NASA Technical Memorandum 103897

p. 37

Passive Range Estimation for Rotorcraft Low-Altitude Flight

B. Sridhar, R. Suorsa, and B. Hussien

(NASA-TM-103897) PASSIVE RANGE ESTIMATION FOR ROTORCRAFT LOW-ALTITUDE FLIGHT (NASA) 37 p

CSCL 01B

N92-10003

Unclas

G3/01 0048782

October 1991

Quick Release – This Technical Memorandum is a preliminary, unedited report. It is being released in this format to quickly provide the research community with important information.

National Aeronautics and Space Administration

NASA Technical Memorandum 103897

Passive Range Estimation for Rotorcraft Low-Altitude Flight

B. Sridhar, R. Suorsa, and B. Hussien, Ames Research Center, Moffett Field, California

October 1991

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

PASSIVE RANGE ESTIMATION FOR ROTORCRAFT LOW ALTITUDE FLIGHT

B.Sridhar, R.Suorsa and B. Hussien
NASA Ames Research Center
Moffett Field, CA 94035

Abstract

The automation of rotorcraft low-altitude flight presents challenging problems in control, computer vision and image understanding. A critical element in this problem is the ability to detect and locate obstacles, using on-board sensors, and modify the nominal trajectory. This requirement is also necessary for the safe landing of an autonomous lander on Mars. This paper examines some of the issues in the location of objects using a sequence of images from a passive sensor, and describes a Kalman filter approach to estimate range to obstacles. The Kalman filter is also used to track features in the images leading to a significant reduction of search effort in the feature extraction step of the algorithm. The method can compute range for both straight line and curvilinear motion of the sensor. An experiment is designed in the laboratory to acquire a sequence of images along with the sensor motion parameters under conditions similar to helicopter flight. The paper presents range estimation results using this imagery.

1 Introduction

Rotorcraft operating in high-threat environment fly close to the earth's surface to utilize surrounding terrain, vegetation, or man-made objects to minimize the risk of being detected by the enemy. Increasing levels of concealment are achieved by adopting different tactics during low-altitude flight. The piloting of the rotorcraft is, at best, a very demanding task and the pilot will need help from on-board automation tools in order to devote more time to mission-related activities. The development of an automation tool, which has the potential to detect obstacles in the rotorcraft flight path, warn the crew, and interact with the guidance system to avoid detected obstacles, presents challenging problems.

The planning of rotorcraft low-altitude missions can be divided into far-field planning and near-field planning [1]. Far-field planning involves the selection of goals and a nominal trajectory between the goals. Far-field planning is based on a priori information and requires

a detailed map of the local terrain. However, the database for even the best surveyed landscape will not have adequate resolution to indicate objects such as trees, buildings, wires and transmission towers. This information has to be acquired using an on-board sensor and integrated into the navigation/guidance system to modify the nominal trajectory of the rotorcraft. Because vision alone will not be adequate for detecting small obstacles such as wires, it is expected that the system will include an active sensor whose search can be directed to complement the vision system while minimizing the risk of detection [2]. Initially, passive imaging sensors such as forward looking infrared (FLIR) and low-light-level-television (LLLTV) will be considered in order to assess the operational potential of passive methods.

The recovery of depth using electro-optical sensors, referred to as passive ranging, is based on triangulation and requires two images of the outside world from two different imaging conditions. In stereo methods, two or more cameras located at different positions are used to obtain images of the outside world. In motion methods, the same camera is moved from one position to another to capture two or more images of the outside world. Passive ranging has been the subject of considerable study in computer vision [3, 4]. Several papers have described the use of a sequence of images to determine the orientation and position of a rigid body [5], recover both vehicle motion and location of surrounding objects [6], compute range by matching vertices of man-made objects [7], and estimate range for linear motions of the vehicle [8, 9]. Krotkov [10] has used focusing to compute the distance to objects lying in the range 1 to 3 meters. Recently, Ma and Olsen [11] have proposed the acquisition of a monocular image sequence by changing the focal length of the camera and using the resulting images for range computation.

The passive ranging methods described earlier have evolved from the needs of the Autonomous Land Vehicle (ALV) and other ground-based robotic applications. Road following is the key guidance function in the ALV whereas the ability to maneuver around obstacles is the challenge for guidance in low altitude flight. The rotorcraft flight at low-altitude has several distinct characteristics: (i) the nature of the scenarios encountered outdoors makes methods dependent on model based vision such as matching verticies of man-made objects not applicable, (ii) due to the curvilinear motion of the rotorcraft, a large class of passive ranging algorithms designed for linear flight are not directly applicable, (iii) the objects of interest during a rotorcraft flight may vary from 50 to 1000 feet resulting in a large variation in the optical flow in the image, and (iv) the availability of the sensor motion parameters from an inertial navigation system. The approach used in this analysis is geared to the requirements of low altitude rotorcraft flight and differs from general motion analysis methods in significant ways. The distinguishing features of our approach are: (i) we do not attempt to estimate the rotorcraft's motion from the images, (ii)the algorithms are designed to be recursive, (iii) the algorithms can handle curvilinear rotorcraft motion, (iv) feature tracking and range estimation are done incrementally to reduce the search space as well as to discard false matches, and (v) a Kalman filter formulation allows the use of several optical sensors and it provides for a natural way of integrating stereo and motion methods. The rotorcraft parameters (angular rate, translational velocity, position and attitude) are assumed to be

computed using an on-board inertial navigation system. Given a sequence of images, using image-object differential equations, a Kalman filter [12] can be used to estimate both the relative coordinates and the earth coordinates of objects on the ground. The Kalman filter can also be used in a predictive mode to track features in the images, leading to a significant reduction of search effort in the feature extraction step of the algorithm. The performance of three different Kalman filters for different rotorcraft maneuvers were examined in [12]. This previous study did not, however, include the processing of real images. This paper describes the computation of the optical flow and uses the resulting optical flow in an Extended Kalman Filter (EKF) to estimate range. We present range estimation results for both linear and curvilinear motions of the camera. The experience gained from the application of this algorithm to real images is very valuable and it is a necessary step before proceeding to the estimation of range during low-altitude curvilinear flight using an Extended Kalman Filter (EKF).

The paper is organized as follows: Section 2 describes the relation between the image, the rotorcraft and objects of interest under full curvilinear motion. Section 3 describes our approach to the estimation of optical flow and the resulting image track. Section 4 describes the recursive range estimation algorithm and Section 5 considers the performance of this algorithm using image data acquired in a laboratory setting. Finally, Section 6 provides a summary, conclusions and ideas for future work.

2 Passive Ranging

Passive ranging is the ability to estimate distances to various objects close to the flight path of the helicopter using passive sensing by electro-optical cameras. This section will describe the basic relations between the two-dimensional (2D) sensor image variables (displacement and velocity) to the three-dimensional (3D) terrain geometry (i.e. points, lines and other features) and sensor motion parameters (i.e. position, attitude, translational and rotational velocity). These relations provide the dynamic models for the estimation of range. For simplicity, the camera is assumed to be fixed at the center of gravity of the rotorcraft with its optical axis oriented along the rotorcraft's longitudinal body axis. Figure 1 shows the viewing geometry of the camera. In actual practice, the camera is mounted at a convenient location away from the center of gravity of the rotorcraft. If necessary, the camera is allowed variable orientation with respect to the body of the rotorcraft. This flexibility is provided in our implementation of the passive ranging algorithm.

Consider an earth-fixed, north-east-down, coordinate system. Let $r_h \equiv (x_h, y_h, z_h)^T$ and $r \equiv (x, y, z)^T$ be the earth coordinates of the rotorcraft and a point O on the ground respectively. The rotorcraft moves with respect to the earth at a translational velocity $V \equiv (V_x, V_y, V_z)^T$.

The orthonormal coordinate transformation from the earth axes to body axes is denoted by the 3x3 matrix T which depends on the rotorcraft attitude . Let T_s be the corresponding

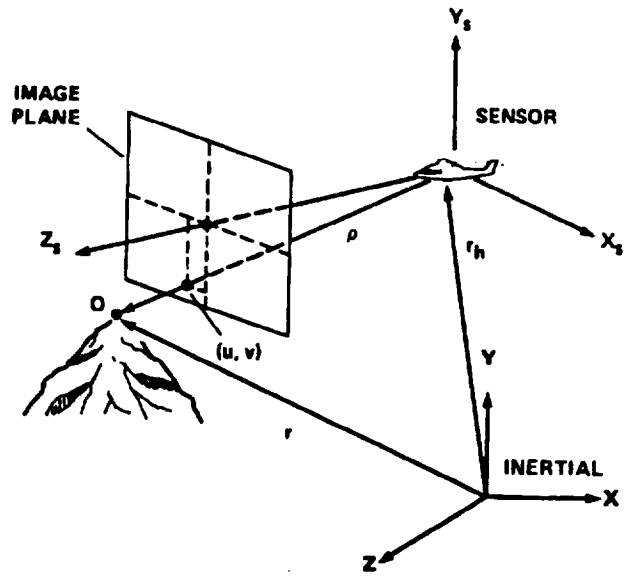

Figure 1: Viewing geometry of the imaging sensor

transformation matrix from body to sensor-axis system.

The relative position, ρ, of any point O with respect to the rotorcraft can be written as

$$\rho = r - r_h \tag{1}$$

The rate of change of this vector as viewed from the moving rotorcraft can be determined by the Coriolis equation relating rate of change of a vector as viewed from a earth-fixed coordinate frame to that viewed from a moving frame. The standard vector form of the Coriolis equation [13] is

$$\dot{\rho}_{fixed} = \dot{\rho}_{moving} + \omega \times \rho \tag{2}$$

Let $\rho_b \equiv [x_b, y_b, z_b]^T$ be the vector between the rotorcraft and point O expressed in body-coordinates. Similarly, $V_b \equiv [V_{bx}, V_{by}, V_{bz}]^T$ and $\omega_b \equiv [\omega_{bx}, \omega_{by}, \omega_{bz}]^T$ represent the translational and rotational velocity of the rotorcraft in body axes. Let $\rho_s \equiv [x_s, y_s, z_s]^T$, $V_s \equiv [V_{sx}, V_{sy}, V_{sz}]^T$ and $\omega_s \equiv [\omega_{sx}, \omega_{sy}, \omega_{sz}]^T$ be the corresponding quantities in the sensor-axes system.

