Integrated 3-D Analysis and Analysis-Guided Synthesis of Flight Image Sequences

Sanghoon Sull, *Member, IEEE*, and Narendra Ahuja, *Fellow, IEEE*

***Abstract*—This paper is concerned with three-dimensional (3-D) analysis, and analysis-guided syntheses, of images showing 3-D motion of an observer relative to a scene. There are two objectives of the paper. First, it presents an approach to recovering 3-D motion and structure parameters from multiple cues present in a monocular image sequence, such as point features, optical flow, regions, lines, texture gradient, and vanishing line. Second, it introduces the notion that the cues that contribute the most to 3-D interpretation are also the ones that would yield the most realistic synthesis, thus suggesting an approach to analysis-guided 3-D representation. For concreteness, the paper focuses on flight image sequences of a planar, textured surface. The integration of information in these diverse cues is carried out using optimization. For reliable estimation, a sequential batch method is used to compute motion and structure. Synthesis is done by using i) image attributes extracted from the image sequence, and ii) simple, artificial image attributes which are not present in the original images. For display, real and/or artificial attributes are shown as a monocular or a binocular sequence. Performance evaluation is done through experiments with one synthetic sequence, and two real image sequences digitized from a commercially available video tape and a laserdisc. The attribute based representation of these sequences compressed their sizes by 502 and 367. The visualization sequence appears very similar to the original sequence in informal, monocular as well as stereo viewing on a workstation monitor.**

***Index Terms*— Integrated segmentation and matching, integrated motion and structure estimation, consistency of structure parameters, sequential-batch processing, analysis-guided syntheses, flight images, recognition of vanishing line.**

I. Introduction

THIS paper is concerned with 3-D analysis and analysis-guided syntheses of images. Both analysis and synthesis are aimed at the characteristics of 3-D motion of an observer relative to a scene. There are two objectives of the paper. First, it presents an approach to recovering 3-D motion and structure parameters from multiple cues present in monocular images. Second, it introduces the notion that the cues that contribute the most to 3-D interpretation are also the ones that would contribute the most to realistic synthesis, thus suggesting an approach to analysis-guided compression. It should be noted that 3-D interpretation here is intended to communicate to the observer certain chosen 3-D characteristics of the scene, such as those that may be useful for navigation. Therefore, analysis, synthesis, and compression in this paper are with reference to such characteristics. It is not expected that, for example, compression and synthesis will retain the original photometric appearance of the images pixel by pixel. Of course, the algorithms presented could be combined with conventional compression techniques to achieve both 3-D and visual fidelity.

A key feature of the approach presented that helps meet both objectives is an integrated use of multiple image attributes or cues. These cues carry the motion and structure information of interest to different degrees and have different, often complementary, strengths and shortcomings. Thus when a given attribute does not contribute significantly to the estimation process, other, more pertinent cues help achieve reliable estimation. The goal is to estimate motion and structure parameters such that the estimates best explain the presence of *all* of the observed image cues throughout the image sequence.

The integrated recovery process gives the estimates of the motion and structure parameters as well as simultaneously identifies the image cues that are found to contribute to these estimates. This amounts to the identification of image characteristics that mutually consistently carry information about the relative motion and structure. This representation power of the image attributes then motivates the introduced premise for image synthesis, namely, the above attributes cost-effectively communicate to the observer the same motion and structure characteristics perceived from the original image sequence. Depictions of the scenes are synthesized using the analysis-selected image attributes. These depictions may thus also be viewed as a 3-D interpretation-based approach to image sequence representation, which obviously should be much more compact than the images themselves, i.e., the depictions should exhibit very high compression ratios while retaining 3-D perceptual characteristics. For two real image sequences used in this paper, such compression ratios of 502 and 367 per frame were achieved. Since object structure does not usually change with time, or may change slowly as with many nonrigid objects, and object motion characteristics also usually change slowly except possibly at some infrequently occurring time instants, the effective compression ratios achieved are K times the above values, where K is the number of frames over which the parameter values can be considered to be constant.

The identification and analysis of the relative strengths of different cues for the problem at hand is a research problem in

Manuscript received October 15, 1992; revised June 1, 1993. This work was supported by the Defense Advanced Research Projects Agency and the National Science Foundation under Grant IRI-89-02728. Recommended for acceptance by Editor-in-Chief A. K. Jain.

S. Sull was with the Beckman Institute and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL. He is now with NASA, Ames Research Center, Moffet Field, CA 94035-1000.

