PATRICIA K. MURPHY and GENE A. HEYLER

IMAGE PROCESSING ABOARD THE MIDCOURSE SPACE EXPERIMENT USING THE ULTRAVIOLET AND VISIBLE IMAGERS AND SPECTROGRAPHIC IMAGERS INSTRUMENT

The Ultraviolet and Visible Imagers and Spectrographic Imagers instrument includes an onboard image processor that augments the closed-loop tracking ability of the Midcourse Space Experiment satellite. The image processor, in conjunction with the satellite's tracking processor, supports closed-loop tracking of various objects of interest, including dedicated launches and atmospheric phenomena such as Earth's aurora. Designed and developed by APL's Space Department, the image- and track-processor systems are being integrated into the Midcourse Space Experiment spacecraft. Experimental analyses using real imagery have been conducted to validate the instrument's tracking performance.

INTRODUCTION

The Midcourse Space Experiment (MSX) is a DoD-sponsored satellite scheduled for launch in 1994. Its expected operating lifetime is 5 years. To accomplish MSX's primary mission of observing target and atmospheric phenomena, a variety of instruments covering multiple wavebands are part of the payload. The Laboratory's Ultraviolet and Visible Imagers and Spectrographic Imagers (UVISI) instrument is one of the scientific devices aboard the MSX satellite. Also included are the Spirit III infrared imager and interferometer, the Space-Based Visible sensor, and APL's S-band Beacon Receiver. The UVISI comprises four imagers and five imaging spectrographs. The imagers include both narrow and wide field-of-view (FOV) visible sensors and narrow and wide FOV ultraviolet sensors. In addition to UVISI's primary function of data collection in the visible and ultraviolet wavebands, this instrument has an embedded image processor (IP) that supports autonomous closed-loop tracking.

The UVISI IP subsystem, in conjunction with the MSX tracking processor (TP) subsystem, can perform closed-loop tracking of various objects of interest. This UVISI tracking function supplements the ability to track dedicated targets with known flight profiles. For such dedicated targets, the beacon receiver will be the principal tracking sensor; UVISI will be used as a backup. The UVISI, however, will be the primary sensor for tracking targets of opportunity and atmospheric phenomena, such as Earth's aurora, that have either unknown or stochastic profiles.

In this article, we describe the distribution of processing roles and the resulting hardware architectures of the IP and TP subsystems along with the image processing algorithms that have been implemented. These algorithms detect and prioritize multiple candidate objects sensed in the UVISI imager's FOV. We discuss features of the TP algorithms specific to UVISI tracking, including track initialization and track file update and management. Finally, we summarize and assess performance on the basis of several real-image data sequences.

SYSTEM CONFIGURATION

The general philosophy of the MSX UVISI tracking and pointing system is that separate subsystems perform the single-frame or spatial processing (UVISI IP) and the frame-to-frame or temporal processing (MSX TP). These subsystems are relatively self-contained with only low bandwidth interface communications between them. Although this independence reduces tracking capability because not all data are available to any one subsystem, it greatly simplifies the development of the flight code software and results in a simpler and cheaper hardware design that uses current radiation-hardened Military Standard 1750A microprocessors.

Figure 1 is a block diagram of this tracking system. The wide-angle instruments have a $13.1 \times 10.5°$ FOV, and the narrow-angle instruments have a $1.6 \times 1.3°$ FOV. Any one of the four available imagers may be selected and used for tracking. The raw 12-bit 256×244 pixel image is forwarded to the UVISI IP at a 2-Hz rate, where spatial processing determines the centroids x_i, y_i of each candidate object in the scene. A figure of merit, or probability p_i, is associated with each candidate, which is related to the probability that the candidate is the desired object of interest. Candidates are ranked by their figures of merit. Up to 23 candidates with the highest probability of being the object of interest are reported to the MSX TP.

The MSX TP takes this list of reported candidates and performs both fine and coarse frame-to-frame data association and velocity filtering to determine which candidate is the true object of interest. Kalman filtering[1] or

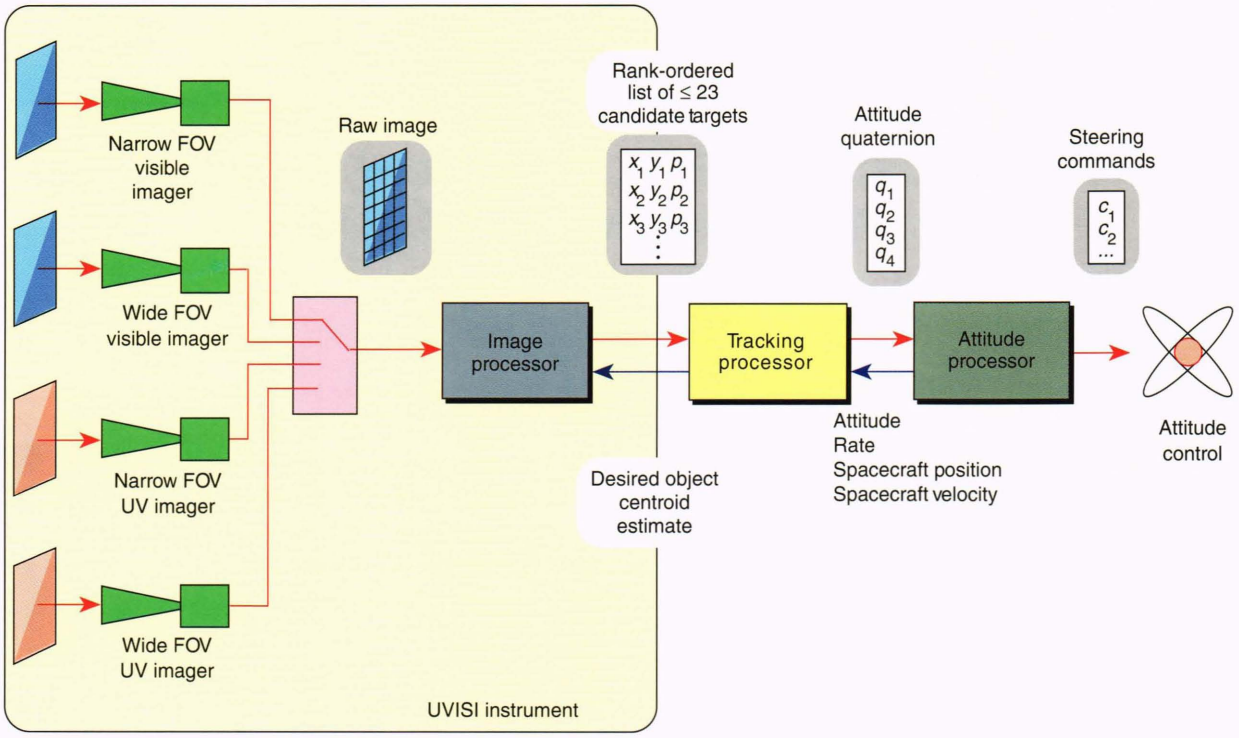


Figure 1. Tracking and pointing system configuration of the MSX's onboard UVISI instrument. (Reprinted, with permission, from Murphy, P. K., Heyler, G. A., and Waddell, R. L., "Image and Track Processing in Space (Part II)," in *Proc. AIAA Computing in Aerospace 9 Conf., Part I*, AIAA 93-4560-CP, San Diego, CA, p. 587 (1993); © 1993 by the American Institute of Aeronautics and Astronautics.)

