JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 17, No. 5, September–October 1994

# Vision-Based Obstacle Detection and Grouping for Helicopter Guidance

Banavar Sridhar and Gano B. Chatterji
*NASA Ames Research Center, Moffett Field, California 94035-1000*

**Electro-optical sensors can be used to compute range to objects in the flight path of a helicopter. The computation is based on the optical flow/motion at different points in the image. The motion algorithms provide a sparse set of ranges to discrete features in the image sequence as a function of azimuth and elevation. For obstacle avoidance guidance and display purposes, these discrete sets of ranges, varying from a few hundred to several thousand, need to be grouped into sets that correspond to objects in the real world. This paper presents a new method for object segmentation based on clustering the sparse range information provided by motion algorithms together with the spatial relation provided by the static image. Object grouping allows interpolation within a group and enables the creation of dense range maps. The paper presents object segmentation results for a sequence of flight images.**

## I. Introduction

THE design of intelligent guidance systems for helicopters will require information about objects in the vicinity of the flight path of the vehicle. The sensor system on the helicopter should be able to detect objects such as buildings, trees, poles and, wires during flight. A complete obstacle detection system may consist of an active ranging sensor and passive ranging using electro-optical sensors. Object location information over the field of view (FOV) is referred to as range map. Several techniques have been proposed for determining range using electro-optical cameras.[1–6] These techniques use optical flow resulting from the relative motion between the camera and objects on the ground together with the helicopter states from an inertial navigation system to compute range to various objects in the scene.

One of the difficulties in the range computation occurs due to the assumption that objects on the ground correspond to unique points in the image. In reality an object on the ground is more likely to occupy a region in the image. Complications are also caused due to the ambiguity in matching features in the images that are projections of the same object on the ground. These difficulties result in range maps where range is available only in certain parts of the image.

The sparse set resulting from the range computations does not define an explicit relationship between its members. For obstacle avoidance guidance and display purposes, these discrete sets of ranges need to be grouped into sets that correspond to objects in the real world. The reduction of discrete ranges, varying from a few hundred to several thousand, to a small number of objects improves the display and simplifies the computation of obstacle avoidance algorithms. The goal in this paper is to achieve object segmentation in sparse range images by using range relations in the three-dimensional world and interpixel relations in the image. The grouping or segmentation of a range map enables one to assume the range to be continuous within a group. It is possible to create a dense range map via interpolation within the groups. Modeling of dense range images has been studied by several authors.[7–9] The dense range images can be modeled into objects by fitting surfaces using polynomials, splines, Delaunay triangles, etc. The grouping of sparse range measurements makes it possible to extend the modeling techniques developed for dense range images to sparse range maps.

This paper is organized as follows: Section II provides a brief introduction to the range estimation algorithm. The clustering method-

**Banavar Sridhar received his B.E. degree in electrical engineering from the Indian Institute of Science and his M.S. and Ph.D. in electrical engineering from the University of Connecticut. From 1973 to 1976, he held a National Research Council Fellowship at the NASA Ames Research Center. He worked at Systems Control, Inc., Palo Alto, CA, and Lockheed Palo Alto Research Center before joining NASA Ames Research Center in 1986. At NASA he is leading a project involved with the development of automation tools for rotorcraft and other vehicles. Currently, he is an editor of *IFAC Journal Control Engineering Practice* and an associate editor of the *IEEE Controls Systems Magazine*. During 1991, he received the Engineer of the Year award from the Bay Area Section of the American Institute of Aeronautics and Astronautics.**

**Gano B. Chatterji received his Bachelor of Technology degree in mechanical engineering from the Indian Institute of Technology, Kanpur, India. He received his M.S. in engineering science from the University of Mississippi. From 1985 to 1990, he was employed by Integrated Systems, Inc (ISI). There he was involved in developing estimation, guidance, control, and trajectory optimization algorithms for exo/endo-atmospheric missiles. Since 1990, he has been at the FSN branch of NASA Ames Research Center as an employee of Sterling Software. At Ames he has worked on problems related to rotorcraft guidance using passive sensors and computer vision techniques. He has also developed techniques for clustering discrete ranges, obtained by the passive ranging schemes, to represent objects in the field of view.**

Received May 18, 1993; revision received Jan. 11, 1994; accepted for publication Jan. 12, 1994.

ology is described in Sec. III. Section IV describes the two helicopter flight image sequences used. Section V describes a technique for classifying features based on separation in depth. A $K$-means algorithm for grouping based on separation in the horizontal plane is presented in Sec. VI. Next, an algorithm based on the minimum spanning tree (MST)[10] that groups features based on the separation in the image plane is described in Sec. VII. Finally, Sec. VIII provides the results for the flight sequences. Ideas on future research and some concluding remarks are presented in Sec. IX.

## II. Range Estimation Algorithm

Consider a helicopter-mounted camera that observes a point $P$ on a stationary object in the environment, as shown in Fig. 1. The image point $\{u, v\}$ corresponding to the point $P$ is given by the perspective projection equations as follows:

$$u = f x_s / z_s, \qquad v = f y_s / z_s \tag{1}$$

where $x_s$, $y_s$, and $z_s$ are components of $\rho_s$, the object's position relative to the camera, and $f$ is the focal length of the camera lens. As the camera moves, the image of $P$ will also move. If $P$ is assumed fixed in the Earth frame, the rate of change of $\rho_s$ in the camera's axes system can be determined using the Coriolis equation as follows:

$$\dot{\rho}_s = -V_s - \omega_s \times \rho_s \tag{2}$$

where $V_s = \{V_{xs}, V_{ys}, V_{zs}\}$ and $\omega_s = \{\omega_{xs}, \omega_{ys}, \omega_{zs}\}$ are the camera's translational and rotational velocities with respect to the Earth frame. Differentiating the perspective projection equations with respect to time and substituting for $\dot{\rho}_s$ according to the above equation yields the well-known optic-flow equations that relate camera motion, object motion in the image, and the object's range. The optic-flow equations are

$$\begin{aligned}\dot{u} &= (-f V_{xs} + u V_{zs})/z_s + \left(\frac{uv}{f}\right)\omega_{xs} \\ &\quad - f\left(1 + u^2/f^2\right)\omega_{ys} + v\omega_{zs} \\ \dot{v} &= (-f V_{ys} + v V_{zs})/z_s + f\left(1 + \frac{v^2}{f^2}\right)\omega_{xs} \\ &\quad - \frac{uv}{f}\omega_{ys} - u\omega_{zs}\end{aligned} \tag{3}$$