Using equations (1) and (2) and matrices T and T_s, we can write the relation

$$\dot{\rho}_s = -\omega_s \times \rho_s - V_s \tag{3}$$

Let the image plane be perpendicular to the optical axis. Then, using similar triangles

$$\begin{aligned} u &= f x_s / z_s \\ v &= f y_s / z_s \end{aligned} \tag{4}$$

where f is the focal length of the sensor. As the rotorcraft moves, the image of the object O moves in the image plane. Differentiating u and v with respect to time, we have

$$\begin{aligned} \dot{u} &= f(-x_s \dot{z}_s + z_s \dot{x}_s)/z_s^2 \\ \dot{v} &= f(-y_s \dot{z}_s + z_s \dot{y}_s)/z_s^2 \end{aligned} \tag{5}$$

Substituting equation (3) into (5) and (4) leads to the result

$$\begin{aligned} \dot{u} &= \dot{u}_t + \dot{u}_r \\ \dot{v} &= \dot{v}_t + \dot{v}_r \end{aligned} \tag{6}$$

where

$$\begin{aligned} \dot{u}_t &= (-V_{sx} + V_{sz} u/f) f/z_s \\ \dot{v}_t &= (-V_{sy} + V_{sz} v/f) f/z_s \\ \dot{u}_r &= \omega_{sx} uv/f - \omega_{sy}(u^2/f^2 + 1) f + \omega_{sz} v \\ \dot{v}_r &= \omega_{sx}(v^2/f^2 + 1) f - \omega_{sy} uv/f - \omega_{sz} u \end{aligned} \tag{7}$$

The velocity $(\dot{u}, \dot{v})$ associated with each point in an image is referred to as optical flow [14] where $(\dot{u}_t, \dot{v}_t)$ denotes the translational component of the optical flow, and $(\dot{u}_r, \dot{v}_r)$ the rotational component of the optical flow. The rotational component of the optical flow is a function of the image position only and provides no information about the location of object O. Thus, given the six motion parameters of the sensor (V_s, ω_s), by computing the optical flow $(\dot{u}, \dot{v})$, the translational component of the flow at a point can be used to estimate the distance of the point O along the optical axis $(0, 0, z_s)$. For a rotorcraft flying in a straight line, the optical flow will be zero at a point $(V_{sx}/V_{sz}, V_{sy}/V_{sz})$ in the image plane. This point (f_x, f_y), referred to as the focus-of-expansion (FOE), corresponds to the intersection of V_{sz} and the image plane and it plays an important role in optical flow computations. We do not use the direct approach suggested by equation (6) in our computation of z_s; instead, the range is computed using the more robust recursive estimation procedure outlined in Section 4.

Given a sequence of measurements $u(k), v(k); k = 1, 2, ...N$, using image-object differential equations (3), we estimate both the relative coordinates $\rho(k)$ and the earth coordinates r of the corresponding object O on the ground. The range estimation consists of two major parts: (a) computation of optical flow by extracting of measurements $(u(k), v(k))$ from the image and (b) estimation of range given the sequence of measurements $u(k), v(k); k = 1, 2, ...N$. The next section will consider the computation of the optical flow.

3 Computation of Optical Flow

The computation of optical flow requires the determination of the displacement of image points over a sequence of images. The main difficulty in the computation is due to the assumption that an object in the terrain space corresponds to a unique point in the image. In an actual image, an object on the ground is more likely to be a region in the image. Another complication in the computation of the optical flow, referred to as the correspondence problem [15], results from the ambiguity in identifying features in two images that are projections of the same entity in the 3-dimensional world.

There are two approaches to the computation of optical flow: (a) field-based techniques and (b) feature-based techniques. The field-based techniques assume a continuous variation of image intensity as a function of position and time. This approach was introduced by Horn [16, 17] and provides a dense map of the optical flow. However, the computational experience with this approach is limited to simple scenes and available algorithms based on this method are very susceptible to noise [18]. Recently, several new field-based methods have been reported in the literature [19, 20, 21, 22]. However, in this paper, we will restrict our attention to feature-based computation of optical flow.

Feature-based techniques [23] make use of features in an image to measure optical flow. Features in an image can be points, lines, contours, regions, or any other geometrical definition that corresponds to a distinguishable part of an object. The complexity of the algorithm depends on the definition of feature and the criteria used for matching. In general, robustness and computational load increase with the amount of modeling involved in the definition of the feature. The matching technique used in this paper is a modified version of area-based matching and is designed to overcome some of the common limitations associated with area-based matching [24]. The selection of areas for matching, based on regions of high pixel variance, eliminates featureless regions in the image. The use of normalized correlation makes matching independent of brightness changes from one image to the next. The knowledge of (V_s, ω_s) and the estimated value of range is used to limit the search area used for matching. The choice of window size affects the performance of matching algorithms. A large window can provide more uniqueness in matching. However, a large window may also cover more than one object and lead to erroneous results near occlusion boundaries. Some authors have addressed this problem by using several window sizes in a hierarchy [25]. Our approach is to use a fixed size window and to discard inconsistent features (i.e. features which do not behave as predicted) during recursive range estimation.

The algorithm used for computing optical flow between two successive image frames I_k and I_{k+1} consists of the following steps:

1. The kth image frame $I_k(u, v)$ is a discrete 2D function where the value of the function at each point (u, v) represents a shade of grey. In our implementation, the grey scale is represented by 8 bits and the image plane is 512 $\times$ 512 pixels.

2. Each image I_k is divided into regions of $n_p \times n_p$ pixels. A feature, S_p, is defined to be

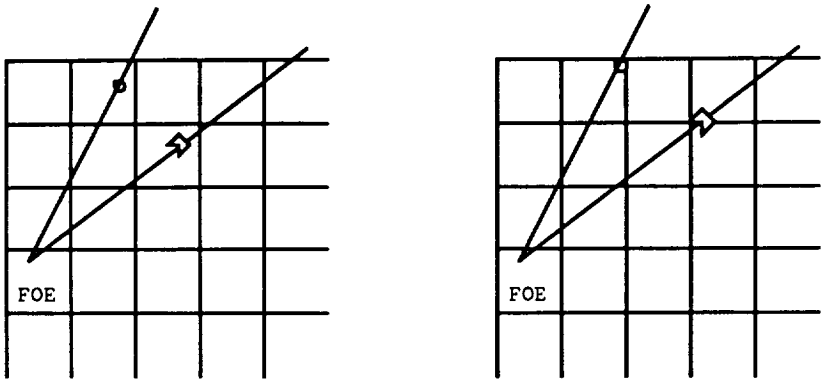

Figure 2: Evolution of features in an image sequence

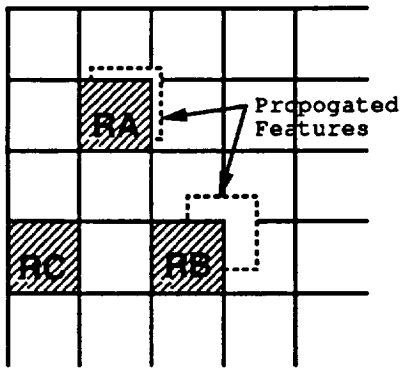

Figure 3: Addition of new features

a region which has a high variance. For our tests, we used regions of 9×9 and 11×11 pixels. The variance threshold was selected as 5 percentage variation in intensity.

3. The initial image is partitioned into $n_p \times n_p$ regions from which features are identified. Thereafter, at each subsequent frame, features from the previous frame are propagated to identify corresponding features in the current frame, possibly shifted in location. The evolution of features in a small segment of the image is shown in Figure 2. In addition, the current image is again partitioned into $n_p \times n_p$ regions to identify newly formed features. Figure 3 illustrates the procedure for adding new features. A region of high variance is picked as a new feature only if a major portion of it is not already covered by an old feature. This is done to keep the number of features to a minimum and is reasonable since not much additional information is gained by tracking completely overlapping features. A 50 percentage threshold is used in the tests. RA, RB, and RC are regions of high variance which qualify as features. However, only RB and RC are chosen as new features. At each frame the total number of features is equal to the features corresponding to the previous frame plus the new features. The total number of features varies from frame to frame and is bounded by the number of $n_p \times n_p$ regions in the image.

4. The feature in image I_k is matched to its corresponding feature in the image I_{k+1} using the normalized correlation coefficient. The normalized correlation , η_c, between two $n_p \times n_p$ regions $A(i,j)$ and $B(i,j)$ is defined by the relation:

$$\eta_c = \frac{n_p^2 \mu_{ab} - \mu_a \mu_b}{\sqrt{n_p^2 \sigma_a - \mu_a^2}\sqrt{n_p^2 \sigma_b - \mu_b^2}} \tag{8}$$

where

$$\mu_a = [\sum_{i=1}^{n_p}\sum_{j=1}^{n_p} A(i,j)] \tag{9}$$

$$\mu_b = [\sum_{i=1}^{n_p}\sum_{j=1}^{n_p} B(i,j)]$$

$$\mu_{ab} = [\sum_{i=1}^{n_p}\sum_{j=1}^{n_p} A(i,j)B(i,j)]$$

$$\sigma_a = [\sum_{i=1}^{n_p}\sum_{j=1}^{n_p} A^2(i,j)] \tag{10}$$

$$\sigma_b = [\sum_{i=1}^{n_p}\sum_{j=1}^{n_p} B^2(i,j)]$$

A value of 1 signifies a perfect match between the regions.

5. A feature in image I_{k+1} corresponding to a given feature S_p is determined by searching the image I_{k+1} for the region which has the highest normalized correlation with S_p. Let $u(k+1), v(k+1)$ be the coordinates of the center of this region and let $u(k), v(k)$ be the corresponding quantities for S_p. $[u(k+1) - u(k), v(k+1) - v(k)]$ is a measure of the optical flow or disparity associated with S_p.