simpler α–β filtering is then used to ensure continued tracking of this single object and generate the attitude quaternions required by the MSX attitude processor subsystem. The attitude processor also supplies current attitude, attitude rate, spacecraft position, and spacecraft velocity data to the TP. The centroid and type, that is, satellite, auroral feature, plume, and so forth, of the object being tracked are fed back to the UVISI IP to be used in determining the figures of merit of future detected candidates.

The UVISI IP is composed of a specially designed preprocessor board used for subimage selection, the Analog Devices digital signal processor ADSP-2100 radiation-hardened board for fast vector processing, and a 1750A microprocessor to perform nonvectorized processing. The MSX TP consists of one primary and one redundant 1750A microprocessor. Most of the flight software is coded in the Ada programming language; low-level routines are coded in assembly language. In addition, both subsystems can be reprogrammed during orbit.

IMAGE PROCESSING ALGORITHMS

The UVISI IP contains a uniquely flexible real-time onboard image processing algorithm suite. One of the four imagers is selected for input to the IP; onboard control logic and ground commands determine which imager is used as well as the criteria for switching from one imager to another. The image processing suite may be likened to a "toolbox" containing linear filters, nonlinear order statistic filters (e.g., median filter), detrenders, and an adaptive thresholder for performing available routines. These can be selected and sequenced to provide the overall processing algorithm via uplink command before a tracking event. This approach was adopted partly because the true nature of the image processing requirements cannot be entirely anticipated before launch; this flexibility allows us to modify the algorithm fairly easily as our knowledge of the data grows.

The baseline IP processing sequence is diagrammed in Fig. 2. Because the UVISI IP software is flexible in the types of objects it can track, the same algorithm, for example, can be run for both auroral and satellite tracking events simply by changing the information used to prioritize the candidates.

Although the image sent to the IP consists of 256 × 244 12-bit pixels, only a 64 × 64 pixel subimage can be processed at the required 2-Hz throughput rate. The entire 256 × 244 pixel image is compacted to generate a coarser-resolution image. Spatial averaging (with rounding) and decimation by a selectable factor of 1, 2, or 4 (1 = no compaction, 2 = averaging each 2 × 2 pixel neighborhood, 4 = averaging over each 4 × 4 pixel neighborhood) are performed in hardware on the custom-designed preprocessor board. The subimage is chosen by isolating a 64 × 64 pixel window within the compacted image centered about the predicted object location fed back from the TP. If no tracking processor information is available, the center of the window defaults to the center of the compacted image.

Processing each subimage entails several steps, including removing background and noise, segmenting candidate objects from the remaining background, and

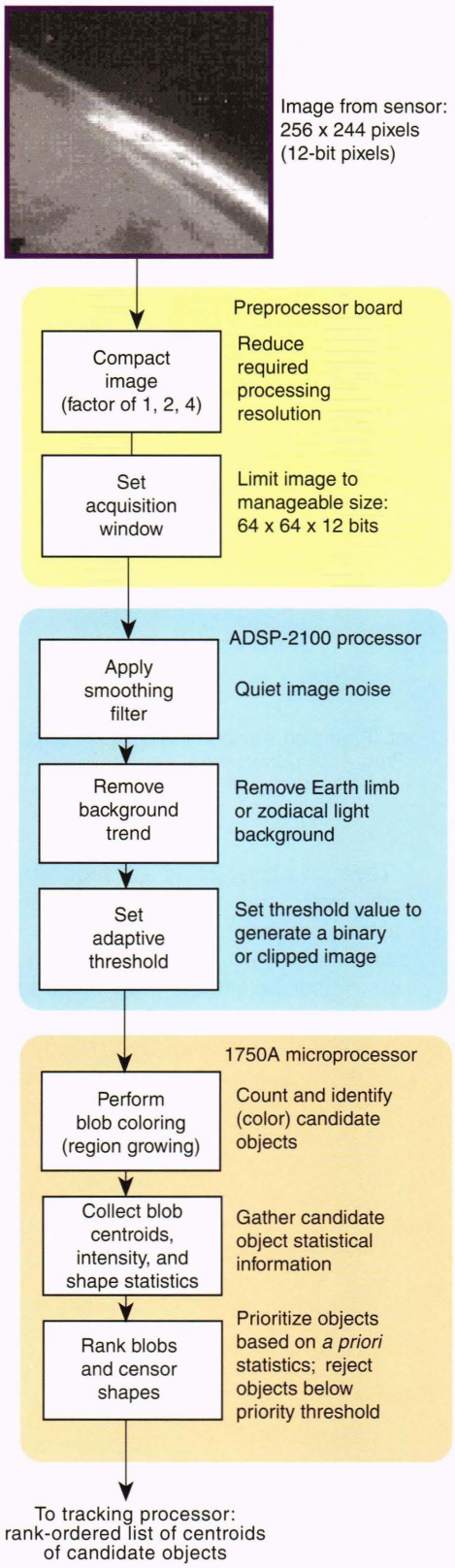


Figure 2. Image processing flow for detecting and prioritizing candidate objects. (Reprinted, with permission, from Murphy, P. K., Heyler, G. A., and Waddell, R. L., "Image and Track Processing in Space (Part II)," in *Proc. AIAA Computing in Aerospace 9 Conf., Part I*, AIAA 93-4560-CP, San Diego, CA, p. 588 (1993); © 1993 by the American Institute of Aeronautics and Astronautics.)

ranking candidates according to their computed object priorities. The following is a typical algorithm sequence:

1. Compacting a 256 × 244 12-bit pixel image by a factor of 1, 2, or 4 into a 64 × 64 12-bit pixel image
2. Linear filtering to attenuate noise (omitted when using a compaction factor of 2 or 4)
3. Image detrending to remove or reduce background trends
4. Linear filtering for performing matched filtering
5. Adaptive thresholding to separate candidate objects from the remaining background
6. Blob coloring to label individual candidates in the thresholded subimage
7. Extracting 12 features describing shape and intensity statistics for each labeled candidate
8. Candidate ranking using a minimum distance classifier with a supplied *a priori* 12-element feature vector

Figure 3 illustrates the application of these processing steps to a test image taken from the shuttle. The matched filter used was a simple 3 × 3 zero-mean circularly symmetric Gaussian kernel. Both the matched filter and object classification criteria were designed to locate a small, circular object in the image.