Decomposing the optic flow into components due to camera translation $(\dot{u}_t, \dot{v}_t)$ and rotation $(\dot{u}_r, \dot{v}_r)$ gives

$$\dot{u} = \dot{u}_t + \dot{u}_r, \qquad \dot{v} = \dot{v}_t + \dot{v}_r \tag{4}$$

where

$$\begin{aligned}\dot{u}_t &= (-f V_{xs} + u V_{zs})/z_s \\ \dot{v}_t &= (-f V_{ys} + v V_{zs})/z_s \\ \dot{u}_r &= \frac{uv}{f}\omega_{xs} - f\left(1 + \frac{u^2}{f^2}\right)\omega_{ys} + v\omega_{zs} \\ \dot{v}_r &= f\left(1 + \frac{v^2}{f^2}\right)\omega_{xs} - \frac{uv}{f}\omega_{ys} - u\omega_{zs}\end{aligned} \tag{5}$$

With the optic-flow equations, range to an object $z_s$ can be determined given measurements of the camera's motion state $(V_s, \omega_s)$, the object's location in the image $(u, v)$, and the optic flow $(\dot{u}, \dot{v})$. Note that only the optic flow due to translation $(\dot{u}_t, \dot{v}_t)$ is a function of the object's range; therefore, range can be estimated only when the camera is translating. Furthermore, the optic flow due to translation will be zero at an image location $(f V_{xs}/V_{zs}, f V_{ys}/V_{zs})$ known as the focus of expansion (FOE). Since the FOE corresponds to the intersection of the velocity vector with the image plane, range estimates cannot be obtained for objects along the camera's instantaneous direction of motion. Additionally, the ability to estimate range deteriorates for objects whose translational optic flow is small, such as objects that are far away or that appear near the FOE.

Optic-flow information and ultimately a range estimate is associated with features in an image, where a feature is defined as a small region of interest within the image. The optic-flow measurements are obtained from the difference between an object's location in successive images. The number of features for which range estimates can be obtained depends directly on the ability to select robust features that can be unambiguously tracked between images. The quality of range estimate for each feature depends on the ability to accurately track the feature between images. In this implementation features are squares of 11 × 11 pixels that exhibit intensity variance greater than some user-specified minimum threshold value. A correlation method is used to determine the feature's location in each new image.

Using the optic-flow equations (3), we formulate the computation of range as a state estimation problem using a Kalman filter. The Kalman filter is well suited to this application because it combines redundant measurements to recursively improve its estimate over time. In addition, the state covariance matrix provided by the Kalman filter gives an indication of the estimate accuracy. The Kalman filter also yields a prediction of the state vector, the state covariance matrix, and an expected location of the feature for the next sample time. This latter information is used to constrain the search area for locating the feature in the next image. As the range information improves, the search window becomes smaller and less computation is required to locate the feature.

Several Kalman filter implementations were studied in Ref. 1, where the best results were obtained by selecting the state vector

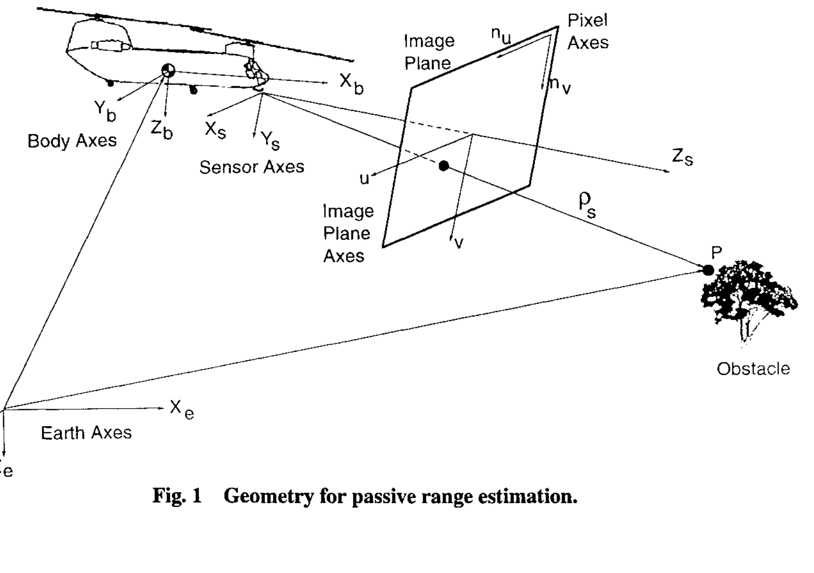

**Fig. 1 Geometry for passive range estimation.**

$X = \rho_s$ and the measurement vector $Z = \{u, v\}$. With these definitions, the Coriolis equation (2) becomes the state equation and the perspective projection equations (1) become the measurement equations. The state and measurement equations can be written as

$$\begin{aligned}\dot{X} &= -[\omega_s]X - V_s \\ Z = h(X) &= [fx_s/z_s,\ fy_s/z_s]^T\end{aligned} \tag{6}$$

where

$$[\omega_s] = \begin{bmatrix} 0 & -\omega_{zs} & \omega_{ys} \\ \omega_{zs} & 0 & -\omega_{xs} \\ -\omega_{ys} & \omega_{xs} & 0 \end{bmatrix} \tag{7}$$

The state equation is a time-varying linear system that depends on the camera's translational and rotational velocities. The measurement equation is a nonlinear function of the state.

The continuous-time state and measurement equations can be converted to their discrete-time equivalents assuming that $V_s$ and $\omega_s$ are constant during the sampling interval $\Delta T$. The discrete-time system equations are

$$X(k+1) = \Phi(k)X(k) + \Gamma(k)U(k) + \Gamma_d(k)\zeta_x(k) \tag{8}$$

$$Z(k) = h[X(k)] + \zeta_z(k) \tag{9}$$

where $\Phi(k)$ is the state transition matrix, $\Gamma(k)$ is the input distribution matrix, $U(k) = -V_s(k)$ is the control matrix, $\Gamma_d(k)$ is the disturbance distribution matrix, and $\zeta_x(k)$ and $\zeta_z(k)$ model the process noise and measurement noise, respectively. Zero-mean Gaussian white noise is assumed such that $R(k) \equiv cov(\zeta_z)$ and $Q(k) \equiv cov(\zeta_x)$. The state transition matrix and the control distribution matrices have been derived in Ref. 1. The measurement equation is linearized about the current estimate of $X$, giving

$$Z(k) = H(k)X(k) + \zeta_z(k) \tag{10}$$

$$\begin{aligned} H(k) &= \frac{\partial h(X)}{\partial X} \\ &= f\begin{bmatrix} 1/z_s & 0 & -x_s/z_s^2 \\ 0 & 1/z_s & -y_s/z_s^2 \end{bmatrix}_{\rho_s = \hat{\rho}_s(k-1)} \end{aligned} \tag{11}$$

where $H(k)$ is computed based upon the best state estimate available just before the measurement update. The discrete-time state equation (8) and the linearized measurement equation (10) are used to recursively estimate the state vector $X$ and the state covariance matrix $P$.