6. The search area in image I_{k+1} for finding correspondence to S_p is limited to a narrow elliptic envelope, as shown in Figure 4, to reduce the amount of computation. The envelope depends on the sensor characteristics, rotorcraft motion, and the error covariance of the range estimates. As will be described in the next section, the predicted location of the feature S_p, $[\hat{u}_p(k+1), \hat{v}_p(k+1)]$, is provided by the recursive range estimation algorithm. $[\hat{u}_p(k+1), \hat{v}_p(k+1)]$ defines the center of the ellipse. The error covariance associated with $\hat{z}_s(k+1)$ is used to define the major axis of the ellipse. However, the search area is always large enough to accommodate a 30 percent error in $\hat{z}_s(k+1)$. The search along the major axis accounts for errors only in $\hat{z}_s(k+1)$. The search area is expanded to a narrow ellipse to account for errors in the location of $u(k), v(k)$. The width of the search area is a compromise between the amount of computation for the search and the likelihood of missing a correct match. The minor axis of the ellipse is chosen to be 20 percent of the major axis for the results reported in this paper.

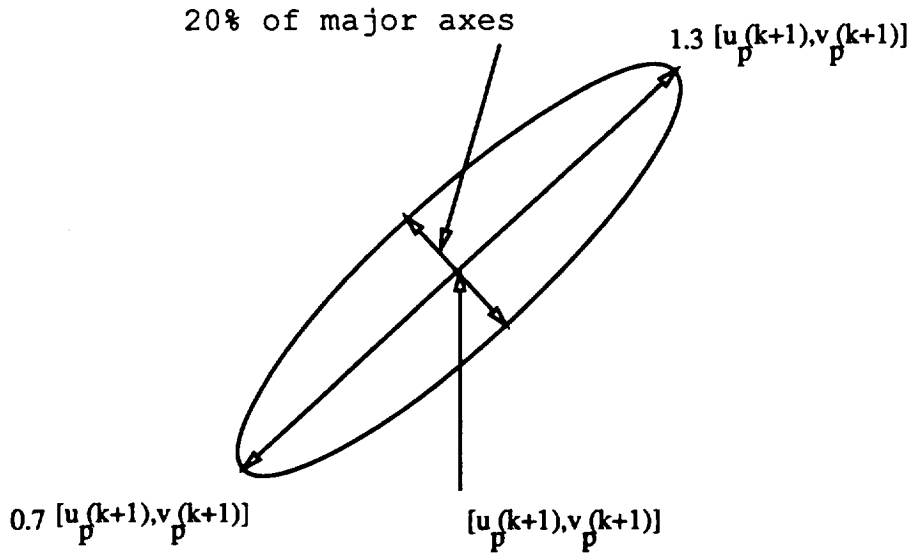

Figure 4: Search area for feature matching

7. The search step results in the location of $[u(k+1), v(k+1)]$ to pixel accuracy. Let (I1, J1) be the corresponding pixel location. We assume that η_c has a qudratic variation in the 8-pixel neighborhood of (I1, J1) and achieve sub-pixel accuracy by finding the maximum of η_c in this nighborhood.

8. The result of the previous step is a sequence of disparity vectors or image track for each feature S_p.

We have used a search procedure to find the maximum of the normalized corelation coefficient. The search procedure can be replaced by a number of techniques which optimize a function of two variables. The classical Newton method is simple and well known but subject to an array of difficulties in practice affecting reliability. Several variations of the Newton's method are available to overcome these difficulties[26]. In general, modified Newton's methods provide a faster solution than the search procedure. However, this improvement may not be significant for the maximization of the normalized coefficient as a function of two variables.

The next section will describe how the image track can be used recursively to estimate range corresponding to every feature in the image.

4 Recursive Range Estimation

Recently, recursive estimation and Kalman filters have been used by several authors to address problems in computer vision. Broida and Chellappa [27] have applied an EKF to estimate translational and rotational motion of a 2-dimensional rigid body. Wu [5] describes an EKF to estimate the positions and orientations of a rigid body by using a sequence of

images. Matthies [8] described a Kalman filter-based algorithm to estimate range using a sequence of images. However, this algorithm is limited to linear motions of the camera. The EKF has also been used to estimate the calibration parameters of the camera [28]. Wunsche [29] describes a Kalman filter-based method for controlling a mobile robot by tracking three features. Three different Kalman filters for recursive range estimation were presented in [12] and their performance was evaluated using simulated image tracks. This approach was applied to a sequence of images for linear camera motion in [30]. Here, we present the algorithms and results for both linear and curvilinear motions of the sensor. The range estimation algorithm is described for one feature to keep the notation simple However, the same analysis holds for all fixed features on the ground.

The object location estimation problem may be formulated as follows. Let a point object O have earth coordinates $r = (x, y, z)^T$. The image point corresponding to this object point has coordinates $(u, v)^T$, where u and v are given by equation (4). The actual image point location will be different from the true value $(u, v)^T$ due to noise in the sensor and errors introduced by the optical flow. Let $(u_m, v_m)^T$ be the measured coordinates of the image point, such that

$$\begin{aligned} u_m(t) &= u(t) + n_u(t) \\ v_m(t) &= v(t) + n_v(t) \end{aligned} \tag{11}$$

where n_u and n_v represent "pixel" noise of the imaging system. n_u and n_v are assumed to be independent scalar white noise processes with standard deviations σ_u and σ_v, respectively. In vector notation, measured or actual image point coordinates can be represented as

$$Z(t) = h(t) + \zeta_z(t) \tag{12}$$

where

$$\begin{aligned} h(t) &= (u, v)^T \\ \zeta_z(t) &= (n_u, n_v)^T \end{aligned}$$

and

$$R \equiv cov(\zeta_z) \equiv \begin{bmatrix} {\sigma_u}^2 & 0 \\ 0 & {\sigma_v}^2 \end{bmatrix} \tag{13}$$

The measured image point coordinates will move in the image plane as the rotorcraft flies a given trajectory. Given, estimates of rotorcraft position and velocity (translational and rotational) along it's trajectory, image point measurements from successive image frames may be used to build a Kalman filter for recursively estimating the object point coordinates in earth axes (r) and in sensor axes (ρ_s). Because the measurements, Z, are nonlinear functions of the object point coordinates, r or ρ_s, an extended Kalman filter must be used. The Kalman filters investigated in this paper have a linear continuous state model of the form

$$\dot{X} = F(t)X(t) + G(t)U(t) + \zeta_x(t) \tag{14}$$

where X is the state vector U is the control input, $\zeta_x(t)$ is a continuous white noise with covariance Q_c (representing modeling uncertainty), and $F(t)$ and $G(t)$ are time-varying matrices. Using a sampling interval of ΔT seconds, equation (14) can be replaced by the discrete form

$$X(k+1) = \Phi(k)X(k) + \Gamma(k)U(k) + \zeta_x(k) \tag{15}$$

where $k \equiv i.\Delta T$, $k+1 \equiv (i+1)\Delta T$, $i = 1, 2, 3...$, $\Phi(k)$ is the state transition matrix and $\Gamma(k)$ is the input distribution matrix. The process noise $\zeta_x(k)$ is used to model uncertainties in the knowledge of V_s and $[\omega_s]$. $\zeta_x(k)$ is a discrete white noise sequence with covariance $Q = Q_c/\Delta T$. The measurements $Z(t)$ are assumed to be available every ΔT_m seconds where $\Delta T_m = M\Delta T$, M being a constant positive integer. The measurements $Z(t)$ are non-linearly related to the state through the vector function $h(X(t))$ and can be linearized to give the measurement equation. Thus, whenever i is an integral multiple of M , we have

$$Z(k) = h[X(k)] + \zeta_z(k) \tag{16}$$

Given the state equations (15) and the image point measurements $Z(k)$ of equation (16) the state estimate $\hat{X}(k)$ and it's error covariance matrix P can be computed recursively using the Kalman filter [31].

The Kalman filter consists of two parts:

1. Measurement Update: The measurement update is done whenever a new measurement is available, i.e., i is an integral multiple of M. Prior to processing a new measurement $Z(k)$, we have the estimated value of the state $\hat{X}$ and the covariances $P(k), Q(k)$ and $R(k)$. The new measurement improves our estimate of the state and it's covariance. The updated values are

$$\begin{aligned} \tilde{X}(k) &= \hat{X}(k) + K(k)[Z(k) - h(X(k))] \\ \tilde{P}(k) &= [I - K(k)H(k)]P(k) \end{aligned} \tag{17}$$

where the matrix of partial derivatives

$$H(k) = \partial h(X)/\partial X \tag{18}$$

and the Kalman filter gain $K(k)$ is computed using the equation

$$K(k) = P(k)H^T(k)[H(k)P(k)H^T(k) + R(k)]^{-1} \tag{19}$$

When i is not a multiple of M,

$$\tilde{X}(k) = \hat{X}(k) \tag{20}$$

$$\tilde{P}(k) = P(k) \tag{21}$$

2. Time Update: This part of the filter accounts for the system dynamics and propagates the state and its covariance matrix until the next measurement is made. The propagated values are:

$$\begin{aligned} \hat{X}(k+1) &= \Phi(k)\tilde{X}(k) + \Gamma(k)U(k) \\ \hat{P}(k+1) &= \Phi(k)\tilde{P}(k)\Phi(k)^T + \Gamma(k)Q(k)\Gamma(k)^T \end{aligned} \tag{22}$$

The actual implementation of the Kalman Filter is done by updating the measurements one at a time. This results in a Kalman filter gain different from $K(k)$. However, the final $\tilde{X}$ and $\tilde{P}$ at the end of the measurement update is the same as before. Due to the excellent numerical properties of the factorized form [32], the error covariance propagation is done by decomposing the state covariance P in the form

$$P(k) = U(k)D(k)U^T(k) \tag{23}$$

where U is an upper triangular matrix and D is a diagonal matrix.

Three different Kalman filter formulations for object point location are discussed in [12]. They are the result of three different representations for the state vector in equations (14) and (15). Here, we choose the relative coordinates of the object point O with respect to the rotorcraft in sensor axes as the state vector. Thus,

$$X \equiv \rho_s \equiv (x_s, y_s, z_s)^T \tag{24}$$

From equation (3),

$$\dot{X} = -[\hat{\omega}_s]X - \hat{V}_s \tag{25}$$

Equation (25) defines a time-varying linear system as in equation (14) with $F(t) \equiv -[\hat{\omega}_s]$, $G(t) \equiv I$, $U(t) \equiv -\hat{V}_s$ and $\zeta_x \equiv 0$. Note that $\hat{\omega}_s$ and $\hat{V}_s$ are the estimated rotorcraft angular and translational velocity vectors in sensor coordinates provided by the on-board inertial navigation system.