Linear Filtering

Linear convolution filters having sizes of 3 × 3, 5 × 5, or 7 × 7 pixels can be applied to the 64 × 64 pixel processing subimage. Filter sizes are limited to these smaller values, since the convolution is performed in the spatial domain. Convolution is frequently done before detrending for noise smoothing to ensure a better trend fit. It can also be used after detrending to help compensate for any artifacts introduced by the detrender. Filters can be uplinked to the satellite to replace the default filters available. Larger convolution kernels are applied via software frequency domain filtering, although throughput concerns limit the use of this option.

Detrending

The processed subimage may be passed through a detrending filter before threshold segmentation. Two types of detrenders are available: a high-pass filter detrender and a second-order polynomial surface-fit detrender. The purpose of either detrender is to remove the background trend of the subimage and thus accentuate the desired object. The adaptive threshold alone (discussed next), however, is enough to remove a constant background level from the data and is often all that is required. Detrending may also be omitted from the processing loop if its effects will not be helpful in locating the object of interest.

The high-pass filter detrender generates an estimate of the background by convolving the 64 × 64 pixel region with a low-pass 7 × 7 pixel normalized Gaussian filter kernel. This background is subtracted from the original subimage to yield the detrended subimage. The alternative detrending algorithm yields an estimate of the background by fitting a two-dimensional second-order polynomial surface to the data, which is then subtracted from the original. The type of tracking event will determine

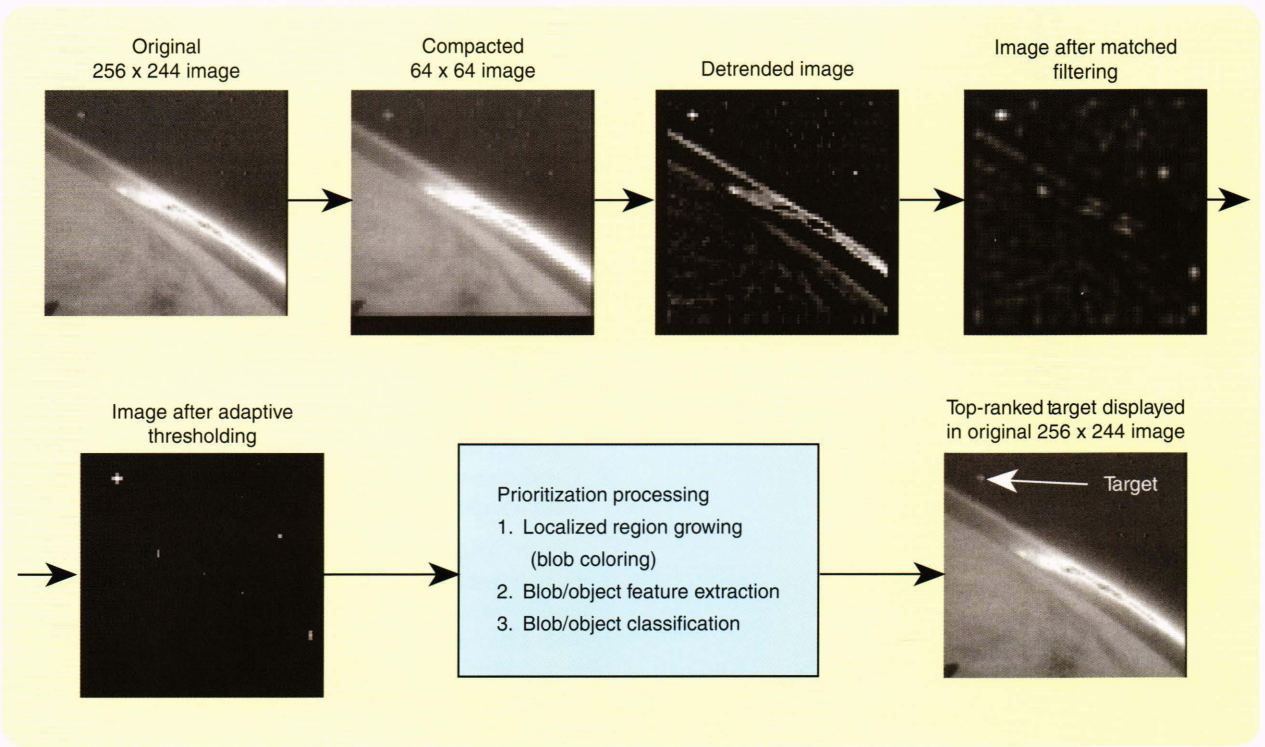


Figure 3. Identification processing of satellite or small extended object. This example shows the processed images at various stages in the UVISI image processing algorithm.

which detrending technique will be most beneficial to object tracking.

Thresholding

Candidate objects in the subimage are separated from the residual background using an adaptive threshold algorithm during segmentation. This algorithm computes a threshold value from the subimage data statistics. Pixels that pass the threshold are either set to a constant value (binarized) or left pristine (clipped), whereas those that do not are set to zero. The threshold is defined as

$$\text{Threshold} = \text{Mean} + \alpha\ \text{Standard Deviation},$$

where the multiplier α is a selectable input. The appropriate setting for α depends on the type of detrending implemented. The high-pass filter detrender can amplify noise as well as signal and therefore requires a larger value for α than is normally used for subimages detrended using the polynomial detrender. Onboard logic exists that increases α if the number of candidates detected becomes too large (approaches 100) or decreases α if the tracking processor does not lock on track with any of the available candidates (useful for dim objects).

Feature Extraction

Feature extraction entails identifying isolated regions or blobs in the thresholded or clipped input subimage, assigning all pixels in a blob a unique number or color associated with the blob, and finally collecting statistical data for each colored blob. The purpose of coloring the blobs is to ensure that they are easily distinguishable so that the statistical features associated with individual blobs can readily be calculated.