The Kalman filter consists of two parts: the measurement update, which improves the state estimate given a new measurement, and the time update, which propagates the state forward in time according to the system dynamics. Before each iteration of the Kalman filter, we know $Q(k)$ and $R(k)$ and we have estimates of $X(k)$ and $P(k)$. The measurement update is then performed according to the equations

$$\begin{aligned}\tilde{X}(k) &= \hat{X}(k) + K(k)[Z(k) - H(k)\hat{X}(k)] \\ \tilde{P}(k) &= [I - K(k)H(k)]\hat{P}(k)\end{aligned} \tag{12}$$

where $H(k)$ is computed from $\hat{X}(k)$ as described above and the Kalman filter gain $K(k)$ is computed using the equation

$$K(k) = \hat{P}(k)H^T(k)[H(k)\hat{P}(k)H^T(k) + R(k)]^{-1} \tag{13}$$

The time update equations are

$$\begin{aligned}\hat{X}(k+1) &= \Phi(k)\tilde{X}(k) + \Gamma(k)U(k) \\ \hat{P}(k+1) &= \Phi(k)\tilde{P}(k)\Phi(k)^T + \Gamma_d(k)Q(k)\Gamma_d(k)^T\end{aligned} \tag{14}$$

As noted above, the Kalman filter requires initial estimates for $X$ and $P$. The initial estimate for $X$ can be derived from the optic-flow equations and the perspective projection equations given a feature's location in two images and the camera's translational and rotational velocities, which are assumed constant during the interval between images. First, the optic-flow equations (3) are solved for $z_s$. The optic-flow equations actually comprise an overdetermined system of two equations in the one unknown $z_s$, so a single quadratic equation in $z_s$ is formed by summing the squares of the two optic-flow equations. Once $z_s$ is found, $x_s$ and $y_s$ can be determined from the perspective projection equations (1). The initial estimate of the state covariance matrix is chosen a priori.

This section has outlined the determination of range as a function of azimuth and elevation using a sequence of images from an electro-optical sensor. The range estimation algorithm described in this section has some characteristics that are similar to a electro-optical (EO) tracker used in infrared search and track (IRST) systems.[11] Both systems track features in successive images. The EO tracker tracks a single object, usually several kilometers away, after selection either manually or from a target recognition algorithm. It does not have the ability to detect and track several hundred features, with widely varying, optical-flow or track velocity, typical of a low-altitude scenario.

## III. Clustering Methodology

The output of the passive ranging algorithm may consist of 500–1000 discrete ranges. The actual number depends on the image sequence. Corresponding to each range $\rho_i$, we have the quintuple $(u_i, v_i; x_i, y_i, z_i, i = 1, 2, \ldots, n)$ where $(u_i, v_i)$ refers to the location of the $i$th point in the image plane, $(x_i, y_i, z_i)$ corresponds to the estimated range or location of this point in the three-dimensional world, and $n$ is the number of range points. The location $(x_i, y_i, z_i)$ can be expressed in camera, body, or earth coordinate systems. Each object in the image is represented by many features and the number of objects is small compared to $n$. The segmentation of the range map into groups, where each group corresponds to an object, is a combinatorial optimization problem.

Clustering[12] has been used for a long time in many disciplines, such as biology, geology, and psychiatry. In computer vision, clustering has been used for classification of multispectral data and image segmentation using attributes like gray level, color, texture, and gradient. Our clustering work is related to work in perceptual grouping using simulated annealing.[13] Ideas such as hierarchical approach and problem-dependent constraints are natural tools to reduce the amount of computation in practical problems. Strong search methods, the name for problem-dependent constraints in artificial intelligence, have been used to reduce the number of combinations in automatic target recognition.[14] The hierarchical approach allows the problem to be solved as a series of subproblems.

We first construct a set of higher confidence discrete range points by checking if the points existed in a number of range maps corresponding to previous images in the sequence. One of the advanatages of this step is that new or unreliable range points are eliminated. For the flight image range maps used in this paper only those points were selected that existed in at least 10 of the 15 previous range maps.

Next, we divide the range into groups based only on how far the objects are directly in front of the vehicle. It is assumed that some of the viewed physical objects are separable in depth. The depth values of the features, corresponding to each of these objects, are expected to have a distribution. For the sake of mathematical convenience we assume these distributions to be Gaussian. The grouping process involves the construction of a depth histogram and its approximation by a number of Gaussians. Grouping involves assigning the features to the groups defined by the Gaussians.

We then start with the grouping based on depth and split them based on their separation in the horizontal plane. The splitting operation is done within the groups. The rationale for separating in the horizontal plane is twofold: first, we prefer, low-altitude helicopter maneuvers that are limited to flying around the object and, second, objects that are at the same depth may be far apart in the horizontal plane. In such a situation a Euclidean distance measure may separate these objects in the horizontal plane. From a rotorcraft guidance point of view, if the separation between two objects is greater than the rotor diameter, then a path between the objects is of interest. If

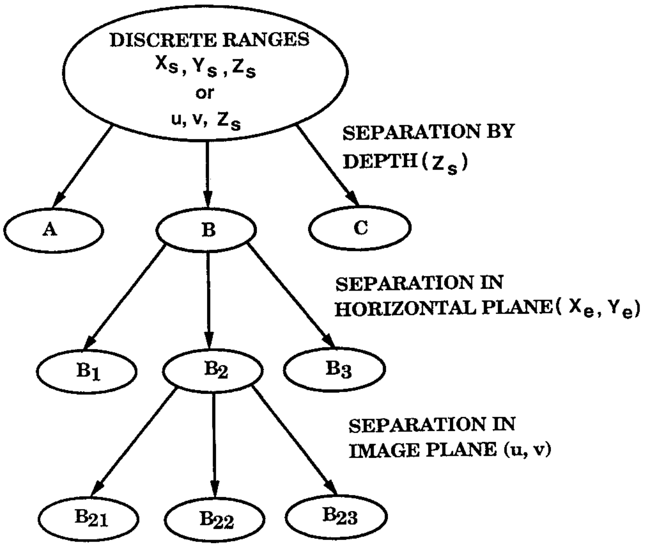

**Fig. 2 Hierarchical clustering method.**

the separation is not sufficient, then the two objects form a single obstacle (single object) in the flight path. With this point of view, the rotor diameter is the natural distance measure used by the $K$-means algorithm for constructing groups whose means are separated by at least the rotor diameter.