The conversion of the continuous time-varying state model in equation (14) to the discrete form (15) is done assuming $F(t)$, $G(t)$ and $U(t)$ to be constant over a small interval of time ΔT. This assumption implies that the rotorcraft linear and angular velocity is constant during the time interval ΔT. This assumption can be satisfied by updating the state equations at a higher rate as is the case in most airborne inertial navigation systems. The conversion from the continuous to discrete form is usually done using numerical techniques [33]. Since F is a 3×3 matrix, by working in the frequency domain we can derive the discrete system equations analytically. This expression gives a more accurate value for $\Phi(t)$ than the numerical approximation resulting from a Taylor series expansion.

Under these conditions, Appendix shows that

$$\Phi(k) = \begin{bmatrix} \Phi_{11}(k) & \Phi_{12}(k) & \Phi_{13}(k) \\ \Phi_{21}(k) & \Phi_{22}(k) & \Phi_{31}(k) \\ \Phi_{32}(k) & \Phi_{32}(k) & \Phi_{33}(k) \end{bmatrix} \tag{26}$$

and

$$\Gamma(k) = \begin{bmatrix} \Gamma_{11}(k) & \Gamma_{12}(k) & \Gamma_{13}(k) \\ \Gamma_{21}(k) & \Gamma_{22}(k) & \Gamma_{31}(k) \\ \Gamma_{31}(k) & \Gamma_{32}(k) & \Gamma_{33}(k) \end{bmatrix} \tag{27}$$

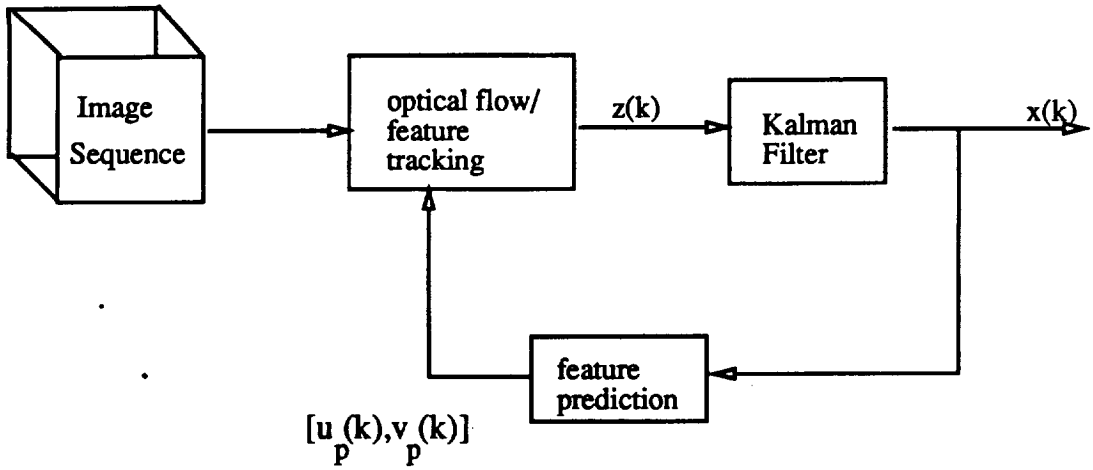

Figure 5: Interaction between computation of optical flow and range estimation

where Φ_{ij} and Γ_{ij} are defined by equations (41)-(48). Using (16), we can write

$$Z(k) = h[X(k)] + \zeta_z(k) \tag{28}$$

where

$$h[X(k)] = [h_1, h_2]^T = [fx_s/z_s, fy_s/z_s]^T \tag{29}$$

The matrix

$$H[X(k)] = \begin{bmatrix} \partial h_1/\partial x_s & \partial h_1/\partial y_s & \partial h_1/\partial z_s \\ \partial h_2/\partial x_s & \partial h_2/\partial y_s & \partial h_2/\partial z_s \end{bmatrix} \tag{30}$$

By taking the partial derivatives, the expression for H reduces to

$$H[X(k)] = H(\rho_s) = f \begin{bmatrix} 1/z_s & 0 & -x_s/z_s^2 \\ 0 & 1/z_s & -y_s/z_s^2 \end{bmatrix}_{\rho_s = \hat{\rho}_s(k-1)} \tag{31}$$

Equations (25) and (31) can be used in the extended Kalman filter to recursively estimate the state $\hat{X} = \hat{\rho}_s = [\hat{x}_s, \hat{y}_s, \hat{z}_s]^T$ and its covariance P_s. In earth coordinates, $\hat{\rho}$ and $\hat{r}$ can be computed using the equation

$$\hat{\rho} = T^T T_s^T \hat{\rho}_s$$
$$\hat{r} = \hat{\rho} + \hat{r}_h$$

Similarly, $P = T^T P_b T$ where $P_b = T_s^T P_s T_s$.

The interaction between optical flow calculations and recursive range estimation is shown in Figure 5. Assume that we want to process the image I_k to improve the range estimate corresponding to a feature $S_p(k-1)$ from image I_{k-1}. The output of the Kalman filter based on measurements $[u(k-1), v(k-1)]$ provide the estimates $\hat{X}(k) = [\hat{x}_s(k), \hat{y}_s(k), \hat{z}_s(k)]^T$. Based on the state estimate, we can predict the location of $S_p(k)$ in image k to be

$$\hat{u}_p(k) = f\hat{x}_s/\hat{z}_s$$
$$\hat{v}_p(k) = f\hat{y}_s/\hat{z}_s \tag{32}$$

The values $[\hat{u}_p(k), \hat{v}_p(k)]$ are provided to the optical flow algorithm to find the location of the corresponding feature at $[u(k), v(k)]$. Generally feature matching algorithms would require the derotation of the image I_{k+1} to account for the rotational motion of the vehicle before matching can be done between features in image I_k and image I_{k+1}. However, by definition, the EKF accounts for total helicopter motion and due to its continuing update no derotation is necessary with the matching algorithm discussed here.

The state error covariance matrix P provides a measure of the accuracy of the estimate. The magnitude of $P(k)$ decreases monotonically as new measurements $[u(k), v(k)]$ are provided to the Kalman filter by the optical flow calculations. The amount of decrease in $P(k)$ over a finite number of image sequences, N, depends on the measurement noise $R(k)$ and N. Let T_c be the amount of time required for the estimates to converge to a satisfactory value and let ΔT_m be the interval between measurements. We have $T_c = N.\Delta T_m$. Increasing ΔT_m to ΔT_1 leads to an increase in the amount of change in $[u(k), v(k)]$, and assuming the same error in the location of $[u(k+1), v(k+1)]$, results in a higher signal to noise ratio. We may need $N_1 \leq N$ images to reach the same level of convergence. The corresponding value of T_c is $T_1 = N_1.\Delta T_1$. Similarly, decreasing ΔT_m to ΔT_2 leads to a decrease in the amount of change in $[u(k), v(k)]$, and assuming the same error in the location of $[u(k+1), v(k+1)]$, results in a lower signal to noise ratio. We may need $N_2 \geq N$ images to reach the same level of convergence. The corresponding value of T_c is $T_2 = N_2.\Delta T_2$. T_c is a nonlinear function of ΔT_m and has a minimum for a specific value of ΔT_m . The variation of T_c for simulated images shows a large decrease in T_c with decreasing ΔT_m upto a certain value, followed by a small decrease in T_c with decreasing ΔT_m[12]. For small values of ΔT_m, the number of samples required before the estimates converge is very large simply because of the poor geometry associated with each successive pairs of images. For large values of ΔT_m the time required for convergence may be prohibitive due to ΔT_m itself. The lower bound on ΔT_m depends on the amount of time required for low-level image processing. The choice of ΔT_m depends on the change in $[u(k), v(k)]$ in successive images. As can be seen from equations (5) and (6), optical flow depends on the rotorcraft motion relative to the object. Larger values of ΔT_m are needed for low rotorcraft speed in order to maintain the same signal to noise ratio in measurements. For the same vehicle maneuver, range estimates to objects far away from the FOE will converge faster than objects closer to the FOE. This suggests a strategy where the features are tracked at a high sampling rate as dictated by the minimum value of ΔT_m; but, designing the Kalman filter to employ a variable measurement update interval which depends on the magnitude of the displacement of the feature within the image. We are in the process of designing a event driven Kalman filter where a new measurement is accepted for update only if it passes a signal to noise threshold. This would result in a uniform signal to noise ratio over a large portion of the FOV. Our evaluation will focus on three different values for ΔT_m.

5 Performance Evaluation

The evaluation of algorithm performance has not received a lot of attention in computer vision reasearch. However, it is essential that algorithms be tested extensively before the technology can be transferred to applications. One reason for the lack of detailed evaluation is the absence of standard data sets of images. Recently, efforts are being made to collect a standard data set of images for testing algorithms. Another difficulty in the evaluation of algorithms is the need for an evaluation criterion. Visualization of the enormous amounts of information poses another challenge in the debugging and display of results. These issues have been recognized by researchers in computer vision and are the subject of several discussions [34, 35, 36]. We will describe the steps followed in the evaluation of the range estimation algorithm and relate our experience to problems and issues in the evaluation of computer vision algorithms.

The evaluation of rotorcraft obstacle detection algorithms needs to be done using both laboratory image sequences and imagery data acquired during rotorcraft flight. NASA Ames has developed a set of flight data [37]. In the laboratory an experiment has been designed to acquire a sequence of images by mounting a camera on a 3 degree-of-freedom motion table. The camera position and orientation is controlled by a computer to achieve desired curvilinear camera motion. The details about the laboratory setup can be found in [38]. The camera can be moved at different speeds. Figure 6 shows the laboratory setup and the location of various objects. A sequence of 80 images was collected by moving the camera along a straight line during the segment AB followed by a cosine curve during the segment BC. Figure 7 shows the details of the camera path. The camera is stationary when the image is captured and is moved to the next position before taking the next image. The camera moves a distance of 0.125in along the direction AB between steps. The optical axis of the camera is tangential to the path ABC. The camera motion parameters were computed from the path ABC and provided as inputs to equation (25). This experiment can be scaled to simulate a helicopter flying at 20 knots where the objects in the FOV vary from 50m to 500m and the images are being processed for range computation every 0.25 seconds.