A localized region-growing technique called blob coloring[2] is used to identify and color each blob. The color assignments are additional identifiers; the algorithm does not change the actual pixel intensities. The subimage is processed a row at a time starting at its upper left corner. The blob-coloring algorithm works for either binary or clipped thresholded images; it treats nonzero pixel values as 1 = on and zero values as 0 = off.

To determine if the pixel under consideration is part of an already established blob, the 4 neighboring pixels, as shown in Fig. 4a, are examined. A 4-bit number n is created from the ordered binary values of these neighbors as follows:

$$n = (abcd) \text{ base } 2,$$

where

a = 1(0) if left neighbor is on(off),

b = 1(0) if upper left neighbor is on(off),

c = 1(0) if upper neighbor is on(off), and

d = 1(0) if upper right neighbor is on(off).

Special cases requiring additional logic include pixels in the first row, the first column, and the last column. A case statement[3] procedure is performed based on the value of the number. The pixel under consideration is assigned a color depending on the case number ($n = 0, \ldots, 15$). For example, when $n = 0$, not one of the

four neighbors of the current pixel is on, and we assign the pixel a new color. However, when $n = 4$, the upper left neighbor is on, and thus the current pixel is assigned the same color as this previously colored neighbor.

The feature extraction method just described includes diagonal connectivity between pixel regions when determining blob extent. Diagonal connectivity can be turned off by setting the appropriate uplink flag, whereby only pixels directly above or directly to the left are considered neighbors.

An additional aspect of this method is that circumstances can arise when already processed pixels must be recolored. Figure 4b illustrates one such situation. Since the subimage is processed line by line, the first row initially is assigned two colors, one for each apparent region. During the processing of the second row, the connection between the two regions is detected, and a single color is determined for the entire connected region. Instead of recoloring the pixels, an intermediate color array allows an elegant solution to this problem. The numeric values assigned are now indices into this color array, thus permitting multiple numeric values to be assigned to the same color value.

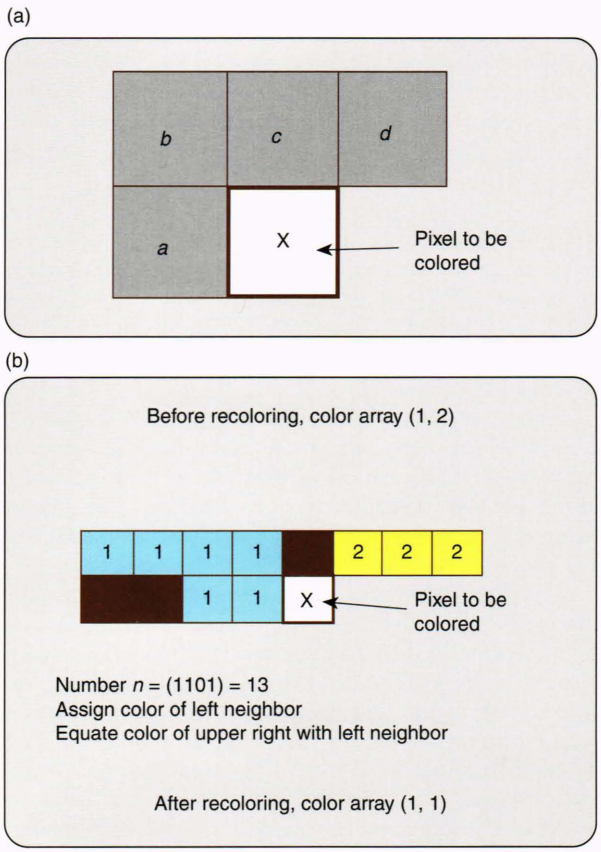


Figure 4. Blob-coloring approaches. (a) Blob-coloring 4-pixel neighborhood. (b) Blob recoloring. (Reprinted, with permission, from Murphy, P. K., Heyler, G. A., and Waddell, R. L., "Image and Track Processing in Space (Part II)," in *Proc. AIAA Computing in Aerospace 9 Conf., Part I*, AIAA 93-4560-CP, San Diego, CA, p. 590 (1993); © 1993 by the American Institute of Aeronautics and Astronautics.)

While the subimage is scanned for blob coloring, basic statistical information is collected for each detected blob, such as the number of pixels in the blob, the sum of pixel intensities, the sum of the weighted column indices, and the sum of weighted row indices. This preliminary information is further processed to produce 12 statistical features for each detected blob.

Blob shape and location statistics include size, intensity or geometric centroid, estimated semimajor and semiminor blob axes lengths, orientation, and eccentricity. An alternative estimate of eccentricity is calculated as the minor to major axis ratio (sigma ratio). Blob intensity statistics calculated are average brightness, maximum brightness, total brightness, and a brightness ratio (the ratio of average to maximum brightness). These 12 statistics constitute the feature vector computed for each detected candidate object (blob). The generated feature vectors are subsequently used to determine the likelihood that the candidate blob is the desired object. Most of the intensity statistics are self-explanatory with the possible exception of the brightness ratio, which gives a measure of how diffuse the intensities are within a blob. The shape statistics calculated for each blob are described in more detail in the next section. The boxed insert lists the equations used to generate each of the 12 statistics.

Shape Statistics

Shape statistics calculated include the size or number of pixels in the blob and the row and column centroids or center of mass in UVISI imager pixel coordinates. The centroids can be calculated as either intensity or geometric centroids via uplink control of an IP intensity flag. If the intensity flag is set true, the actual pixel intensity values within the blob region are used in the calculation of the centroids. Using the actual pixel values increases the accuracy of the centroid calculation. We estimate that blob centroids can be determined to within subpixel accuracies of approximately 1/10 of a processed pixel. Translating this accuracy to an angular measurement, however, depends not only on the imager FOV but on the compaction factor used to generate the processed subimage. These centroids are reported to the MSX TP, which has access to the appropriate boresight offsets and pointing data to transform them into angular measurements.

Additional shape statistics calculated are orientation and eccentricity. The orientation is a statistical moment that is invariant to changes in position or scale.[2] It is used to determine the angular orientation of the blob with respect to a reference coordinate system. The eccentricity is a position, scale, and orientation invariant moment that provides a measure of blob elongation.[2] These statistics are useful for differentiating the desired object from star streaks in the sensed image, which share a common orientation and have a higher eccentricity than the desired object.

