tracks containing the frame number, unique ID, position, velocity, and amplitude. The list is empty in the beginning. The following steps are repeated for every frame for which the list of features is received from the image-processing stage.

1) *Track Update*: For each track in the list of tracks, the list of features is scanned to obtain features in a neighborhood window around the track position. If one or more such features are found, the one with the largest amplitude is selected as the continuation of the track. Using the coordinates (z_1,z_2) of this feature, as well as the current track position (x_1,x_2) and velocity (u_1,u_2), the expected position and velocity for the next frame is estimated using a Kalman filter. The filter is applied separately for horizontal $(i = 1)$ and vertical $(i = 2)$ directions. For each direction, the state vector is given by $X_i = [x_i \;\; u_i]^t$, and the observation is the feature coordinate z_i. The track life n of the track is the number of frames in which the target has been observed, with adjustments made in the frames where the target is not observed. The measurement update is given by

$$\begin{aligned} x_i^+(n) &= x_i(n) + K_1(n)(z_i - x_i) \\ u_i^+(n) &= u_i(n) + K_2(n)(z_i - x_i). \end{aligned} \tag{34}$$

The state update is given by

$$\begin{aligned} x_i(n+1) &= x_i^+(n) + u_i^+(n) \\ u_i(n+1) &= u_i^+(n). \end{aligned} \tag{35}$$

The Kalman filter matrix $K(n) = [K_1(n) \;\; K_2(n)]^t$ is precomputed using the inverse covariance formulation of the Kalman filter. The computation is performed for a number of $n = 1 \ldots N$, where N is large enough so that $K(N)$ does not change significantly with N.

The track amplitude is updated using recursive averaging according to the following equation:

$$F(n+1) = f(n) + \alpha F(n) \tag{36}$$

where $F(n)$ and $F(n+1)$ are the track amplitudes for the current and next frames, $f(n)$ is the feature amplitude, and α is the forgetting factor. The track life n is incremented by one.

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. Theoretically, this would mean that the values of the Kalman filter matrix would have to be recomputed. To avoid such a computation, the value of the track life n is reduced by a factor to approximately simulate the effect of having "lost track" of the feature. The feature amplitude is updated using $f(n) = 0$ in (36).

2) *Formation of New Tracks*: 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 with its track life $n = 1$. Its position (x_1,x_2) will be the same as feature position (z_1,z_2), whereas velocity (u_1,u_2) is initialized to zero. The actual velocity will be computed only in the next frame.

3) *Pruning the List of Tracks*: If the number of tracks is too large, the stage can get overloaded and fail to operate in real time. To eliminate this possibility, if the number of tracks are greater than a particular number, the weakest tracks are deleted.

4) *Merging Similar Tracks*: It may happen that two or more tracks may be formed corresponding to the same object. Hence, tracks which are very close to each other and have nearly the same velocity are merged, retaining the one with the larger track amplitude.

5) *Output*: Tracks which satisfy the criteria of the object, including having an amplitude larger than a threshold, as well as a significant and consistent motion are outputted as potential objects.

C. 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/s. 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/s. 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. Fig. 8 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 nmi (5.4 km). The aircraft is located at the end of the track in this image.

Table I summarizes the performance of the crossing target tracking algorithm on a number of image sequences with different distances between