The position of the camera when the 36th, 60th, and 76th image is captured is referred to as camera position 1, 2, and 3 respectively in subsequent discussion. These positions are indicated in Figure 7. Figures 8, 9, 10 and 11 show the zeroth, the 36th, the 60th and the 76th image in the sequence of 80 images. Five different regions are highlighted in the images. In Figure 9, these correspond to the edge of a bracket at 15.2 inches from the camera, a region which includes the FOE on a soda can about 22.7 inches from the camera, a pencil to the right of FOE and located at about 19.5 inches, another pencil to the left of FOE and located at about 24.5 inches, and a tape located at 56 inches from the camera. These regions were selected to examine the behavior of the algorithms at different ranges and at different angles with respect to the direction of motion of the sensor.

It is difficult to state a single measure for the evaluation of range estimation algorithms.

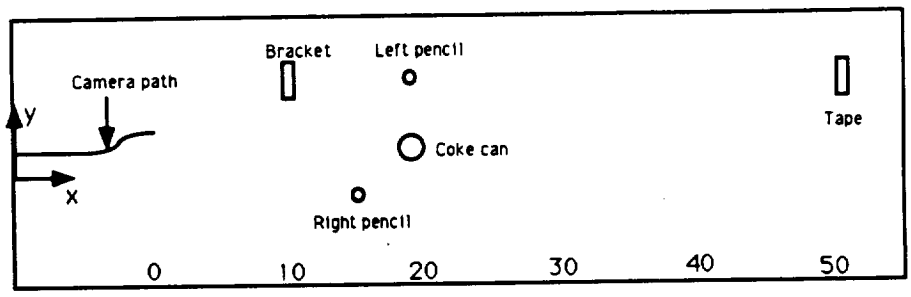

Figure 6: Laboratory setup and location of objects

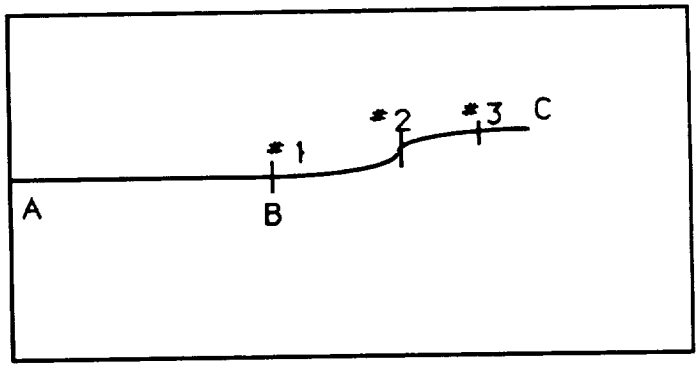

Figure 7: Details of camera path

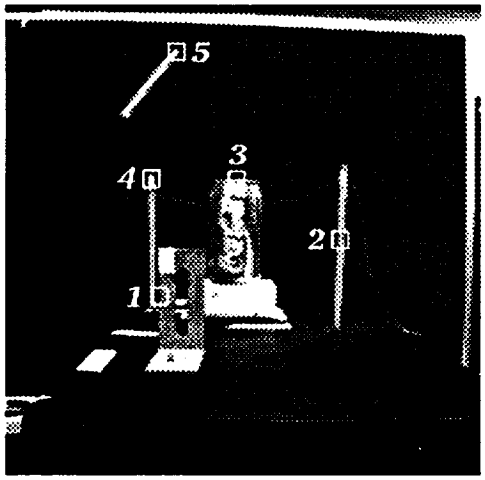

Figure 8: Zeroth image

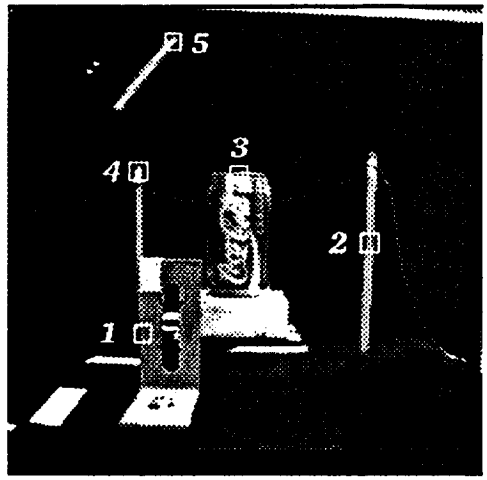

Figure 9: 36th image

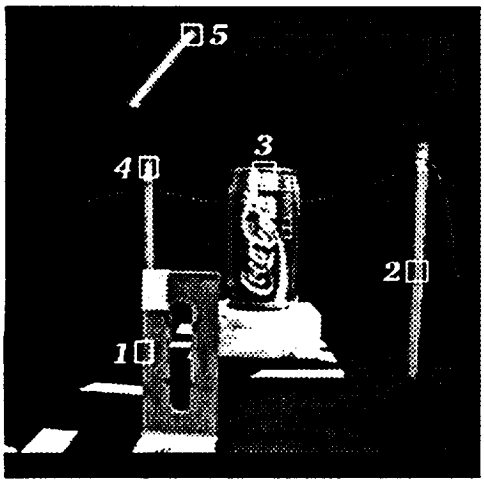

Figure 10: 60th image

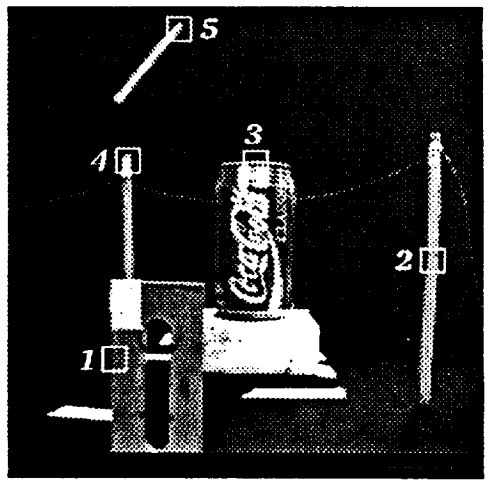

Figure 11: 76th image

If the true terrain data is available, an obvious measure is the accuracy of the estimate at each pixel location. The usefulness of the range information depends on the accuracy at different locations in the field-of-view (FOV) as well as regions in the image where range measurements are available. The desired accuracy and the critical regions in the FOV will vary with the obstacle avoidance algorithm and its implementation. For regions in the FOV where no true terrain data is available, the covariance of the range estimate can be used as a measure of the quality of the estimate. The accuracy of Kalman filter-based range estimates depends on the error in individual measurements at each frame and the number of frames used to estimate the range. The convergence of a Kalman filter depends on several different factors and we will explore the relationship between accuracy, time interval between images and the number of images.

We adopted the following procedure in the evaluation of the algorithms:

1. Compute range to all tracked features in the image and store their range and covariance as a function of the identification number of the feature and image sequence.

2. The range estimates at various points in the FOV are color-coded and superimposed on the original image. Since different features corresponding to the same object should have approximately the same range, the color-coding gives a global picture of the range algorithm performance.

3. Manually select regions in the image which correspond to the same object and for which truth data is available

4. Compute the mean and standard deviation of the estimated range corresponding to all the features in a given region

5. Examine the performance of the algorithm as a function of sampling time and the number of samples used to estimate the range.

Next, we present range estimation results using the laboratory data following the procedure outlined above. Features were detected in these images using algorithms described in section 3. The number of features in the image sequence varied from 600 to 1000. A physical object such as a pencil in the FOV gives rise to several features and their associated image tracks. Figure 12 shows the image tracks generated by four features corresponding to different parts of the right pencil. The feature tracks are of varying length since new features are created and some old features may either move out of the FOV or cannot be tracked when a new image is acquired for processing. Table 1 shows the identification number associated with the some of the features corresponding to the right pencil, the respective image plane coordinates and the axial component of range estimate ($\hat{z}_s$) at the 80th image sequence. The identification number is used as a link to associate all the information and display relating to a single feature. Figure 13 shows the variations in the axial component (z_s) of the edge corresponding to the right pencil as a function of the number of iterations. The object position converges to the true value as the number of iterations increase. All the features corresponding to the same object will exhibit similar convergence and the plot can be used

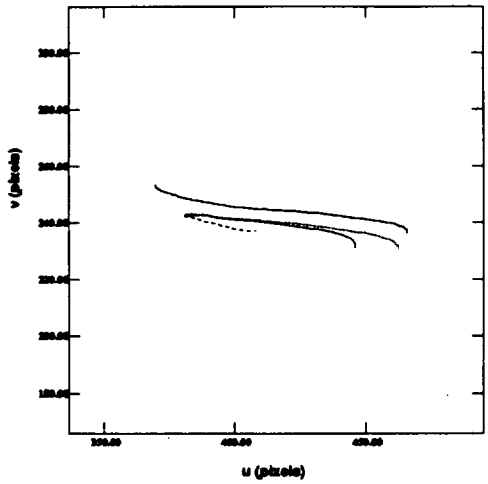

Figure 12: Image tracks generated by 4 features corresponding to the right pencil

to identify feature tracks which do not correspond to a real object on the ground. Figure 14 shows a histogram of the estimated range using features corresponding to the right pencil. This histogram was computed at the end of the 76th image and has 98 features. Recall that the camera is moving along a straight line when the first 40 images were acquired and the camera has both translational and rotational velocity during the time when the next 40 images are acquired. The time interval between images is fixed. This image sequence was processed in three different ways. Data set F uses all the 80 images. Data set H uses alternate images from data set F and has 40 images. Data set Q is a subset of data set H and is made up of 20 alternate images. Figure 15 illustrates the relationship between the three data sets. The sensor motion along the line AB between two images in data sets F, H, and Q is 0.125in, 0.25in, and 0.5in respectively. The three data sets provide us with three different measurement update intervals_ slow, moderate and fast_ in the estimation of range.