The approximate lengths along the semimajor and semiminor axes of each blob are calculated by performing a rotational transformation by the orientation angle θ_n on the central moments of the blob. The semimajor and semiminor axis lengths for each blob are then approximated from a multiple of the square root of the

TWELVE SHAPE, LOCATION, AND INTENSITY STATISTICAL FEATURES CALCULATED FOR EACH OF THE NBLOB CANDIDATES DETECTED IN EACH UVISI SUBIMAGE

Statistical Features

$$\text{Size}_n \equiv \sum_{i=1}^{256}\sum_{j=1}^{244} \text{I}_n(i,j) \text{ for } n = 0, 1, 2, 3, \ldots \text{NBLOB} - 1$$

$$\text{Centroid column}_n \equiv \text{column}_n = \sum_{i=1}^{256}\sum_{j=1}^{244} (i + 1/2)\text{p}_n(i,j)$$

$$\text{Centroid row}_n \equiv \text{row}_n = \sum_{i=1}^{256}\sum_{j=1}^{244} (j + 1/2)\text{p}_n(i,j)$$

$$\text{Orientation } \theta_n \equiv \frac{1}{2}\tan^{-1}\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right)$$

$$\text{Eccentricity}_n \equiv \frac{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}{}^2}{(\text{size}_n)^2}$$

$$\begin{aligned}\text{Semimajor axis length}_n \approx \{12[&\mu_{20}\cos^2(\theta_n) \\ &+ 2\mu_{11}\cos(\theta_n)\sin(\theta_n) \\ &+ \mu_{02}\sin^2(\theta_n)] + 1\}^{1/2}\end{aligned}$$

$$\begin{aligned}\text{Semiminor axis length}_n \approx \{12[&\mu_{20}\sin^2(\theta_n) \\ &- 2\mu_{11}\cos(\theta_n)\sin(\theta_n) \\ &+ \mu_{02}\cos^2(\theta_n)] + 1\}^{1/2}\end{aligned}$$

$$\text{Sigma ratio}_n \equiv \frac{\text{Semiminor axis length}_n}{\text{Semimajor axis length}_n}$$

$$\text{Total brightness}_n \equiv \sum_{i=1}^{256}\sum_{j=1}^{244} \text{Intensity}(i,j)\text{I}_n(i,j)$$

$$\text{Average brightness}_n \equiv \frac{\text{Total brightness}_n}{\text{Size}_n}$$

$$\text{Maximum brightness}_n \equiv \max[\text{Intensity}(i,j)\text{I}_n(i,j); \forall(i,j)]$$

$$\text{Brightness ratio}_n \equiv \frac{\text{Average brightness}_n}{\text{Maximum brightness}_n}$$

Definitions

NBLOB = number of blobs detected

$$\text{Indicator } \text{I}_n(i,j) \equiv 1; \quad \text{for } (i,j) \in \text{blob}_n$$
$$\equiv 0; \quad \text{else}$$

$$\text{Normalized intensity } \text{p}_n(i,j) \equiv \frac{\text{Intensity}(i,j)\text{I}_n(i,j)}{\sum_{k=1}^{256}\sum_{m=1}^{244}\text{Intensity}(k,m)\text{I}_n(k,m)}$$

$$\text{Central moment } \mu_{pq} \equiv \sum_{i=1}^{256}\sum_{j=1}^{244}(i + 1/2 - \text{column}_n)^p (j + 1/2 - \text{row}_n)^q \text{p}_n(i,j),$$

where p and q are any two integers

transformed variances. An alternative measure of blob elongation, the sigma ratio, is also calculated as the ratio of the minor to the major axes.

Normalizing Statistics

After all of the statistics are calculated for a blob, one important action remains. To remove the scale-dependent variations of some of the statistics for different compaction factors, processing windows, and imaging FOVs, these statistics are normalized to the reference scale of the visible wide FOV imager with a compaction factor of 1 (no compaction). This process requires a final normalization of the size, the semimajor and semiminor axis lengths, and the total brightness. The compaction factor and processing window parameters are also used to project the blob centroids into the original coordinate system of the imager that generated the data.

Candidate Ranking

After blob coloring is implemented to isolate candidate objects and the statistical features are extracted for each candidate, a minimum distance classifier is used to rank the candidates. Prioritization involves comparing the shape and intensity statistics of each candidate with the *a priori* object record. The various objects of interest (e.g., star, satellite, or auroral feature) each have an *a priori* feature record in the UVISI IP. Depending on the mission event (e.g., auroral tracking), the appropriate *a priori* record is chosen and used to prioritize candidates. New *a priori* records can be uploaded after launch as our knowledge of the shape and intensity characteristics of objects increases.

The *a priori* feature record required as input to the IP specifies the expected mean value, standard deviation, and a weighting factor for each of the shape and intensity feature statistics calculated. Two additional parameters, a lower and upper quantization threshold value, are also required for each statistic and are used during the coarse quantization operation.

Table 1 presents the auroral feature record that was used in the testing sequences discussed later in the Summary section. This table was determined from a training subset of the available auroral imagery. The feature means m_i and standard deviations σ_i were calculated from the training data set.

Although the IP decision and control logic can dynamically change which *a priori* feature record is used from information supplied by the TP, the values in each *a priori* record are static except for the mean column and row centroids. These values are initially set to the center of the processing FOV but are updated each frame to the predicted object location supplied in the TP message. This process helps ensure that the same object is given highest priority from frame to frame and avoids alternating between competing candidate objects. A more robust algorithm would update all of the statistical features from frame to frame; however, the data association and filtering required to update all features dynamically are beyond available UVISI IP computing resources.

Table 1. *A priori* target record for auroral features.

Parameter	Mean	Standard deviation	Weight	Lower–upper quantization thresholds
Size	600.0	100.0	1.0	100.0–1000.0
Centroid column	127.0	5.0	1.5	85.0–175.0
Centroid row	127.0	5.0	1.5	85.0–175.0
Orientation	0.0	10.0	0.0	0.0–180.0
Eccentricity	5.0	1.0	1.0	3.0–8.0
Semimajor axis	50.0	10.0	1.0	10.0–40.0
Semiminor axis	10.0	5.0	1.0	10.0–40.0
Sigma ratio	0.2	0.1	1.0	0.25–0.75
Total brightness[a]	1000.0	1000.0	0.0	1000.0–10,000.0
Average brightness[a]	230.0	5.0	1.0	200.0–255.0
Maximum brightness[a]	255.0	5.0	1.0	240.0–255.0
Brightness ratio	0.85	0.1	0.5	0.25–0.75

[a]Eight-bit data assumed for this example.