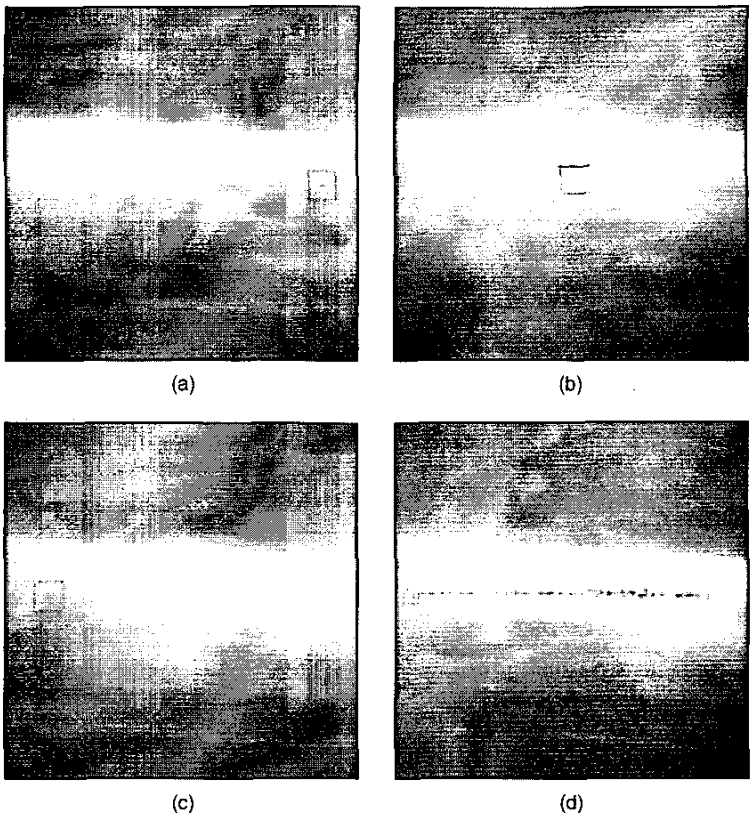


Fig. 8. Tracking algorithm applied on image sequence with target aircraft crossing from right to left at distance of 3 nmi (5.4 km). (a)–(c) Three frames from sequence. (d) Detected track of target aircraft.

TABLE I
Performance of the Crossing Target Detection Algorithm for Image Sequences Corresponding to a Number of Target Distances

Distance nmi	Distance km	# Frames	MD Rate	FA Rate
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

Note: The false alarm (FA) rate is the ratio of the total number of false alarms throughout the sequence to the number of image frames in the sequence. 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.

the host and the target aircraft. The false alarm rate is measured as the ratio of the total number of false alarms throughout the sequence to the number of image frames in the sequence. The mis-detection rate is measured as the ratio of the number of frames in which the target was missed to the total number of frames. The false alarm rate depends on the amount and motion of clutter in the images, whereas the mis-detection rate depends on the target size and contrast, and therefore increases with the target distance 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.

VI. 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. The image translation and expansion of an object on a near collision course were expressed in terms of parameters such as the distance of passage, time to passage, object distance, and relative velocity between the host aircraft and the obstacle. Corresponding expressions for ground objects (not on collision course) were

derived in terms of aircraft height, aircraft velocity, angle of inclination, and other parameters. Conditions under which image translation and expansion can be used to discriminate collision course object from background clutter were derived. Morphological filter was used for extracting image features corresponding to potential target, while removing most background clutter. The features were then tracked over a large number of frames to estimate their translation and expansion, which were used to remove remaining clutter. The results were shown on a real image sequence captured from a camera mounted on an aircraft, with another aircraft flying towards it.

In case of crossing objects, the target strength, motion, and motion consistency were used to discriminate genuine targets from clutter. The image-processing stage consisted of simple operations, such as image differencing, low-stop filtering, nonmaximal suppression, histogram formation, and thresholding which were implemented on a pipelined image-processing system called DataCube MaxPCI. The features obtained after these operations were tracked over a large number of frames to estimate the target strength, motion, and consistency. Since the number of features is small the tracking stage could be implemented on the host machine associated with DataCube MaxPCI. The whole system for detection of crossing objects was implemented in real time, and successfully tested on several real image sequences during flight tests.

The following avenues for 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. It should be noted that the camera motion could produce images having subpixel displacements relative to each other, with a possibility of obtaining higher resolution images by combining multiple images. This process is known as superresolution [12].

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 [8] 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 would be 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 [11] 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.

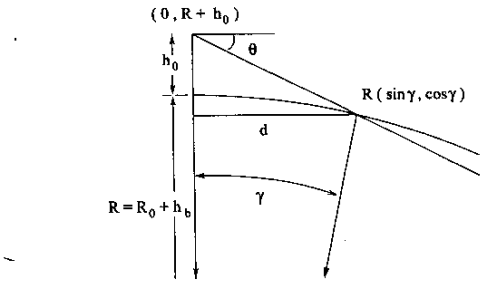


Fig. 9. Geometry of Earth's curvature. Coordinates used are with respect to Earth's center.

At present, only the crossing object detection is implemented on DataCube, while the collision course object detection was tested offline with available image sequences. When adequate computational resources, are available, the collision course algorithm can be implemented on DataCube in similar manner. Image processing can be done on pipelined architecture, while tracking and discrimination can be done on host machine. Both algorithms can be performed simultaneously on the images to detect crossing as well as collision course objects. If an object changes trajectory from crossing to collision course, the crossing object algorithm would lose track and collision course algorithm would pick it up. However there may be a time when none or both of the algorithms would be tracking the object. This would depend on the clutter level and whether an object in intermediate course can be reliably distinguished from clutter. Better clutter elimination would have to be developed if one wants to track objects in such course.

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.