We examine the range estimates using datasets F, H and Q respectively in the region corresponding to the Bracket. Figure 16.a, 16.b and 16.c show range estimates as a function of frame number using datasets F, H and Q respectively. It should be recalled that the number of range updates is equal to frame number/M where M takes the values 1, 2 and 4 for the datasets F, H and Q respectively. The dotted lines in the figure show the 5% estimation error boundaries. The estimation error converges to within 5% of the true value faster in dataset F. This is due to the fact that the disparity between images is significant in the dataset F in the region corresponding to the Bracket. At frame number 20, the range has been updated 20, 10 and 5 times in Figure 16.a, 16.b and 16.c respectively. Figure 17.a, 17.b and 17.c shows similar results for the Right Pencil. The estimation error performance of the Right Pencil is similar to the Bracket. Figures 18.a, 18.b and 18.c show the range estimation results in the region corresponding to the can. The can represents a region in the image very close to the FOE. The range estimation results are not very reliable in this region. However, as can be seen from Figure 18.c the range estimation errors are smaller

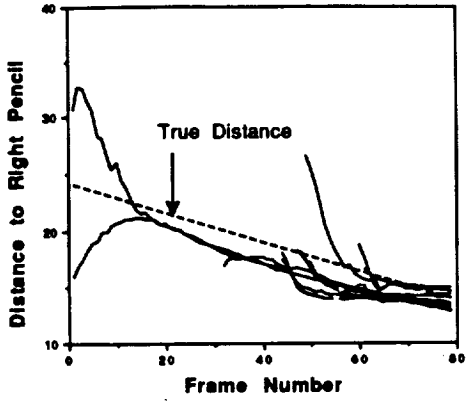

Figure 13: Variation in range with number of iterations (right pencil)

Table 1: Identification number and range

id	u (pixels)	v (pixels)	range (in.)
143	465.77	274.98	13.43
1357	462.59	280.16	12.54
1643	448.16	278.32	13.45
1931	445.84	279.88	12.50
2064	449.91	263.81	13.36
2110	444.65	267.51	13.72
2162	445.89	264.24	12.85
2165	459.54	279.21	13.09
2280	448.78	276.31	13.85
2872	456.10	262.25	17.26
3371	457.67	271.29	14.82
3419	456.46	271.13	15.78
3535	444.30	271.10	15.09
3637	456.21	270.35	14.69
3788	444.42	268.47	15.51

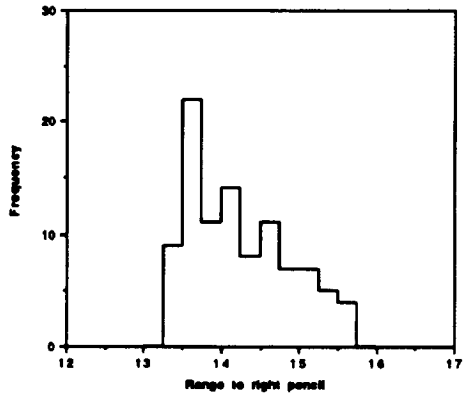

Figure 14: Histogram of range estimates (Right pencil, data set F, 76th image)

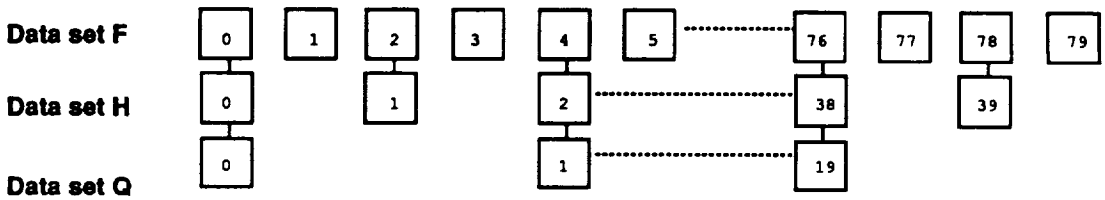

Figure 15: Relationship between data sets F, H, and Q

for dataset Q than the other two sets. Figure 19 shows the range estimation results in the region corresponding to the Left Pencil. The range estimation results agree well with the true values. The estimation errors are smallest for the moderate update interval.

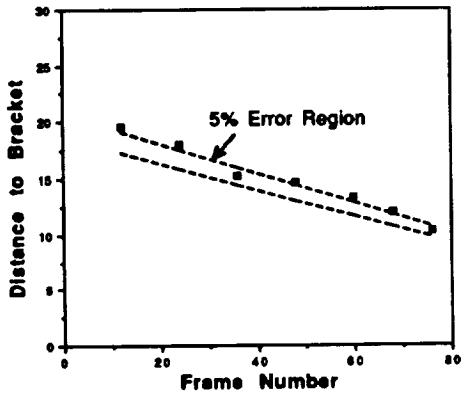

Figure 16: (a) Range Estimation in the Bracket Region (Dataset F)

Figures 20.a, 20.b and 20.c show the range estimation results in the region corresponding to the Tape. The region surrounding the tape was selected to examine the behavior of the algorithm at distances far away from the sensor. The changes in disparity of a feature corresponding to the tape using data set F, data set H and data set Q are shown in Figure 21. The estimation errors are smallest for moderate update intervals. The reason for the poor performance of the algorithm using the small ΔT_m is the disparity between images is very small. This results in a very low signal to noise ratio in the measurements resulting from the optical flow algorithm using data set F. At large update intervals, the number of updates for a given time is small.

The above results indicate that it is possible to estimate range to an accuracy of 5 to 10 percentage. In problems involving large variations of optical flow within the same image, the selection of the measurement update interval affects the estimation accuracy. This problem can be addressed by a variable update Kalman filter where the update rate is chosen depending on the disparity at a feature location. The main source of error in z is due to errors in camera calibration and knowledge of the camera co-ordinate system geometry with respect to the world-axis located on the motion table. The errors can be

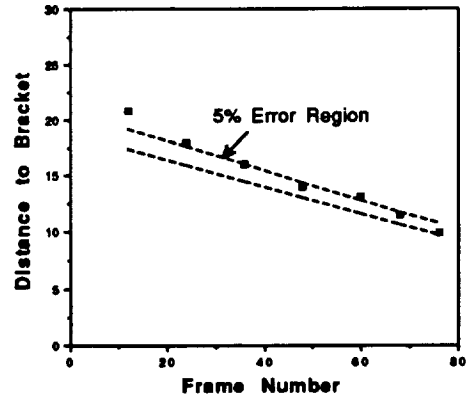

Figure 16: (b) Range Estimation in the Bracket Region (Dataset H)

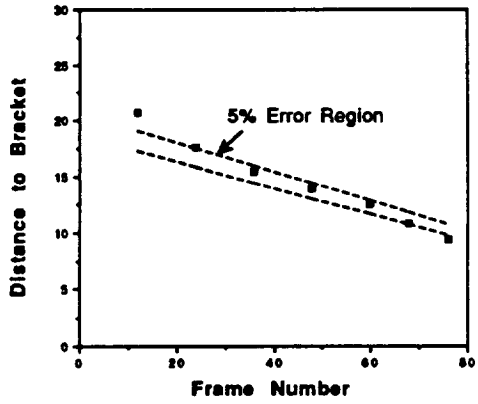

Figure 16: (c) Range Estimation in the Bracket Region (Dataset Q)

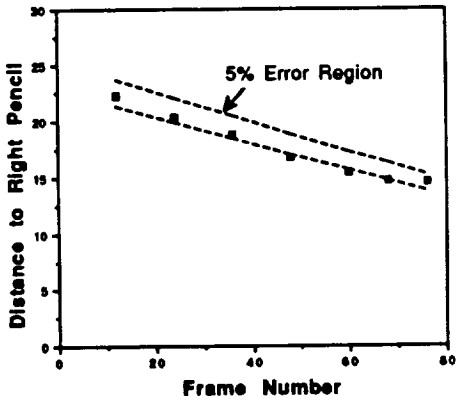

Figure 17: (a) Range Estimation in the Right Pencil Region (Dataset F)

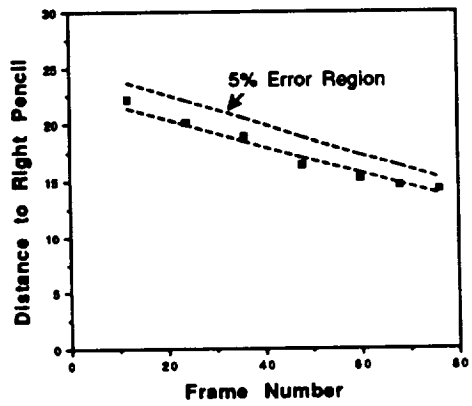

Figure 17: (b) Range Estimation in the Right Pencil Region (Dataset H)

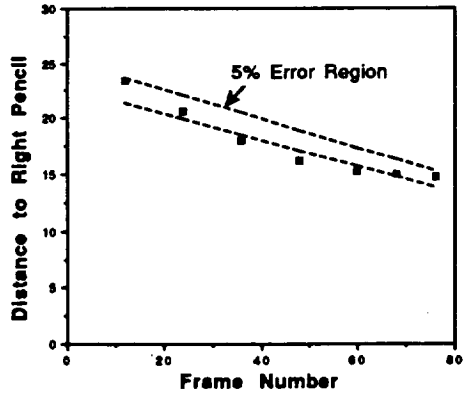

Figure 17: (c) Range Estimation in the Right Pencil Region (Dataset Q)

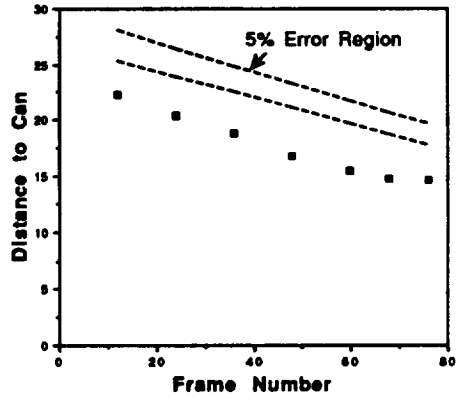

Figure 18: (a) Range Estimation in the Can Region (Dataset F)