The next level of grouping takes place in the image plane. The computation of range in the optic-flow algorithms is usually done on a local basis to reduce the amount of computation. Thus the range computation does not take into account the interpixel relations between ranges belonging to the same group. The groups are examined for consistency in the image plane, and any groups with inconsistent members are subdivided. We use the MST algorithm to achieve grouping in the image plane.

The hierarchical steps are summarized in Fig. 2. The figure shows that the discrete range points are grouped into three groups, $A$, $B$, and $C$, based on fitting Gaussians to the depth histogram. The group $B$ is then partitioned into three groups, $B_1$, $B_2$ and $B_3$, by using the $K$-means algorithm in the horizontal plane. Finally, the group $B_2$ is further split into three groups, $B_{21}$, $B_{22}$, and $B_{23}$, by application of the MST algorithm in the image plane. The hierarchical clustering algorithm groups the discrete points into seven groups, $A$, $C$, $B_1$, $B_3$, $B_{21}$, $B_{22}$, and $B_{23}$ for this example.

## IV. Helicopter Flight Images

The clustering methodology for object segmentation described in the previous section will be applied to two helicopter flight image sequences. The first image sequence is named Line sequence and the second one is named Posts sequence. These image sequences are part of a database being developed at NASA to support the verification of computer vision algorithms. The flight image sequences were acquired by a camera mounted under the rotorcraft nose and oriented roughly in the direction of the flight so that the designated obstacles could be observed. The position of the camera and its orientation with respect to the rotorcraft were held constant throughout the flight. The details of the methodology used to develop the flight database consisting of imagery, rotorcraft and sensor parameters, and ground-truth range measurements are described in Ref. 15.

The line flight image sequence consists of 240 images. The 45th image is shown in Fig. 3 and the 60th image in Fig. 4. The objects in the view of the imaging sensor are labeled $A, B, \ldots, H$ in Fig. 3 where $A$, $B$, $C$, $D$, and $E$ are trucks on the runway, $F$ is the time stamp, $G$ is the runway, and $H$ is the rotorcraft nose boom. Trucks $A$, $B$, and $C$ are arranged with $A$ being closest to the camera and $C$ being farthest and $B$ in between. Truck $D$ is in front of truck $E$.

The posts flight image sequence consists of 90 images. The 45th and the 60th images are shown in Figs. 5 and 6, respectively. The various objects are labeled $A, B, \ldots, M$ in Fig. 5. The road is labeled $G$. Some of the posts along the side of the road are labeled $A, \ldots, F, H, \ldots, J$. A building in the back is labeled $K$. The time stamp and the rotorcraft nose boom are labeled $L$ and $M$, respectively.

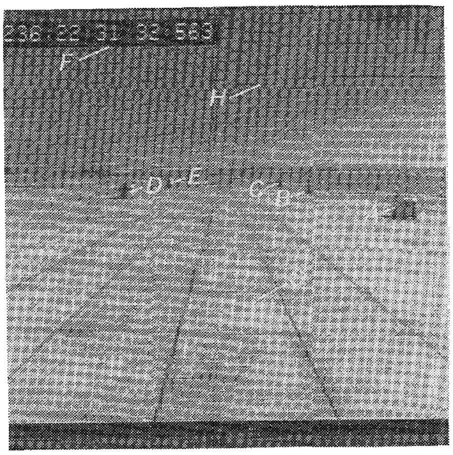

**Fig. 3 Forty-fifth image, line sequence.**

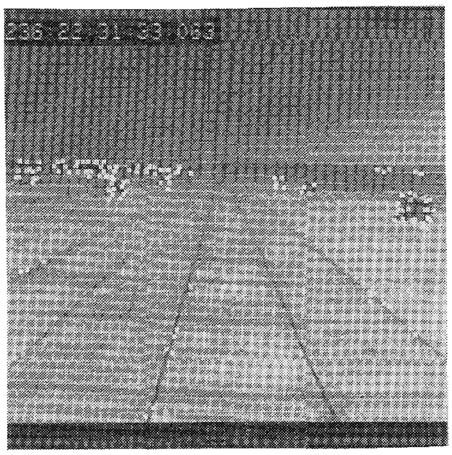

**Fig. 4 Sixtieth image, line sequence.**

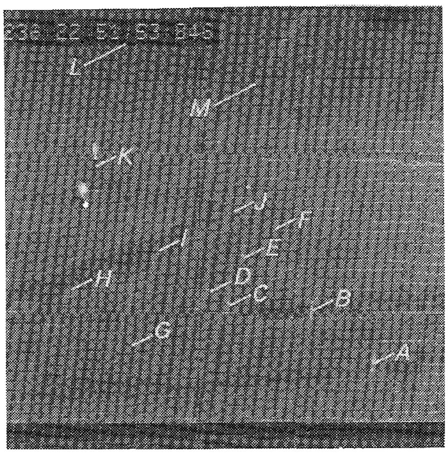

**Fig. 5 Forty-fifth image, posts sequence.**

## V. Separation in Depth

Several coordinate frames are encountered in relating object coordinates in the real world to its image coordinates in the camera image plane. The various coordinate frames are shown in Fig. 1. As described in Sec. II, the ranging algorithm outputs a sparse range map that consists of a set of features with the following attributes: the $u$ and $v$ location in the image plane, an identification tag, and a depth. The $u$, $v$, and depth are directly related to the $x$, $y$, and $z$ loca-

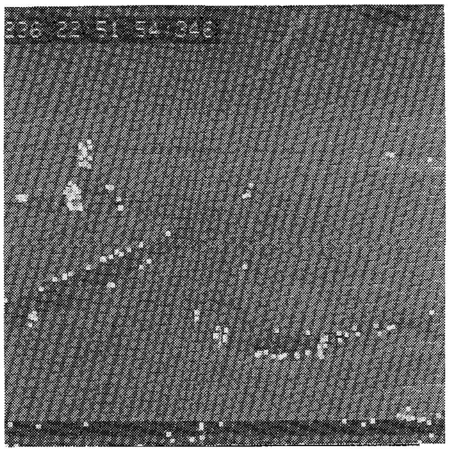

Fig. 6 Sixtieth image, posts sequence.

tions in the earth frame via the inverse perspective projection equations, the camera-to-body and body-to-earth transformation matrices. The identification tag helps identify the feature through a series of images.

It is assumed that some of the viewed physical objects are separable in depth. The depth values of the features, corresponding to each of these objects, may be expected to have a distribution. This is due to curvature of the object being viewed and incorrect depth estimates. For ease of mathematical modeling we assume these distributions to be Gaussian.

The first step in the grouping process is the construction of a histogram (see Figs. 7 and 8) consisting of the number of features as a function of the depth.