APPENDIX: EFFECT OF HORIZON

The function describing the effect of the curvature of the Earth is calculated here, neglecting the effects of refraction. Fig. 9 shows the geometry of the Earth's curvature. The coordinates used are with respect to the Earth's center. Using this, we have

$$d = R\sin\gamma, \qquad d\tan\theta = h_0 + R(1-\cos\gamma) \simeq h_0 + d^2/(2R) \tag{37}$$

where $R = R_0 + h_b$, h_b is the altitude of the background, R_0 is the radius of Earth, and γ is the angle subtended on the center of the Earth by the triangle. Solving this equation yields

$$d = R\left[\tan\theta \pm \sqrt{\tan^2\theta - 2h_0/R}\right]. \tag{38}$$

The correct solution is the smaller value of d, since the larger value represents the other intersection of the line of sight with the Earth.

$$d = R\left[\tan\theta - \sqrt{\tan^2\theta - 2h_0/R}\right]$$

$$= \frac{2h_0}{\tan\theta + \sqrt{\tan^2\theta - 2h_0/R}}. \tag{39}$$

By substituting in (13), we have

$$f(h_0,\theta) = \frac{2}{1 + \sqrt{1 - 2h_0/(R\tan^2\theta)}}. \tag{40}$$

If $\theta \simeq \pi/2$, or R is large, h is small, then $f \simeq 1$, i.e., the Earth's curvature can be neglected. However, where the line of sight just touches the Earth, i.e., at the horizon, the discriminant under the square root is zero, then $f = 2$ and the corresponding θ is

$$\theta_h = \tan^{-1}\sqrt{2h_0/R}$$

Any value of θ smaller than this value corresponds to the line of sight not touching the Earth, i.e., background above the horizon.

ACKNOWLEDGMENTS

This research was supported in part through Grant NAG2-1152 by NASA Ames Research Center. The authors would like to acknowledge the technical support provided by Datacube MaxPCI engineers and the crew and staff of U.S. AirForce Total In Flight Simulator (TIFS) aircraft. We also thank Himanshu Vajaria for his help in the preparation of this paper.

REFERENCES

[1] Ancona, N., and Poggio, T. (1995)
Optical flow from 1-D correlation: Application to a simple time-to-crash detector.
International Journal of Computer Vision, **14** (1995), 131–146.

[2] Arnold, J., Shaw, S., and Pasternack, H. (1993)
Efficient target tracking using dynamic programming.
IEEE Transactions on Aerospace and Electronic Systems, **29**, 1 (Jan. 1993), 44–56.

[3] Baram, Y., and Barniv, Y. (1996)
Obstacle detection by recognizing binary expansion patterns.
IEEE Transactions on Aerospace and Electronic Systems, **32**, 1 (Jan. 1996), 191–197.

[4] Baram, Y., Barniv, Y., and Sony, T. (1997)
Detecting collision from gray-level expansion by a neural network.
Neurocomputing, **16** (1997), 77–84.

[5] Barniv, Y. (1985)
Dynamic programming solution for detecting dim moving targets.
IEEE Transactions on Aerospace and Electronic Systems, **21**, 1 (Jan. 1985), 144–156.

[6] Bird, J. S., and Goulding, M. M. (1992)
Rate-constrained target detection.
IEEE Transactions on Aerospace and Electronic Systems, **28**, 2 (Apr. 1992), 491–503.

[7] Casasent, D., and Ye, A. (1997)
Detection filters and algorithm fusion for ATR.
IEEE Transactions on Image Processing, **6**, 1 (Jan. 1997), 114–125.

[8] Francois, E., and Bouthemy, P. (1990)
Derivation of qualitative information in motion analysis.
Image and Vision Computing, **8**, 4 (Nov. 1990), 279–288.

[9] Gandhi, T. (2000)
Image sequence analysis for object detection and segmentation.
Ph.D. dissertation, Pennsylvania State University, 2000; approved for publication.

[10] Gandhi, T., Yang, M-T., Kasturi, R., Camps, O., Coraor, L., and McCandless, J. (2000)
Detection of obstacles in the flight path of an aircraft.
In *Proceedings of IEEE Computer Vision and Pattern Recognition* (2000).

[11] Irani, M., and Anandan, P. (1998)
A unified approach to moving object detection in 2D and 3D scenes.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **20**, 6 (June 1998), 577–589.

[12] Irani, M., and Peleg, S. (1991)
Improving resolution by image registration.
CVGIP: Graphical Models and Image Processing, **53**, 3 (May 1991), 231–239.

[13] Kasturi, R., Camps, O., Coraor, L., Gandhi, T., and Yang, M-T. (2000)
Obstacle detection algorithms for aircraft navigation.
Technical report CSE-00-002, Dept. of Computer Science and Engineering, Penn State University, Jan. 2000.

[14] Krause, S. S. (1995)
Avoiding Mid-Air Collisions.
Blue Ridge Summit, PA: TAB books, Mc-Graw Hill, 1995.

[15] Meyer, F., and Bouthemy, P. (1992)
Estimation of time-to-collision maps from first order motion models and normal flows.
In *Proceedings of the International Conference on Pattern Recognition*, 1992, I:78–82.