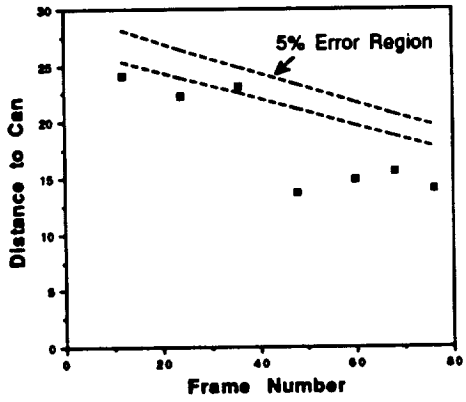

Figure 18: (b) Range Estimation in the Can Region (Dataset H)

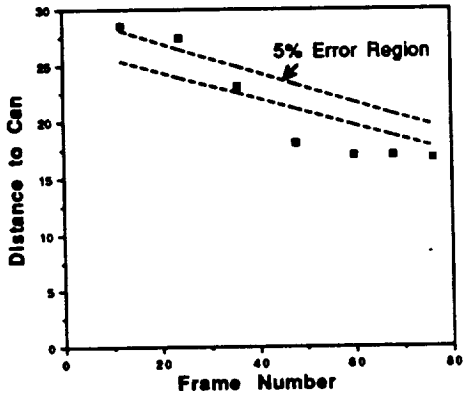

Figure 18: (c) Range Estimation in the Can Region (Dataset Q)

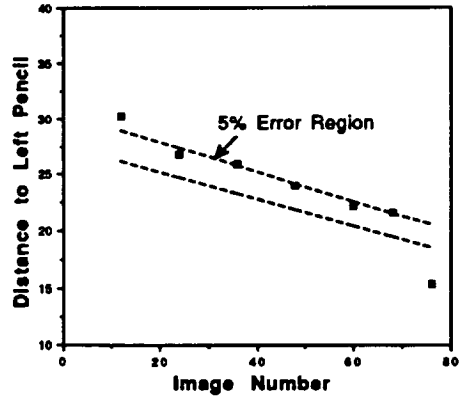

Figure 19: Range Estimation in the Left Pencil Region (Dataset H)

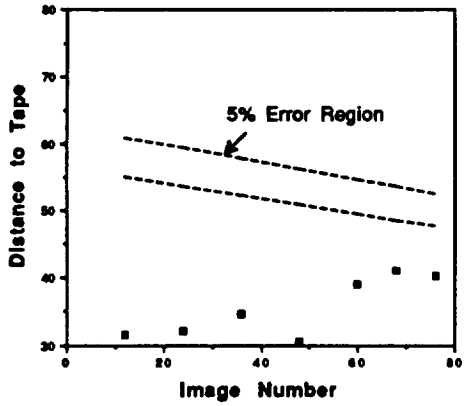

Figure 20: (a) Range Estimation in the Tape Region (Dataset F)

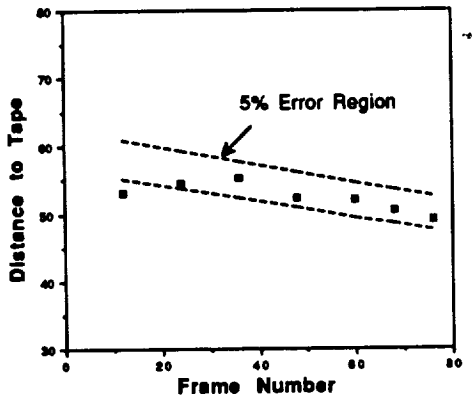

Figure 20: (b) Range Estimation in the Tape Region (Dataset H)

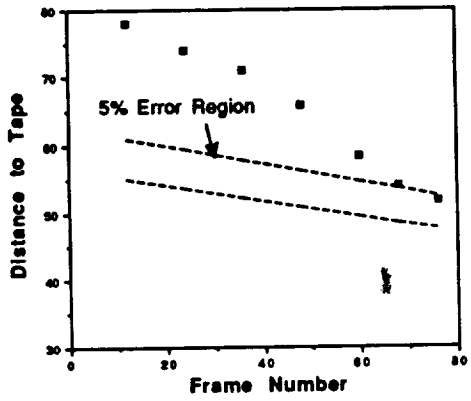

Figure 20: (c) Range Estimation in the Tape Region (Dataset Q)

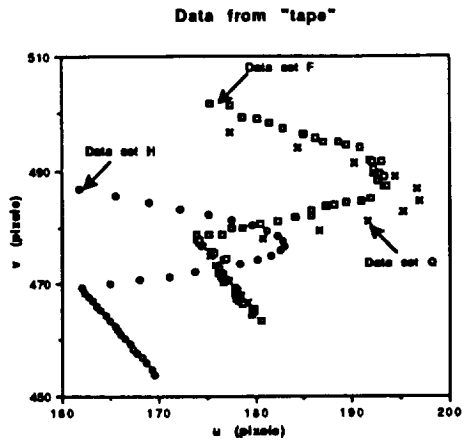

Figure 21: Motion in the image plane between images using data sets F, H, and Q

further reduced by camera calibration which will reduce the uncertainty in the location of the camera co-ordinate system with respect to the motion table.

6 Summary and Future Work

We have presented a recursive method to estimate range to objects using a sequence of images. The components of our algorithm were based on the characteristics of low-altitude rotorcraft flight. They are (i) natural scenarios, (ii) curvilinaer motion, (iii) large variation in the dynamic range of the optical flow, and (iv) the availability of sensor motion parameters. These characteristics made it impossible to directly use several algorithms reported in the computer vision literature. The method was evaluated extensively using real images in a laboratory setup and produces good range estimates. The performance of the method can be improved further by camera calibration to remove inaccuracies in the determination of the image plane. Recently, we have tested the algorithms on several scenarios containing images from a CH-47 helicopter flight. The algorithm performs very well and the results will be reported in a forthcoming paper.

We are considering the use of a Kalman filter with variable time intervals between estimate updates to deal with the wide dynamic range of optical flow in an image. Like most ranging algorithms depending on motion, the performance of the algorithm is poor close to the FOE. This problem is being addressed by the use of stereo and by integrating stereo and motion in a Kalman filter formulation. The method needs further evaluation using several different image sequences to test its robustness.

So far, we have not addressed the real-time implementation issues. We plan to parallelize our algorithms for implementation on a parallel machine such as Intel's iWARP. Alternatively, we are exploring ways to reduce the number of features per object by supplementing

feature tracking with knowledge about contiguous objects/surfaces.

Feature detection methods provide a sparse range map in the FOV. They have to be supplemented by intelligent algorithms such as context dependent scene analysis [39] to fill the gaps in the range maps. These range maps may have to be further interrogated by an active system before the range information can be used by the obstacle avoidance system.

The research reported in this paper is part of an on-going effort at NASA Ames to develop technologies for the automation of rotorcraft low altitude flight. The object detection and range estimation algorithms discussed are quite general and have potential applications in robotics and autonomous navigation of vehicles. The laboratory and flight image sequences together with the optical flow and recursive estimation software can be requested from NASA/Ames. In addition to these feature-based algorithms, there are parallel efforts to investigate field-based techniques for the same range estimation applications [19, 20].

7 Appendix

This appendix derives analytical expressions for the discrete equivalent of the continuous system in equation (25). Assume the linear and angular velocity of the sensor (V_s, ω_s) to be constant over a small interval of time ΔT. For a time-invariant matrix F, the state transition matrix is given by the equation

$$\Phi(t) = \exp(Ft) = I + Ft + \frac{1}{2!}F^2t^2 + \frac{1}{3!}F^3t^3 + ... \tag{33}$$

$\Phi(t)$ is evaluated numerically [33]. Because F is a 3×3 matrix, we can derive an expression for $\Phi(t)$ by working in the frequency domain. We have,

$$\Phi = \mathcal{L}^{-1}[sI - F]^{-1} \tag{34}$$

where s is the Laplace transform variable and $\mathcal{L}^{-1}$ is the inverse Laplace transform. In the present case,

$$F = [\omega_s(t)]|_{t=l*\Delta T} \tag{35}$$

We have

$$[sI - F] = \begin{bmatrix} s & -\omega_{sz} & \omega_{sy} \\ \omega_{sz} & s & -\omega_{sx} \\ -\omega_{sy} & \omega_{sx} & s \end{bmatrix} \tag{36}$$

Further,

$$[sI - F]^{-1} = Adj[sI - F]/det[sI - F] \tag{37}$$

where I is the 3×3 identity matrix,

$$det[sI - F] = s(s^2 + \alpha^2) \tag{38}$$

and

$$\alpha^2 = \omega_{sx}^2 + \omega_{sy}^2 + \omega_{sz}^2 \tag{39}$$

The adjoint matrix is given by the equation

$$Adj[sI - F] = \begin{bmatrix} s^2 + \omega_{sx}^2 & -s\omega_{sz} + \omega_{sx}\omega_{sy} & s\omega_{sy} + \omega_{sz}\omega_{sx} \\ s\omega_{sz} + \omega_{sx}\omega_{sy} & s^2 + \omega_{sy}^2 & -s\omega_{sx} + \omega_{sy}\omega_{sz} \\ -s\omega_{sy} + \omega_{sz}\omega_{sx} & s\omega_{sx} + \omega_{sy}\omega_{sz} & s^2 + \omega_{sz}^2 \end{bmatrix} \tag{40}$$