The weighting factor w_i associated with each feature reflects the importance of that feature in distinguishing the desired object from competing objects. This allows us to tailor the *a priori* feature record to emphasize certain statistics and ignore others. For example, during satellite tracking, orientation may be a useful feature, whereas, during auroral tracking, orientation is not useful owing to the stochastic nature of the aurora. Because it is important to choose a consistent object to follow, centroids are highly weighted as compared with other statistical features.

In generating the probability that a candidate is the desired object, the IP sums the weighted squared and normalized differences between the candidate and *a priori* features ($n = 12$) to produce a distance metric as follows:

$$d = \sum_{i=0}^{n-1} \frac{w_i (x_i - m_i)^2}{\sigma_i^2}.$$

In essence, the weighting factors modify the variances. If we make the assumption that each candidate feature x_i is an independent Gaussian random variable, this distance becomes a chi-squared random variable with n degrees of freedom (DOF). Although these features are not really independent, the common assumption of independence greatly simplifies the resulting analysis and implementation. A goodness-of-fit (GOF) probability is then calculated as the probability that the distance (random variable) is greater than or equal to the observed distance value d,

$$\mathrm{GOF}(d) = \int_d^\infty f_d(\gamma) d\gamma = 1 - F_d(d),$$

where $f_d(d)$ and $F_d(d)$ are the probability density function and cumulative distribution function, respectively, of a chi-squared n DOF random variable.[4]

This GOF probability is a smoothly decreasing function that starts at a value of 1.0 when $d = 0$ (d is always nonnegative) and decays to zero as d approaches infinity. Unfortunately, no closed-form expression for $F_d(d)$ exists. Our primary interest in the GOF probability is for use in ranking the candidate objects. However, the GOF values are also used to reject candidates with probabilities less than an uploadable probability threshold. An approximate GOF function that is easier to evaluate is

$$\mathrm{GOF}(d) = \exp[-d^2 \ln(2) / n^2].$$

Figure 5 is a graph of the exact chi-squared GOF and the exponential approximation for $n = 10$ DOF. Using the approximation does not affect the performance of the IP

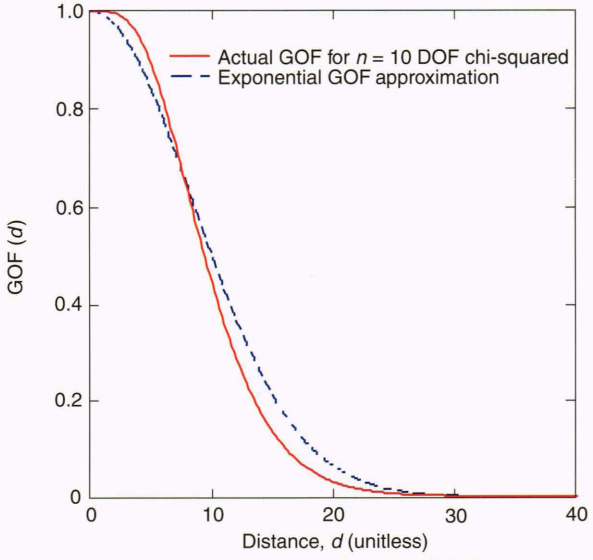


Figure 5. Actual $n = 10$ degree-of-freedom (DOF) chi-squared goodness-of-fit (GOF) probability versus distance d and the computationally simpler exponential GOF approximation used in the flight software.

algorithm and saves on computation for the flight software implementation. The exponential approximating function has the property that when d takes on its mean value ($d = n$), the GOF probability becomes 1/2. The number of DOF, n, is not always equal to 12 but is instead determined by the number of nonzero weights w_i.

An additional feature that has been included to provide extra robustness to the probability computation is the coarse quantization option. When the quantization flag is set to true, both the *a priori* and the candidate statistical features are compressed from their original values to values between 0 and 1 by means of the upper and lower quantization thresholds provided in the *a priori* target record. The compressed values are then used in calculating the distance metric. The compression function was originally simply a multilevel staircase function (with small = 0, medium = 1/2, and large = 1 categories) but has been modified to be a sigmoidal function that transitions smoothly from 0 to 1.0. This sigmoidal nonlinearity is useful for categorizing all observations greater than some upper threshold as equivalently large or, likewise, all observations less than some lower threshold as equivalently small. The sigmoidal function used to transform the *a priori* mean and observed candidate feature is

$$\hat{m}_i = 1.0 - \exp(-\alpha^2 m_i^2) \quad \text{and} \quad \hat{x}_i = 1.0 - \exp(-\alpha^2 x_i^2),$$

where

$$\alpha \equiv \frac{[-\ln(1/2)]^{1/2}}{[(\text{upper} + \text{lower})/2]}.$$

The value of α is chosen so that $\hat{m}_i = 1/2$ when $m_i =$ (upper + lower)/2. This function is roughly linear near this midpoint but is compressed for values outside the upper and lower thresholds. The variance is set to that of a random variable that is uniformly distributed on the unit interval, that is, $\hat{\sigma}_i^2 = 1/12$. Note that this transformation is fashioned after the classical $\hat{x} = F(x)$ transformation that makes $\hat{x}$ a uniform [0, 1] random variable.[4]

TRACK PROCESSING ALGORITHMS

For dedicated launches, the TP contains the complete dynamic model of the tracking event, including spacecraft attitude and attitude rate, spacecraft position and velocity, and object *a priori* position and velocity; thus, it is the TP rather than the UVISI IP that can distinguish the object from the candidate list on the basis of velocity.

Up to 23 observations are received in each data frame from the UVISI. Although the UVISI ranks the observations according to the figures of merit, the rankings are determined only from individual frame spatial processing. That is, there is no frame-to-frame correlation over time. Thus, especially during observations of objects appearing as point sources against space backgrounds, stars can easily be identified by the UVISI IP as high-probability candidates and passed on to the TP. The velocity filtering algorithm in the TP uses the known event dynamics to observe trends in the UVISI data, thereby distinguishing objects that move in inertial space from stars that do not.

Each observation is converted from UVISI imager pixel centroids to two angles from boresight in the spacecraft body coordinates. A binary hypothesis approach based on apparent motion of the observations in this coordinate system identifies the candidates as stars or moving objects. Both star and object hypotheses are propagated forward for each initial observation. Hypothesis testing continues until the error accumulations result in firm decisions in favor of either the star or moving object. Verified object observations are then passed on to a 6-state extended Kalman filter.

For targets of opportunity and special tracking events such as auroral tracking, no *a priori* object trajectory information is available. Under these circumstances, the MSX TP performs no velocity filtering. The highest-ranked candidate observation that falls within the angular tracking gate window is input to a simpler α–β tracking filter.