The second step in the grouping process is detection of the peaks of the histogram. A peak is defined as a maximum bounded on both sides (left and right) by minima such that the difference between the peak and the valleys exceeds a threshold ("peakiness"). A peak is determined by bracketing it between a proper left minimum and a proper right minimum such that the peakiness criterion is satisfied. A proper minimum is determined by bracketing the minimum between two peaks such that the peakiness criterion is satisfied with respect to the bounding peaks. For the histograms in Figs. 7 and 8, the circles around the peaks show the detected peaks. The peakiness value used was 0.1. Next, we approximate the histogram as a sum of $m$ Gaussians. The approximation to the histogram is achieved by minimizing the sum of the squares of the error:

$$\min_{k_j,\mu_j,\sigma_j} \sum_{1\le i\le n} e_i^2, \qquad 1 \le j \le m \tag{15}$$

where

$$e_i = \zeta_i - \sum_{1\le j\le m} k_j \exp\left(-\frac{(\delta_i-\mu_j)^2}{2\sigma_j^2}\right) \tag{16}$$

Here, $k_j$ is the scale factor, $\mu_j$ is the mean, and $\sigma_j$ is the standard deviation of the $j$th Gaussian. At $n$ depth locations, the depth is $\delta_i$ and the normalized frequency is $\zeta_i$. For minimization, a MINPACK[16] routine LMDIF1 is used. The routine LMDIF1 is a modified version of the Levenberg-Marquardt algorithm. Based on the detected peaks, an initial estimate of the $k_j$, $\mu_j$, and $\sigma_j$ for each Gaussian is provided to the minimization routine. The individual Gaussians approximating the depth histogram (in Fig. 7) are shown in Fig. 9. Similarly, the three Gaussians approximating the Posts flight sequence depth histogram (see Fig. 8) are shown in Fig. 10.

Finally, the features are assigned to the groups represented by the Gaussians. For this purpose, consider two Gaussians next to each other. In order for them to intersect, the following relationship must hold:

$$k_1 \exp\left(-\frac{(\delta-\mu_1)^2}{2\sigma_1^2}\right) = k_2 \exp\left(-\frac{(\delta-\mu_2)^2}{2\sigma_2^2}\right) \tag{17}$$

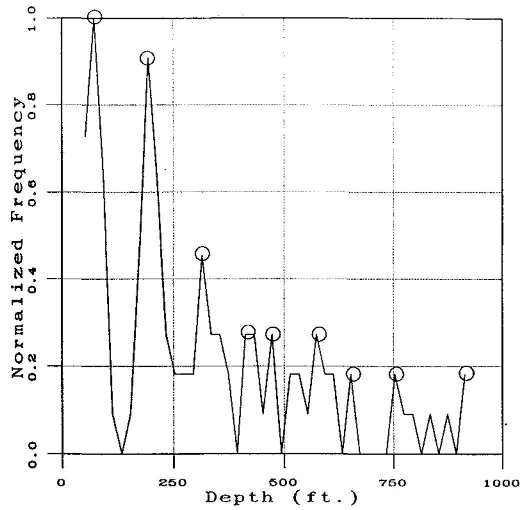

Fig. 7 Depth histogram, line sequence.

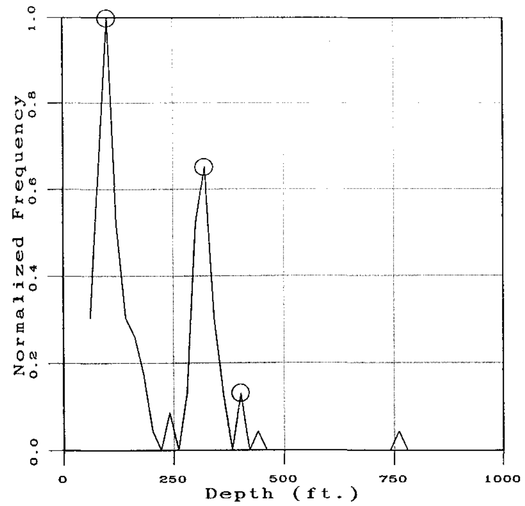

Fig. 8 Depth histogram, posts sequence.

This simplifies to:

$$\left(\sigma_2^2-\sigma_1^2\right)\delta^2 - 2\left(\mu_1\sigma_2^2-\mu_2\sigma_1^2\right)\delta + \left(\mu_1^2\sigma_2^2-\mu_2^2\sigma_1^2\right) - 2\sigma_1^2\sigma_2^2 \ln\frac{k_1}{k_2} = 0 \tag{18}$$

This equation may be solved to obtain the depth $\delta$ at which the two Gaussians intersect. It may be noted that if the standard deviations $\sigma_1$ and $\sigma_2$ are equal, Eq. (18) is reduced to a linear equation.

The intersection depth marks the outer limit of Gaussian 1 and inner limit of Gaussian 2. By repeating this procedure, inner and outer limits are determined for each Gaussian. The minimum depth marks the inner limit for the first Gaussian and the maximum depth marks the outer limit of the last Gaussian. Each feature is allocated to one of the Gaussians based on whether the depth corresponding to the feature lies within the inner or outer limits of that Gaussian. The resulting grouping is coded in a matrix $G$, where the number of columns is equal to the number of groups and the identification tags of all features belonging to a particular group are stored in a single column. The number of members in each group is coded in a vector $N$. This way each member of the group is accessible via the vector $N$ and the matrix $G$.

## VI. Separation in Horizontal Plane

The previous section was devoted to grouping based on depth only. In this section we start with the grouping based on depth and split them based on their separation in the horizontal ($x_e$-$y_e$ plane in the earth axes) plane. The splitting operation is done within the

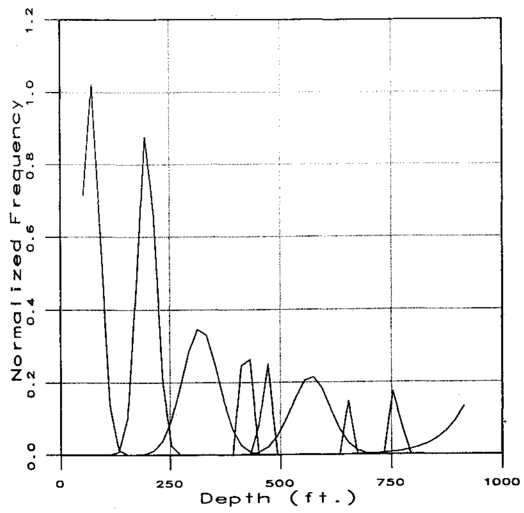

Fig. 9 Gaussian approximation, line sequence.