N. Ahuja is with the Beckman Institute and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801.

IEEE Log Number 9215849.

itself. In general, the available cues, and sometimes even their relative merits, depend upon the scene under consideration. In this paper we focus on the problem of an observer moving above a planar, textured surface such as while in an aircraft that is landing or taking off. The goal is to recover the translational and rotational motion of the observer and the orientation of the plane as a function of time, from multiple attributes present in the sequence of images of the plane acquired during the motion. The approach we present allows the use of the following image cues: point features, optical flow, regions, lines, texture gradient, and vanishing line. This list of cues could be changed to achieve increased robustness for any given scenario while still following the basic approach presented.

The framework for integration used in this paper is one of optimization. The objective function to be minimized is based on the differences between the observed image attributes and those corresponding to motion and structure parameters.

Synthesis is performed using the image attributes extracted from the image sequence, or by using simple, artificial image attributes that are not present in the original images. The original attributes are shown by displaying the appropriate pixels from the original images. Alternatively, we could simplify the displays, for example by using average intensities over attribute regions. Real and/or artificial attributes are each shown in a monocular as well as a binocular (stereo) sequence. Binocular display further highlights the recovered motion and structure parameters. One outcome of this is that a monocular image sequence is converted into a binocular sequence.

Section II discusses the motivation for integration of the use of different cues. It also presents an overview of our approach to the problem of integrated 3-D recovery and synthesis of motion over a planar, textured surface from a monocular image sequence. The terms, plane orientation, surface (plane) normal, and structure are used interchangeably. Sections III and IV discuss the mathematical formulation and the different steps of the algorithm, respectively. The moving objects are assumed to be rigid, piecewise planar, undergoing general 3-D motion, and viewed under perspective projection. Section V presents the performance of the integrated approach, the details of implementation and the results obtained in experiments with one synthetic sequence and two sequences of 29 and 33 images, digitized from a commercially available videotape and a laserdisc of films taken from flying aircrafts. Estimates of image compression achieved are given. The synthesized (visualization) sequences appear compellingly similar to the originals when the two are played side by side on a SUN (monocularly) and SGI workstation monitor (binocularly), though we have not performed any rigorous psychophysical experiments to test the perceptual similarity. Section VI presents conclusions and planned extensions.

II. Motivation and Approach

In this section we first discuss motivation behind the two major themes of this paper, integrated 3-D analysis and analysis-guided syntheses. Then, we present an overview of the approach we have developed to perform integrated recovery and synthesis of motion above a ground plane.

A. The Need for Integrated Estimation

Three-dimensional image interpretation by an integrated analysis of multiple cues has many advantages. Some of these are summarized below.

A feature of any *a priori* specified type, intended for 3-D analysis, may not occur in a given scene. For example, there are no line features in the image sequence shown in Fig. 8(a). Further, even if the feature is present, the detection process may often miss it. For example, it is very difficult to extract the same point features between any two consecutive images in the sequences used in our experiments. (See Figs. 8(a) and 10(a).) To reduce such problems, it is desirable to use a large variety of features.

Using multiple features also helps achieve three additional, related advantages. First, it increases the likelihood that the overall count of the detected features is larger. Second, the features are more likely to be spatially well distributed. Each of these two properties helps increase the precision of the resulting estimates if there is no outlier present, as we will see in Section V–B. Further, the integrated method using multiple frames can greatly reduce the effect of the outliers, as discussed in Section V–B, without using a computationally expensive robust regression method. Third, using a large variety of features reduces the probability that the features form configurations that are degenerate for 3-D estimation. For example, if the detected feature points are on a straight line, the motion and structure cannot be computed from them. Considering additional types of features at the same time (e.g., lines) reduces the probability of encountering only degenerate configurations.

Different detected features may also act as reliability filters for each other. For example, the optical flow may be used only at those locations where a point feature detector responds. This helps in selecting flow information that is reliable (since point features are usually detected at locations having high intensity gradients), while simultaneously avoiding the loss of computation time and accuracy of results due to processing less reliable flow.

Different cues often have complementary strengths and shortcomings. For example, region correspondences are easier to find because regions have nonzero sizes, but they give coarse estimates of motion; on the other hand, points and point correspondences are harder to find but have better positional accuracy and hence give more accurate estimates. Further, certain features may have special significance and utility for a specific types of scene. For example, for images taken from a flying aircraft, such as those used in Section V–C, the vanishing line is an important cue since it carries information about aircraft orientation with respect to the ground plane. (The vanishing line is defined as the intersection of the image plane with a plane that includes the camera center and is parallel to the object plane.)