Track Initialization

During initialization, two tracking gates are centered on each observation: one object gate and one star gate. These gates are then propagated to the next frame in accordance with velocity assumptions for the respective object and star hypotheses. Star gates are propagated on the basis of spacecraft attitude rate only, whereas object gates must be propagated considering both the *a priori* object motion and attitude rate. Each hypothesis gate is then smoothed with the new observation(s) inside the gate using an α–β tracker, and the squared differences or residuals between the observation and the gate center are accumulated. The position gain α for the α–β tracker is selected empirically to generate a desired response time; $\alpha = 0.25$ provides a 90% response in eight frames (4 s). The velocity gain β is almost always set to 0, since it is assumed velocities of stars and objects are known *a priori* from the supplied nominal trajectory. In fact, it must be 0 or both types of gates will follow the data and differentiation based on velocity cannot occur. The only exceptions are events for which no nominal trajectory is supplied; then nonzero values for β are necessary.

Hypothesis Rejection

Gate hypotheses can be rejected in two ways, either by the gate being empty of observations for several successive frames or by losing an error residual test to the alternate hypothesis. When a gate (either object or star) is smoothed with its associated observation, the squared difference of the observation and the gate center is accumulated into an error residual variable assigned to that gate. The difference of the two accumulated residuals (object versus star) can be thresholded to determine the identity of the observation.

Since there are two types of gates and two hypotheses for each gate, four rules control the velocity filtering algorithm as follows:

1. If a star gate is empty for n successive frames but the object gate remains populated, conclude that the observation is a moving object and delete the associated star gate.

2. If an object gate is empty for n successive frames but the star gate remains populated, conclude that the observation is a star and delete both the star and object gates.
3. If the accumulated residuals of a star gate exceed those of the object gate by a specified amount, conclude that the observation is a moving object and delete the associated star gate.
4. If the accumulated residuals of an object gate exceed those of the star gate by a specified amount, conclude that the observation is a star and delete both the star and the object gates.

If, and only if, all gates have been deleted and no valid object is found, reinitialization occurs, with a set of new gates placed on the current UVISI IP imager frame's observations. Because onboard processing capacity is limited, and since the set of tracking event types is constrained in such a way that it is highly probable the moving object is within the FOV, we have not considered it necessary to add additional gates for new observations entering the FOV that cannot be associated with any established star or object gates. For these reasons, and since the number of tracks is expected to be low, no track pruning or track splitting is implemented.

Trajectory Estimation

The location of the single valid target object identified by velocity filtering is forwarded to the recursive filter for refining the object state estimate. For satellite tracking, the filter equations incorporate a plant model with a 6-parameter state vector (inertial position and velocity) integrated with a fourth-order Runge–Kutta method.[5] The filter plant model can also use an uplinked set of

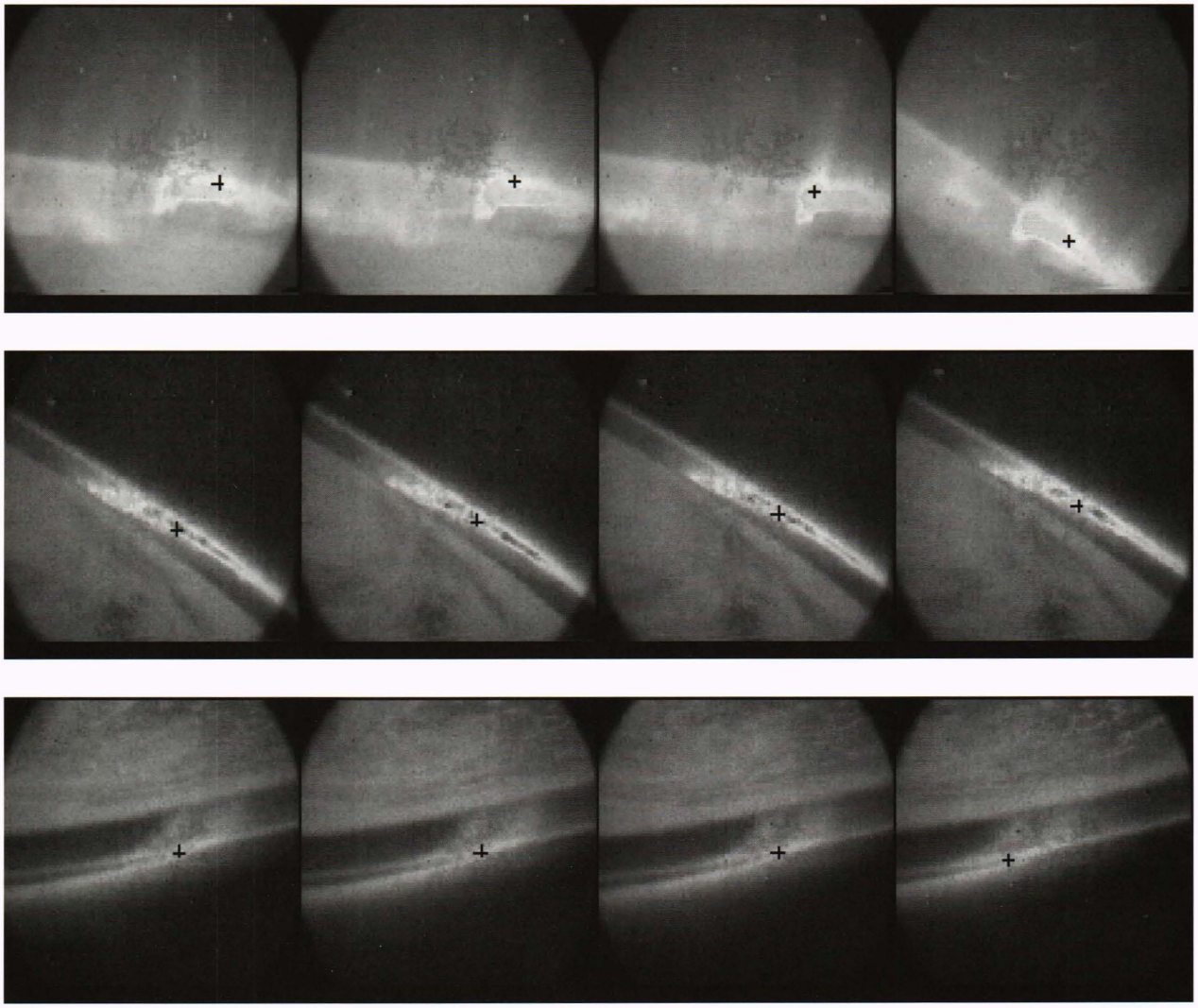


Figure 6. Auroral tracking by the UVISI IP. Three separate sequences of auroral imagery taken from NASA's shuttle-based 51-B Auroral Image Experiment were used as input for testing. Even though the sequences contain distinctly different auroral features, the same *a priori* auroral feature record was used for each. The centroid of the highest-ranked auroral feature in each image is identified by a plus sign. (Reprinted, with permission, from Murphy, P. K., Heyler, G. A., and Waddell, R. L., "Image and Track Processing in Space (Part II)," in *Proc. AIAA Computing in Aerospace 9 Conf., Part I*, AIAA 93-4560-CP, San Diego, CA, p. 596 (1993); © 1993 by the American Institute of Aeronautics and Astronautics.)