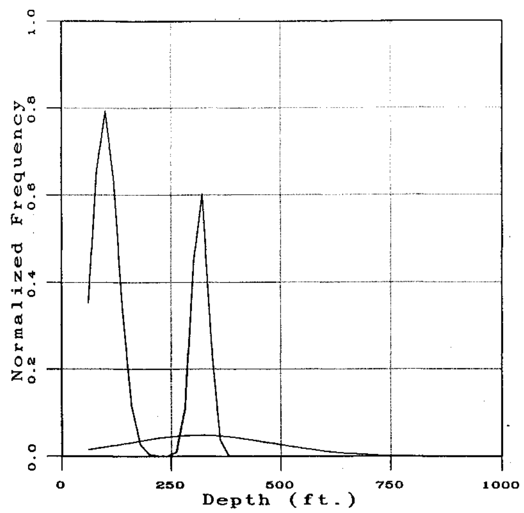

Fig. 10 Gaussian approximation, posts sequence.

groups. For example, three objects at the same depth but separated along the $y_e$ axis will be split into three groups by this method.

Given the $u$, $v$, and $z_s$ (depth) coordinates of the features in the image plane, the $x_s$ and $y_s$ locations of the object in the camera frame may be found by using the perspective projection equations (1) as

$$x_s = \frac{uz_s}{f} \tag{19}$$

$$y_s = \frac{vz_s}{f} \tag{20}$$

where $f$ is the focal length of the camera.

In the body frame, the feature location is given as

$$\begin{Bmatrix} x_b \\ y_b \\ z_b \end{Bmatrix} = \begin{Bmatrix} l_{xs} \\ l_{ys} \\ l_{zs} \end{Bmatrix} + T_{b/s} \begin{Bmatrix} x_s \\ y_s \\ z_s \end{Bmatrix} \tag{21}$$

Here $l_{xs}$, $l_{ys}$, and $l_{zs}$ are the coordinates of the camera and $x_b$, $y_b$, and $z_b$ are the coordinates of the feature, both with respect to the body in the body frame. The term $T_{b/s}$ is the transformation matrix from the camera frame to the body frame. Similarly, the feature position in the earth frame is given as

$$\begin{Bmatrix} x_e \\ y_e \\ z_e \end{Bmatrix} = \begin{Bmatrix} l_{xb} \\ l_{yb} \\ l_{zb} \end{Bmatrix} + T_{e/b} \begin{Bmatrix} x_b \\ y_b \\ z_b \end{Bmatrix} \tag{22}$$

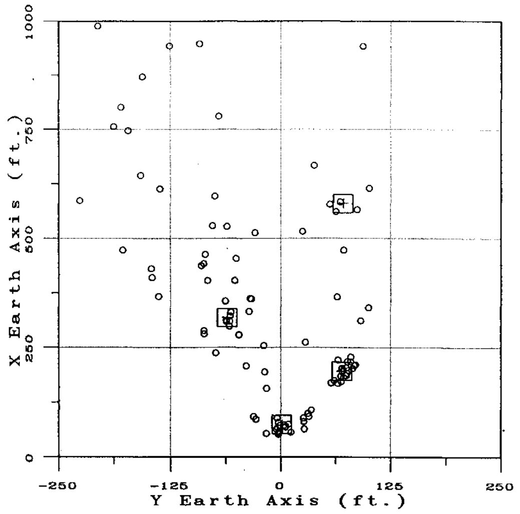

Fig. 11 K-means clustering, line sequence.

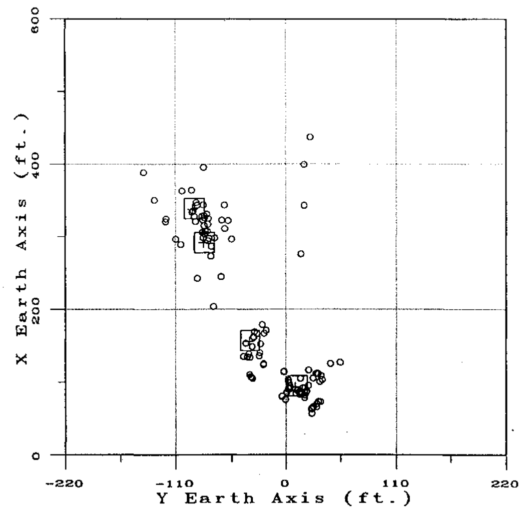

Fig. 12 K-means clustering, posts sequence.

where $x_e$, $y_e$, and $z_e$ are the coordinates of the feature in the earth frame; $l_{xs}$, $l_{ys}$, and $l_{zs}$ are the coordinates of the body in the earth frame, and $T_{e/b}$ is the transformation matrix from the body frame to the earth frame.

Once the features are transformed to the earth frame, the $z_e$ coordinate (altitude) is ignored and the $K$-means algorithm, described below, is used for grouping them in the horizontal plane.

The $K$-means algorithm may be summarized as follows.[17] Given a group of features, the group mean is found. Next, a feature that is farthest from the mean is found. We will call this feature the new pivot. All the features are then allocated to the two groups based on their closeness to the mean or the pivot. New means for these two groups are determined. This completes the initialization of the $K$-means grouping. Next, a new pivot feature is found based on its distance from the two group means. For this purpose a minimum distance from a feature to the groups is computed. This is the smallest distance between a feature and any of the group means. This is repeated for all features, and the feature whose minimum distance to the groups is a maximum is selected as the pivot for the new group. Features are then allocated among the previous groups and the new group based on their closeness to the group means and the pivot. The group means are then updated. If the distance between any two group means is less than the specified rotor diameter, the groups are merged. This process is continued until it is no longer possible to create a new group. Finally, groups with a small number of features are eliminated.

The above process is repeated for all the groups obtained previously by separation in depth. For the example groups, shown in

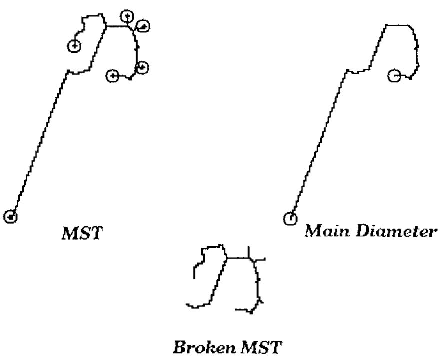

**Fig. 13 MST for truck A, line sequence.**

Figs. 9 and 10, the results of the $K$-means grouping algorithm are shown in Figs. 11 and 12. The $K$-means algorithm regroups the nine groups into four groups for the Line sequence and three groups into four groups for the Posts sequence. The group means are demarcated by squares in Figs. 11 and 12. For these examples, a rotor diameter of 60 ft was used. Groups with less than five features were eliminated.