B. The Need for Integrated Estimation and Synthesis

A central theme of this paper is the introduction of the following notion about image synthesis: Displaying those image attributes selected and used for 3-D recovery from an image

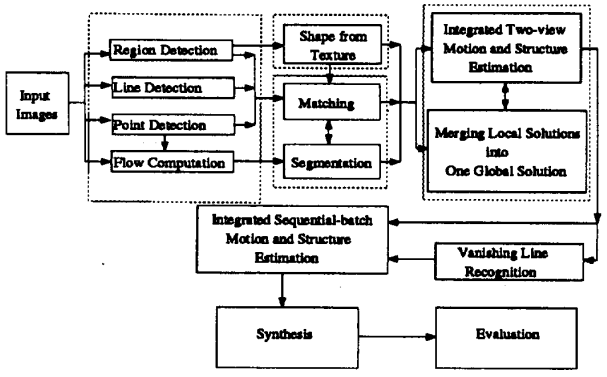


Fig. 1. A flow diagram summarizing the presented approach.

sequence communicates to the observer the same motion and structure characteristics as perceived by the observer from the original image sequence. In other words, the analytical power of the attributes is viewed as an indicator of 3-D information these attributes contain, and therefore it is expected that their presence in a synthesized image sequence would be effective in communicating the 3-D scene characteristics. The synthetic sequence is in general much simpler than the original sequence and hence results in a significant data compression.

By analyzing and visualizing the images taken from the flying airplane, an approach like ours can be used to automatically build a database with realistic 3-D and photometric structure. Integrated analysis and synthesis of the flight images can also help the pilot navigate by providing enhanced views of the scene. For example, the pilot can be presented views of a runway (such as that shown in Fig. 6) during takeoff and landing, in which artificial features have been added to the runway surface to enhance the perceptual salience of the images.

C. Overview of the Approach

This section outlines our approach to integrated analysis and analysis-guided synthesis, which consists of eight major steps as shown in Fig. 1.

The goal of the first step is to independently detect points, lines, and regions in each frame. Optical flow is computed between each pair of adjacent frames.

In the second step, the plane orientation is estimated from the detected regions in each frame using texture gradient.

The third step establishes correspondences between features and segments features in each pair of adjacent images using a first-order model of the image plane displacement of the features described in [14]. In general, correspondences are not found for all features contained in an object; some of them remain unmatched. The outliers of the computed flow vectors are also removed at this stage.

In the fourth step, the correspondences found are used to merge any distinct first-order segments into the different planes if they have compatible motion and structure parameters. In the problem at hand, the largest moving plane is selected since we are interested in the ground. Then, for this plane segment, the motion and structure parameters are linearly computed from pairs of adjacent images. This yields dual solutions. One of these solutions is selected by using the structure estimates from the previous frames. If the solution gives the estimate of plane orientation that yields a vanishing line within the image, then those outlier features that are on the side of the estimated vanishing line away from the ground are excluded from the plane segment.

In the fifth step, the vanishing lines, if they exist, are identified from the set of detected lines using two-view estimates obtained in the fourth step.

The objective of the sixth step is to use the established feature correspondences to determine robust motion and structure parameters by using multiple frames. These parameters were computed in a linear fashion from pairs of adjacent images in the fourth step. However, when an image is paired with its predecessor and successor images in the sequence, the resulting structure parameters will in general not be identical. Thus, the requirement of such consistency of structure parameters must be explicitly enforced. This makes the motion and structure estimation a nonlinear problem. To enforce this requirement, we must consider a batch of frames at a time. We therefore perform motion and structure estimation over a sliding window of N frames along the image sequence. For each such window, the motion and structure parameters are estimated by minimizing an objective function that is proportional to the image plane disparity between observed image attributes and those corresponding to motion and structure parameters. We note here that it is not necessary to track features to enforce the structure consistency since all features are on the plane and consistency means conservation of the orientation determined by the features. The orientation estimates obtained from candidate vanishing lines and texture are included in the objective function so that any deviation of the result from these estimates is penalized in proportion to the support the estimates have. The motion and structure estimates obtained from each batch are compatible with the attributes of the images in the batch. Clearly, the larger the batch, the more compatible the estimates will be with the image sequence at the expense of computation time and memory. The motion parameters derived from batch computations are sequentially updated. The result of the above 3-D analysis is a set of estimates of plane orientation, rotation, and normalized translation parameters. Then, the translation and structure parameters are scaled starting from the initial frame at t_0.