curve-fit coefficients that describe position, velocity, and acceleration of the object. The 6-parameter filtered state vector again represents position and velocity, whereas acceleration is derived from the curve fit for the filter propagation equations.

Blob centroid data from the UVISI IP are transformed into a unit vector in the imager coordinate system. Because the angles measure deviation from the sensor boresight, the measurement unit vectors will have the form $(\approx 1, \epsilon_1, \epsilon_2)$, where the first element is redundant information. By assuming the second and third components of the unit vector to be independent, a technique called sequential measurement updating is employed to incorporate the data. The second and third components are incorporated individually in the filter as scalar measurements rather than jointly with the first component as a measurement vector. The filter is operated twice, once for each component. This technique avoids computing a matrix inverse in the gain matrix equation as well as needless state updating based on the redundant (first) component of the measurement vector.

SUMMARY

The UVISI IP and MSX TP perform the critical functions of identifying and tracking candidate objects. The algorithms have been tested on both real and synthetic image data. In-house simulations have been developed to provide synthetic UVISI imagery of dedicated launch targets and targets of opportunity. The UVISI IP algorithms have been tested extensively on these synthetic data. For this article, however, we present some results for auroral feature tracking using real image data.

We obtained several sequences of image data containing real auroral features by digitizing VHS videotapes of auroral imagery taken from NASA's 51-B Auroral Image Experiment aboard the space shuttle. Although the imaging resolution, FOV, bandwidths, and viewing geometry were not necessarily the same as those of the MSX UVISI instrument suite, these real-world examples of auroral imagery provided a good opportunity to demonstrate the robustness of the UVISI IP algorithms. The waveband and FOV in this imagery approximate those of the wide FOV visible UVISI imager. The imagery is also taken from a space platform (rather than a ground-based site) and has an appropriately fast temporal sampling rate. The images exhibit considerable blooming and severe frame-to-frame motions due to the original handheld camera control. We show that the algorithm can consistently identify auroral features in the imagery by a judicious choice of *a priori* feature record.

The VHS tapes sampled exhibited considerable noise and were at least second-generation duplicates. The images were digitized by a commercial image processing board that limited the dynamic range to 8 bits. Spatial averaging was performed to turn the digitized video into the appropriate 256×244 pixel wide FOV images. Three sequences containing distinctly different auroral features were used in testing the software. Active aurorae exhibit localized features that appear bright in both the ultraviolet and visible wavebands. These bright features, or so-called auroral "surges" or "hot spots," are of primary interest for tracking.

The implemented UVISI IP algorithm did not detrend the auroral data and used the *a priori* feature record listed in Table 1 for all image sequences. Simple α–β tracking was simulated with the UVISI IP top-ranked candidate as input. Figure 6 shows sequences taken from the three auroral data sets. The plus sign identifies the centroid of the UVISI IP top-ranked candidate surge that was fed to the tracking filter.

The UVISI IP was able to identify the auroral surges as the top-ranked candidate in 98% of the test images. Proper setting of the *a priori* target record, however, is important in obtaining consistent results. Note that the UVISI IP does much more than simply find the brightest object in the sensed FOV; size, centroid, eccentricity, semimajor and semiminor axes, sigma ratio, total brightness, average brightness, maximum brightness, and the brightness ratio are all features used to rank the candidate objects. Ranking via the *a priori* target record assures successful object identification for other tracking event scenarios in which the object of interest is not necessarily the brightest object in the FOV.

REFERENCES

[1] Kalman, R. E., and Bucy, R. S., "New Results in Linear Filtering and Prediction Theory," *Trans. ASME*, 95–108 (Mar 1961).

[2] Ballard, D. H., and Brown, C. M., *Computer Vision*, Prentice-Hall, Englewood Cliffs, NJ (1982).

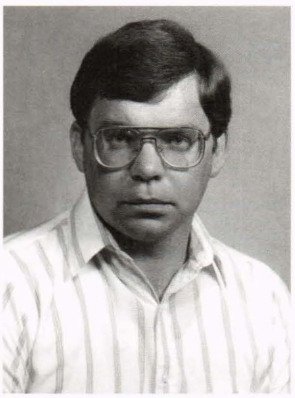
[3] Watt, D. A., Wichmann, B. A., and Findlay, W., *ADA Language and Methodology*, Prentice-Hall Internat. Series in Comp. Sci., Englewood Cliffs, NJ (1987).

[4] Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, 3rd Ed., McGraw-Hill, Inc., NY (1991).

[5] Gelb, A., *Applied Optimal Estimation*, The MIT Press, Cambridge, MA (1974).

THE AUTHORS

PATRICIA K. MURPHY received a B.S. and a Ph.D. in electrical engineering from Purdue University in 1979 and 1984, respectively. From 1984 to 1986, she worked in the areas of antijam communications and signal processing at the Government Aerospace Systems Division of the Harris Corporation in Melbourne, Florida. Dr. Murphy joined APL in 1986 and is currently a Senior Staff member of the Computer Science and Technology Group in the Space Sciences Branch. She has been involved in designing, developing, and testing the MSX's UVISI onboard image processor. Her present interests include image and signal processing, data compression, image restoration, and scene characterization and modeling.



GENE A. HEYLER received a B.S. in engineering science from Pennsylvania State University in 1978 and an M.S. in computer science from The Johns Hopkins University in 1985. From 1978 through the present, he has been a member of the Computer Science and Technology Group in the Space Sciences Branch. His areas of expertise include orbital analysis, such as orbit adjustments and lifetime predictions, integrators, and perturbations, as well as spaceborne closed-loop tracking and pointing techniques incorporating sensor fusion and Kalman filtering.