## VII. Separation in the Image Plane

The basic motivation for grouping in the image plane is that features that could not be separated based on depth alone or by the separation algorithm in the horizontal plane may be widely separated in the image plane and, therefore, may belong to different objects. Another aspect of grouping in the image plane is that $(u, v)$ are direct measurements, whereas $(x_e, y_e, z_e)$ locations are estimated values. One of the possible ways of separating objects in the image plane is by using the $K$-means algorithm. However, unlike in the horizontal plane, where the rotor diameter was used as a distance measure, no such measure is directly available in the image plane for use by the $K$-means algorithm. The MST provides a relationship between the features that may be used to construct reasonable distance measures. A spanning tree is defined as a connected set (contains paths between any pair of points) with no loops that contains all points of the problem. The MST of a set is that spanning tree whose sum of the edge lengths (distance between connected points) is a minimum. It has been shown in Ref. 10 that properties of the MST may be used to successfully cluster a point set into groups such that points within the group are close to each other.

The application of the MST algorithm[18] to the group corresponding to truck A (see Fig. 3), obtained after separation in depth and in the horizontal plane, is shown in Fig. 13. For clarity the MST has been shown enlarged. End nodes are defined as nodes of the MST that have only one neighbor. For the example MST shown in Fig. 13, some of the end nodes are encircled. The main diameter of the MST is defined as the longest path connecting two end nodes. This path may be thought of as the trunk of the tree.

To this point, the only information directly coded in the MST data structure is of local nature. However, the global relationships between the nodes may be extracted by starting at an end node and creating a parent-node-children-node tree structure. We start with an end node and assign it zero weight. Each node, except the starting node, has a parent node and may have one or more children nodes. Each parent-child pair has a weight (distance) assigned to it. The weight associated with each node is equal to the sum of the weight of the parent plus the associated parent-child weight. The node that has the largest weight forms the other end of the diameter. This process is repeated for all the end nodes. The end-node pair that has the largest weight forms the main diameter. The main diameter computed by the above algorithm is shown next to the MST in Fig. 13. The two nodes shown encircled on the main diameter are the end nodes. After determining the main diameter, the next step is to determine inconsistencies along the main diameter. We start with an element on the main diameter and examine the nodes near the two ends of the element and compute the average weights. If the element weight is significantly different than the average weights of the element end nodes, the element is inconsistent. The average weight of the end node is determined by using the weights of the elements near the end node. The broken MST is shown below the MST in Fig. 13. The MST is broken up in places where inconsistencies were detected.

This procedure is applied to all the groups resulting from the separation in the horizontal plane.

## VIII. Results

All the algorithms described above were applied sequentially to group the features shown in Figs. 4 and 6. The results are described below. As shown in Figs. 7 and 9, the depth histogram of the Line sequence features is approximated by nine Gaussians; therefore, all features are members of one of the nine groups. Similarly, the features of the Posts sequence are members of the three groups corresponding to the three Gaussians shown in Fig. 10. These Gaussians approximate the depth histogram shown in Fig. 8. The resulting nine groups for the Line sequence and three groups for the Posts sequence were then processed by the $K$-means algorithms with small group elimination to regroup the features based on their separation in the horizontal plane. For these examples a distance of 60 ft between the means was used as the merging criterion. The algorithm resulted in four groups for both sequences, as may be seen in Figs. 11 and 12. The MST for the groups of the Line sequence are shown in Fig. 14. The four groups based on the separation in the horizontal plane were further processed by the MST algorithms for regrouping based on their separation in the image plane. It may be noted that each group was broken up if inconsistencies were detected along the main diameter of their MST. The broken MSTs for the Line sequence are shown in Fig. 15. The broken MSTs are regrouped into new groups by the MST grouping algorithm. For the Line sequence example, 6 groups shown in Fig. 16 are obtained after MST grouping. Similarly for the Posts sequence, 11 final groups shown in Fig. 17 are obtained. A comparison of Figs. 16 and 17 with Figs. 3 and 5, respectively, indicates that the object segmentation algorithm is able to group most objects correctly. The algorithm cannot identify objects like the helicopter nose boom (see objects $H$ in Fig. 3 and $M$ in Fig. 5) for which no range estimates are available.

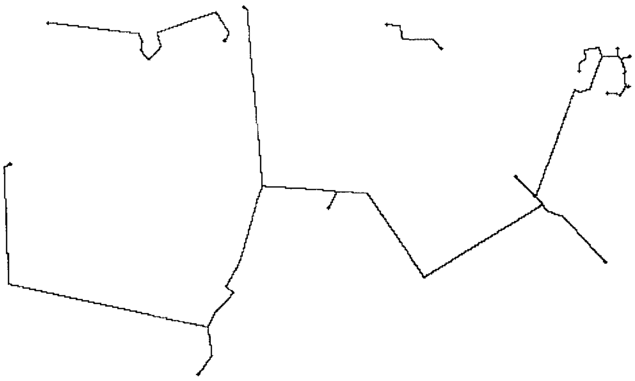

**Fig. 14 MST, line sequence.**

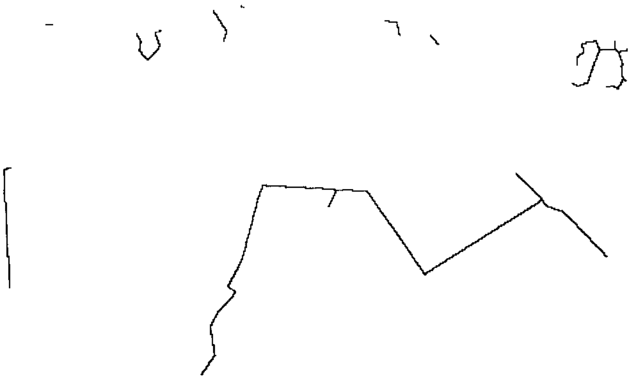

**Fig. 15 Broken MST, line sequence.**

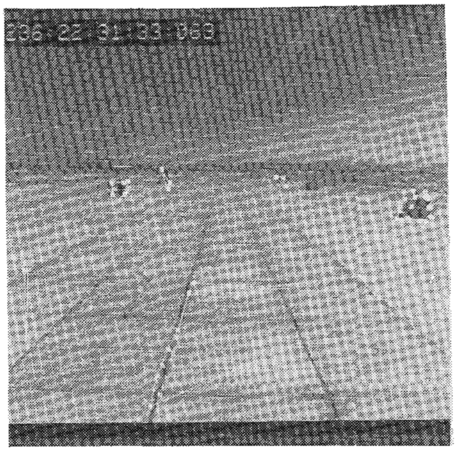

Fig. 16 Final groups, line sequence.