In the seventh step, the input sequence is synthesized using the image attributes and the result of the 3-D analysis.

Finally, in the eighth step, the result of 3-D analysis is evaluated using the estimated vanishing lines, the average image error, and the visualization sequence.

III. Mathematical Formulation

In this section, we present the mathematical formulation of our approach. First, we present the equations that are used to represent the displacement field induced by a rigid motion of a planar surface. Second, we describe how to noniteratively estimate motion and structure parameters from individual cues

such as points (or flow), lines, and regions using two frames. We also describe the estimation of the plane orientation from texture gradient and vanishing line in a single image. Third, we describe the methods for integrated estimation of motion and structure from multiple cues: noniterative estimation from two views, iterative estimation from multiple views, and sequential-batch estimation.

A. Description of Displacement Field

Let the right-handed coordinate system (X, Y, Z) be fixed on the camera with the origin coinciding with the projection center of the camera. Without loss of generality, we assume that the focal length is unity. Thus, the image plane is located at $z = 1$. Then, the perspective projection (x, y) on the image of a point (X, Y, Z) is given by

$$x = X/Z,$$
$$y = Y/Z.$$

Consider a point P on the object in 3-D. Let $\vec{X} = [X, Y, Z]'$ be the 3-D coordinate vector of P at time t_1, and let $\vec{X}' = [X', Y', Z']'$ be the corresponding vector at time t_2. Let $\vec{T}$ and $\mathbf{R}$ denote the translation vector and rotation about the unit axis $\vec{n}_\omega = [n_x, n_y, n_z]'$ by an angle ω, respectively. Then,

$$\vec{X}' = \mathbf{R}\vec{X} + \vec{T}, \tag{1}$$

where

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2}$$

and

$$\vec{T} = [T_X T_Y T_Z]'.$$

Let (x, y) and (x', y') be the image coordinates corresponding to $\vec{X}$ and $\vec{X}'$, respectively. If the point P is on the plane $aX + bY + cZ = 1$ at time t_1, then

$$Z = \frac{1}{ax + by + c}. \tag{3}$$

Here, we represent the plane by using $aX + bY + cZ = 1$ rather than $Z = pX + qY + r$, which is often used [1], [3], since we cannot express the plane when $c = 0$. This case frequently occurs, for example, when the vanishing line is seen in the image taken from the aircraft or the navigating vehicle. Hence, from (1), (2), and (3), we get

$$x' = \frac{X'}{Z'} = \frac{a_1 x + a_2 y + a_3}{a_7 x + a_8 y + a_9} \tag{4}$$

$$y' = \frac{Y'}{Z'} = \frac{a_4 x + a_5 y + a_6}{a_7 x + a_8 y + a_9}, \tag{5}$$

where

$$\left.\begin{array}{lll} a_1 = r_{11} + aT_X & a_2 = r_{12} + bT_X & a_3 = r_{13} + cT_X \\ a_4 = r_{21} + aT_Y & a_5 = r_{22} + bT_Y & a_6 = r_{23} + cT_Y \\ a_7 = r_{31} + aT_Z & a_8 = r_{32} + bT_Z & a_9 = r_{33} + cT_Z \end{array}\right). \tag{6}$$

Then, the displacement vector (D_x, D_y) is defined as

$$D_x \stackrel{\text{def}}{=} x' - x$$
$$D_y \stackrel{\text{def}}{=} y' - y. \tag{7}$$

Note that the surface structure represented here by (a, b, c) can only be estimated up to the scale factor if the translation is nonzero. In general, for a plane $a_k X + b_k Y + c_k Z = 1$ at t_k, we define

$$\text{scale}_k \stackrel{\text{def}}{=} ||a_k^2 + b_k^2 + c_k^2||. \tag{8}$$

Then, scale_k simply represents the distance from the camera center to the plane. Setting scale_k equal to one, $\vec{n}_{S,k} = [a_k, b_k, c_k]'$ becomes the unit surface normal that can be parametrized by two spherical coordinates (latitude and longitude).