Let Φ_{ij} be the ijth element of the 3×3 state transition matrix. Φ_{ij} is the Laplace inverse of the ijth element of the matrix $[sI - F]^{-1}$. Then, using equations (37), (38) and (40), we have

$$\Phi_{11} = \mathcal{L}^{-1} \frac{s^2 + \omega_{sx}^2}{s(s^2 + \omega_{sx}^2)} = a\omega_{sx}^2 + cos(\alpha\Delta T) \tag{41}$$

where

$$a = \frac{1 - cos(\alpha\Delta T)}{\alpha^2} \tag{42}$$

Similarly,

$$\begin{aligned} \Phi_{12} &= a\omega_{sx}\omega_{sy} + b\omega_{sz} \\ \Phi_{13} &= a\omega_{sz}\omega_{sx} - b\omega_{sy} \\ \Phi_{21} &= a\omega_{sx}\omega_{sy} - b\omega_{sz} \\ \Phi_{22} &= a\omega_{sy}^2 + cos(\alpha\Delta T) \\ \Phi_{23} &= a\omega_{sy}\omega_{sz} + b\omega_{sx} \\ \Phi_{31} &= a\omega_{sz}\omega_{sx} + b\omega_{sy} \\ \Phi_{32} &= a\omega_{sy}\omega_{sz} - b\omega_{sx} \\ \Phi_{33} &= a\omega_{sz}^2 + cos(\alpha\Delta T) \end{aligned} \tag{43}$$

where

$$b = \frac{sin(\alpha\Delta T)}{\alpha} \tag{44}$$

Assuming the input $U(t) = V_s(t)$ to be constant during the interval, the input distribution matrix Γ in equation (15) can be computed using the equation

$$\Gamma(l) = \int_0^{\Delta T} \Phi(t)G(t)dt \tag{45}$$

Since $G(t) = I$,

$$\Gamma(l) = \int_0^{\Delta T} \Phi(t)dt \tag{46}$$

Using the values for $\Phi(t)$ from equations (41) and (44) in equation (46), and defining

$$c = \frac{\Delta T}{\alpha^2} - \frac{sin(\alpha\Delta T)}{\alpha^3} \tag{47}$$

we have

$$
\begin{aligned}
\Gamma_{11} &= c\omega_{sx}^2 + sin(\alpha\Delta T)/\alpha \\
\Gamma_{12} &= c\omega_{sx}\omega_{sy} + a\omega_{sz} \\
\Gamma_{13} &= c\omega_{sz}\omega_{sx} - a\omega_{sy} \\
\Gamma_{21} &= c\omega_{sx}\omega_{sy} - a\omega_{sz} \\
\Gamma_{22} &= c\omega_{sy}^2 + sin(\alpha\Delta T)/\alpha \\
\Gamma_{23} &= c\omega_{sy}\omega_{sz} - a\omega_{sx} \\
\Gamma_{31} &= c\omega_{sz}\omega_{sx} + a\omega_{sy} \\
\Gamma_{32} &= c\omega_{sy}\omega_{sz} - a\omega_{sx} \\
\Gamma_{33} &= c\omega_{sz}^2 + sin(\alpha\Delta T)/\alpha
\end{aligned}
\tag{48}
$$

References

[1] V.H.L. Cheng and B. Sridhar. Considerations for automated nap-of-the-earth flight. In *Proceedings of ACC*, Atlanta, GA, June 1988.

[2] V.H.L. Cheng and B. Sridhar. Integration of active and passive sensors for obstacle avoidance. *IEEE Control Systems Magazine*, 10(4):43–50, June 1990.

[3] J.K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images- a review. *Proceedings of the IEEE*, 76(8):917–935, 1988.

[4] U.R. Dhond and J.K. Aggarwal. Structure from stereo from sequences of images- a review. *IEEE Trans. on Systems, Man and Cybernetics*, 19(6):1489–1510, 1989.

[5] J.J. Wu, R.E. Rink, T.M. Caelli, and V.G. Gourishankar. Recovery of the 3-d location and motion of a rigid object through camera image (an extended kalman filter approach). *International Journal of Computer Vision*, 3(1):373–394, 1988.

[6] R.Y. Tsai and T.S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surface. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(1):13–26, 1984.

[7] Tsukiyama and T.S. Huang. Segment-based stereo matching. *Pattern Recognition*, 31(1):2–18, 1987.

[8] L. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. Technical Memorandum CMU-RI-TR-88-1, Carnegie Mellon University, Carnegie Mellon University, Pittsburgh, PA 15213, January 1988.

[9] G. Xu, S. Tsuji, and M. Asada. A motion stereo method based on coarse-to-fine control strategy. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(3):332–336, 1987.

[10] E. Krotkov. Focusing. *International Journal of Computer Vision*, 1:223–237, 1987.

[11] J. Ma and S.I. Olsen. Depth from zooming. *JOSA*, 7(10):1883–1890, 1990.

[12] B. Sridhar and A.V. Phatak. Simulation and analysis of image-based navigation system for rotorcraft low-altitude flight. In *Proceedings of the AHS Meeting on Automation Application for Rotorcraft*, Atlanta, GA, April 1988.

[13] H. Goldstein. *Classical Mechanics.* Addison-Wesley, Reading, MA, 1965.

[14] D.H. Ballard and C.M. Brown. *Computer Vision.* Prentice-Hall, Englewood Cliffs, N.J, 1982.

[15] S.T. Barnard and W.B. Thompson. Disparity analysis of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2(4):333–340, 1980.

[16] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(3):185–203, August 1981.

[17] B.K.P. Horn. *Robot Vision.* M.I.T. Press, Cambridge, MA, 1986.

[18] T. Pogio. Mit progress in understanding images. In *DARPA Image Understanding Workshop*, Los Angeles, CA, February 1987.

[19] P.K. Menon and B. Sridhar. Passive navigation using image irradiance tracking. In *AIAA Guidance, Navigation and Control Conference*, Boston, MA, August 1989.

[20] Y. Barniv. Velocity filtering applied to optical flow calculations. Technical Memorandum 102802, NASA, Ames Research Center, Moffett Field, CA, August 1990.

[21] W.B. Kendall and W.J. Jacobi. Passive electro-optical sensor processing for helicopter obstacle avoidance. Contractor Report NAS2-12774, NASA, Ames Research Center, Moffett Field, CA, August 1989.

[22] K. Skifstadt and R. Jain. Range estimation from intensity gradient analysis. *Machine Vision and Applications*, 2(1):81–102, 1989.

[23] D.T. Lawton, J. Rieger, and M. Steenstrup. *Vision, Brain and Cooperative Computation*, chapter ??, pages 111–999. M.I.T. Press, Cambridge, MA, 1987.

[24] G. Medioni and R. Nevatia. Segment-based stereo matching. *Computer Vision, Graphics, and Image Processing*, 31(1):2–18, 1985.

[25] M.J. Hannah. Sri's baseline stereo system. In *DARPA Image Understanding Workshop*, Miami Beach, FL, December 1985.

[26] D.A. Luenberger. Introduction to linear and nonlinear programming. Reading, MA, 1973. Addison-Wesley.

[27] T.J. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-8(1):90–99, January 1986.

[28] O.D. Faugeras and G. Toscani. The calibration problem for stereo. In *Proceedings of IEEE Conference on Computer Vision aand Pattern Recognition*, Miami Beach, FL, June 1986.

[29] H.J. Wunsche. Detection and control of mobile robot motion by real-time computer vision. In *Proceedings of the SPIE Conference on Mobile Robots*, Cambridge, MA, October 1986.

[30] B. Sridhar, V.H.L. Cheng, and A.V. Phatak. Kalman filter based range estimation for autonomous navigation using imaging sensors. In *Proceedings of the 11th Symposium on Automatic Control in Aerospace*, Tsukuba, Japan, July 1989.

[31] B.D.O. Anderson and J.B. Moore. *Optimal Filtering.* Prentice-Hall, Englewood Cliffs, N.J, 1979.

[32] G.Bierman. *Factorization Methods for Discrete Sequential Estimation.* Academic Press, New York, NY, 1977.

[33] G.F. Fraklin and J.D. Powell. *Digital Control of Dynamic Systems.* Addison-Wesley, Reading, MA, 1980.

[34] B. Bhanu. Understanding scene dynamics. In *DARPA Image Understanding Workshop*, Palo Alto, CA, May 1989.

[35] C.R. Bidlack, C. Chen, and S. Marapane. Prospects for fully autonomous robotic systems. *Computer*, 23(2):93–95, 1990.

[36] D. Petrovic. The need for accuracy verification of machine vision algorithms and systems. In *Proceedings of IEEE Conference on Computer Vision aand Pattern Recognition*, San Diego, CA, June 1989.

[37] P. Smith. Flight data acquisition for validation of passive ranging algorithms for obstacle avoidance. In *Proceedings of the American Helicopter Society Forum*, Washington, D.C., May 1990.

[38] R. Suorsa and B. Sridhar. Validation of vision based obstacle detection algorithms for low altitude flight. In *Proceedings of the SPIE International Symposium on Advances in Intelligent Systems*, Boston, MA, November 1990.

[39] B. Bhanu and P.Symosek. Interpretation of terrain using hierarchical symbolic grouping for multi-spectral images. In *DARPA Image Understanding Workshop*, Los Angeles, CA, February 1987.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1991	Technical Memorandum

4. TITLE AND SUBTITLE

Passive Range Estimation for Rotorcraft Low-Altitude Flight

5. FUNDING NUMBERS

505-66-11

6. AUTHOR(S)

B. Sridhar, R. Suorsa, and B. Hussien

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Ames Research Center
Moffett Field, CA 94035-1000

8. PERFORMING ORGANIZATION REPORT NUMBER

A-91242

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TM-103897

11. SUPPLEMENTARY NOTES

Point of Contact: B. Sridhar, Ames Research Center, MS 210-19, Moffett Field, CA 94035-1000; (415) 604-5450 or FTS 464-5450

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified — Unlimited; Quick Release
Subject Category 01

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

The automation of rotorcraft low-altitude flight presents challenging problems in control, computer vision and image understanding. A critical element in this problem is the ability to detect and locate obstacles, using on-board sensors, and modify the nominal trajectory. This requirement is also necessary for the safe landing of an autonomous lander on Mars. This paper examines some of the issues in the location of objects using a sequence of images from a passive sensor, and describes a Kalman filter approach to estimate range to obstacles. The Kalman filter is also used to track features in the images leading to a significant reduction of search effort in the feature extraction step of the algorithm. The method can compute range for both straight line and curvilinear motion of the sensor. An experiment is designed in the laboratory to acquire a sequence of images along with the sensor motion parameters under conditions similar to helicopter flight. The paper presents range estimation results using this imagery.

14. SUBJECT TERMS

Computer vision, Rotorcraft guidance, Low-altitude flight

15. NUMBER OF PAGES

36

16. PRICE CODE

A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		