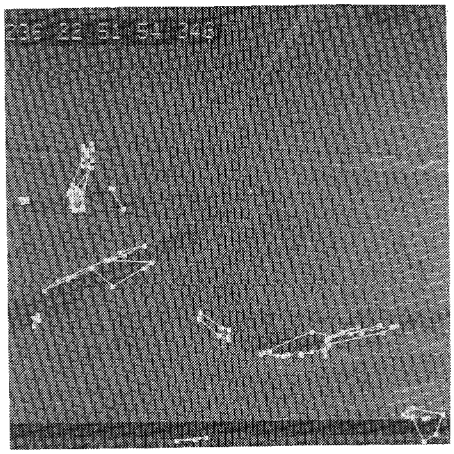

Fig. 17 Final groups, posts sequence.

## IX. Concluding Remarks

We have described an approach to cluster range estimates from a passive sensor into groups of objects useful in building a world model that can be used for obstacle avoidance during helicopter flight. The approach consists of hierarchical grouping methodology combined with problem-dependent constraints to reduce the number of combinations in the problem. Object segmentation results were presented using two helicopter flight images.

Successful grouping using the sum of Gaussians to represent the depth histogram depends on the number of features representing the object and the separation between the objects. The first property affects the height of the Gaussians and therefore that threshold that would detect the peak. The second property results in small overlap of the fitted Gaussians. Usually objects that are far away will not be grouped separately but will be grouped with a larger object that is relatively closer to the camera. This is because objects that are far away occupy few pixels in the image and, therefore, only a few range features are determinable that correspond to this object. Both peak detection and approximation with Gaussian algorithms have been extensively tested on a number of images. The algorithms perform consistently and correctly.

For separation in the horizontal plane the $K$-means split-and-merge algorithm was developed. The idea for the $K$-means algorithm is not new. The key idea is the maximum minimum distance by which the next possible group is started. Usually, the $K$-means algorithm employs a distance measure as a stopping criterion. In other words, when the distance between any two means is smaller than a specified number, the algorithm is stopped. This criterion results in at least one group whose diameter is approximately the size of the specified distance. This occurs when the groups are not widely separated. In our version of the algorithm we have not used the distance measure in this sense; rather, if the distance between the two means is less than the distance measure, we group the two sets of features, represented by the two means, as one group. This is the merging operation. If no new groups are created by repeating the algorithm, the algorithm is stopped. The $K$-means split-and-merge algorithm was tested on a number of images. In all cases the algorithm converged.

The MST concept was implemented by developing the algorithmic details for the design of the data structure, determination of the end nodes, extraction of the main diameter, checking for inconsistencies along the main diameter, and grouping using the MST. The algorithm has been tested on a number of images.

The object segmentation algorithm is able to group most objects correctly. The combination of range estimation algorithm and the grouping technique described in this paper provides a basis for the object information required for helicopter guidance.

## References

[1] Sridhar, B., and Phatak, A. V., "Analysis of Image-Based Navigation System for Rotorcraft Low Altitude Flight," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 22, No. 2, 1992, pp. 290–299.

[2] Sridhar, B., Suorsa, R., and Hussien, B., "Passive Range Estimation for Rotorcraft Low Altitude Flight," *Machine Vision and Applications*, Vol. 6, No. 1, 1993, pp. 10–24.

[3] Menon, P. K. A., Chatterji, G. B., and Sridhar, B., "Passive Obstacle Location for Rotorcraft Guidance," AIAA Guidance, Navigation, and Control Conference, Vol. 2, New Orleans, LA, Aug. 1991, pp. 684–691.

[4] Bhanu, B., Roberts, B., and Ming, J. C., "Inertial Navigation Sensor Integrated Motion Analysis," *Proceedings of the DARPA IUW*, May 1989, pp. 747–763.

[5] Chellappa, R., and Shekar, C., "Passive Ranging Using a Moving Camera," *Journal of Robotic Systems*, Oct. 1992, pp. 729–752.

[6] Matthies, L., "Stereo Vision for Planetery Rovers: Stochastic Modeling to Near Real-Time Implementation," Jet Propulsion Laboratory, Pasadena, CA, JPL D-8131, Jan. 1991.

[7] Kweon, I. S., and Kanade, T., "High-Resolution Terrain Map from Multiple Sensor Data," *IEEE Transactions on Pattern Analysis Machine Intelligence*, Vol. 14, No. 2, 1992, pp. 278–293.

[8] Besl, P. J., and Jain, R. C., "Segmentation Through Variable-Order Surface Fitting," *IEEE Transactions on Pattern Analysis Machine Intelligence*, Vol. 10, No. 2, 1988, pp. 167–190.

[9] Krishnapuram, R., and Casasent, D., "Determination of Three-Dimensional Object Location and Orientation from Range Images," *IEEE Transactions on Pattern Analysis, Machine Intelligence*, Vol. 11, No. 11, 1989, pp. 1158–1167.

[10] Zahn, C. T., "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," *IEEE Transactions on Computers*, Vol. C-20, No. 1, 1971, pp. 68–86.

[11] Accetta, J. S., Conley, T. D., Steinberg, R., Maxwell, J. R., and Kryskowski, D., "Infrared Search and Track Systems," *The Infrared and Electro-Optical Systems Handbook, edited by S. B. Campana*, Vol. 5, Environmental Research Institute of Michigan and SPIE Optical Engineering Press, Bellingham, WA, 1993, pp. 209–344.

[12] Andrews, H. C., *Mathematical Techniques in Pattern Recognition*, Wiley-Interscience, New York, 1972.

[13] Kahn, P., Winkler, A., and Chong, C. Y., "Perceptual Grouping as Energy Minimization," *Proceedings of the International Conference on Systems, Man and Cybernetics* (Los Angeles, CA), 1990, pp. 542–546.

[14] Rao, K., "Combinatorics Reduction for Target Recognition in ATR Applications," *Proceedings of the SPIE Conference on Automatic Object Recognition II*, Orlando, FL, April 22–24, 1992, Vol. 1700, pp. 285–295.

[15] Smith, P. N., "Flight Data Acquisition Methodology for Validation of Passive Ranging Algorithms for Obstacle Avoidance," NASA TM 102809, Oct. 1990.

[16] Garbow, B. S., Hillstorm, K. E., and More, J. J., "Documentation for MINPACK Subroutine LMDIF1," Appl. Math. Div., Argonne National Laboratory, Argonne, IL, March 1980.

[17] Gonzales, R. C., and Tou, J. T., *Pattern Recognition Principles*, Addison-Wesley, Reading, MA, 1981.

[18] Prim, R. C., "Shortest Connection Networks and Some Generalizations," *Bell System Technical Journal*, Nov. 1957, pp. 1389–1401.