[16] Nelson, R. C., and Aloimonos, J. Y. (1989)
Obstacle avoidance using flow field divergence.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **11**, 10 (1989), 1102–1106.

[17] Nishiguchi, K., Kobayashi, M., and Ichikawa, A. (1995)
Small target detection from image sequences using recursive max filter.
In *Proceedings of SPIE*, **2561** (July 1995), 153–166.

[18] Ringach, D. L., and Baram, Y. (1994)
A diffusion mechanism for obstacle detection from size-change information.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **16**, 1 (Jan. 1994), 76–80.

[19] Tonissen, S. M., and Evans, R. J. (1996)
Performance of dynamic programming techniques for track-before-detect.
IEEE Transactions on Aerospace and Electronic Systems, **32**, 4 (Oct. 1996), 1440–1451.

Tarak Gandhi was born in 1970 in Mumbai (Bombay), India. He received his Bachelor of Technology degree in computer science and engineering at the Indian Institute of Technology, Bombay, India in 1991. He received the M.S. degree in computer engineering in 1995, and the Ph.D. in computer science and engineering in 2000, both from Pennsylvania State University, University Park.

He worked at Adept Technology, Inc., City of Industry, CA from 2000 to 2002. He is currently working as a postgraduate researcher at the University of California, San Diego.

Mau-Tsuen Yang received his Ph.D. degree in computer science and engineering from the Pennsylvania State University, University Park, in 2000.

He is currently an Assistant Professor in the Department of Computer Science and Information Engineering of the National Dong-Hwa University, Taiwan. His research interests include computer vision, virtual reality, and real-time systems.

Rangachar Kasturi (S'82—M'82—SM'88—F'96) received a B.E. in electrical engineering in 1968 from Bangalore University, Bangalore, India, and an M.S.E.E. in 1980 and a Ph.D. in 1982 from Texas Tech University, Lubbock.

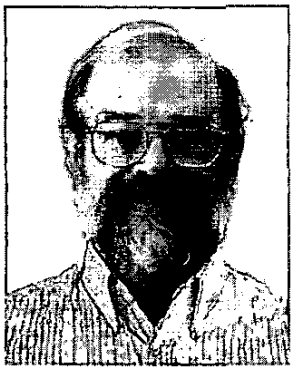
He is currently a Professor of Computer Science and Engineering at Pennsylvania State University, University Park.

Dr. Kasturi is the President of the International Association for Pattern Recognition (IAPR). He has served as Vice President for Publications of IEEE Computer Society (2001–2002) and as Editor-in-Chief of *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1995–1998). He is a coauthor of the textbook *Machine Vision*, McGraw-Hill,1995. He is a Fellow of IAPR.

Octavia I. Camps (S'89—M'91) received the B.S. degree in computer science and the B.S. in electrical engineering from the Universidad de la Republica, Uruguay, in 1981 and 1984, respectively, and the M.S. and Ph.D. degrees in electrical engineering from the University of Washington, Seattle, in 1987 and 1992, respectively.

In 1991 she joined the faculty of the Pennsylvania State University, University Park, where she is an Associate Professor in the Departments of Electrical Engineering and Computer Science and Engineering since 1999. Her current research interests include robust computer vision applied to visual tracking and object recognition, pattern recognition, and image-processing.

Dr. Camps is a member of the IEEE Computer, and Robotics and Automation Societies, and Tau Beta Pi. She was awarded an NSF Research Initiation Award in 1993 for her work on robust 3D object recognition.



Lee D. Coraor received a B.S. in electrical engineering from Pennsylvania State University, University Park, in 1974 and the Ph.D. in electrical engineering from the University of Iowa, Iowa City, in 1978.

He was a member of the faculty at The Southern Illinois University–Carbondale from August 1978 to August 1980 and then joined the Penn State faculty in the Department of Electrical Engineering. He is currently an Associate Professor in the Department of Computer Science and Engineering at Penn State. His current research interests include reconfigurable computing applications, intelligent memory designs, computer architecture, and digital systems. Recent projects have included: the use of reconfigurable FPGAs for implementing event-driven simulation, implementation of special purpose hardware/software on aircraft for real time collision avoidance, the design and implementation of a dual computer control system for a micro-gravity continuous flow electrophoresis system, and the development of the SmartDIMM Platform for use as a reconfigurable System-On-Chip prototype.

Jeff McCandless attended graduate school at the University of California at Berkeley, where he received a M.S. in mechanical engineering and a Ph.D. in vision science.

Since 1996, he has worked at NASA Ames Research Center in Moffett Field, CA. During his first three years at NASA, he developed image-processing algorithms for detecting aircraft in the sky. Since 1999, he has helped design updated displays for the cockpit of the Space Shuttle. His interests are human factors, human-computer interaction, and image-processing.