B. Estimation from Individual Cues

In this section, we describe how to noniteratively estimate motion and structure parameters from individual cues such as points (or optical flow), lines, and regions using two frames. The basic approach used in two-view estimation is as follows: First, we linearly solve for the intermediate parameters $a_1, \cdots, a_9$, and second, from the intermediate parameters obtained, we noniteratively compute the dual set of parameters for motion and plane orientation using the existing methods in [4], [8]. Next, we describe the estimation of the orientation of a textured plane in each frame from image texture gradient that is based on the method presented in [17]. We also describe the estimation of the orientation of a plane from a vanishing line in each frame.

Estimation from Point Correspondences From (4) and (5), we have the two equations for each point correspondence (or optical flow vector):

$$xa_1 + ya_2 + a_3 - xx'a_7 - x'ya_8 - x'a_9 = 0 \tag{9}$$
$$xa_4 + ya_5 + a_6 - xy'a_7 - yy'a_8 - y'a_9 = 0. \tag{10}$$

Using the above two equations, if four or more point correspondences (or flow vectors) are given, we linearly compute the eight coefficients $a_1, \cdots, a_8$ with a_9 set to 1 since the nine coefficients can only be determined up to a scale factor. Then, using the algorithms presented in [4] and [8], we can noniteratively solve for the motion and plane normal.

To evaluate the motion and structure estimates obtained, we define the image error of a point correspondence i between t_k and t_{k+1} as

$$E_{k,i,P} \stackrel{\text{def}}{=} \sqrt{E_{x,k,i,P}^2 + E_{y,k,i,P}^2} \tag{11}$$

where $E_{x,k,i,P}$ and $E_{y,k,i,P}$ are defined by the residual errors of (4) and (5), respectively. $E_{k,i,F}$ for a flow vector is defined in the same way.

Estimation from Line Correspondences: Liu and Huang [16] presented linear and nonlinear motion algorithms based on straight line correspondences from three views. They showed that motion cannot be determined uniquely from two views. When the lines are on the same plane, a linear algorithm can be formulated from two views. In this case, if we compute the intersections of each pair of lines, we can use the existing point-based algorithms for a planar surface [4], [8]. However, it is desirable to use line features directly if we can develop the linear equations from the lines on a plane for the following reasons: First, if the point-based equations are used for the intersection points from the lines with similar slopes, the equations ((9) and (10)) for those points are effectively given unfairly large weights since the intersection points are far from the image plane boundary (i.e., large image coordinates). Second, even though no intersection point is obtained by the two parallel lines in the image plane, the equations obtained from those lines can still constrain the range of the motion and structure parameters.

Given a pair (L_1, L_2) of the corresponding lines at t_1 and t_2, the 2-D equations of L_1 and L_2 in the image plane are given by $A_1x + B_1y + C_1 = 0$ and $A_2x + B_2y + C_2 = 0$, respectively. Consider two end points P and Q on a line at t_1 in 3-D. Let $\vec{X}_p$ and $\vec{X}_q$ be the 3-D coordinate vectors of P and Q at t_1, respectively. We define (x_p, y_p) and (x_q, y_q) as the image coordinates of $\vec{X}_p$ and $\vec{X}_q$, respectively. Then, the image coordinates $(\widehat{x}_p, \widehat{y}_p)$ and $(\widehat{x}_q, \widehat{y}_q)$ at t_2 predicted from the current motion and structure estimates (i.e., nine coefficients $a_1, \cdots, a_9$) are expressed using (4) and (5). Since the corresponding points on the corresponding lines are not known, we cannot use the constraints based on point correspondences. Instead, we use the constraint that the corresponding line L_2 at t_2 and the predicted line from the estimates should be identical in the sense that the perpendicular distances from the two end points on the predicted line to L_2 should be zero. This is illustrated in Fig. 2.

Let l_p and l_q be the perpendicular distances from two predicted image coordinates to L_2, respectively. Since both l_p and l_q should be zeroes, we minimize the sum of squares of the following two terms for each line correspondence,

$$l_p \stackrel{\text{def}}{=} \frac{A_2\widehat{x}_p + B_2\widehat{y}_p + C_2}{\sqrt{A_2^2 + B_2^2}} \tag{12}$$

$$l_q \stackrel{\text{def}}{=} \frac{A_2\widehat{x}_q + B_2\widehat{y}_q + C_2}{\sqrt{A_2^2 + B_2^2}}. \tag{13}$$

After replacing $(\widehat{x}_p, \widehat{y}_p)$ and $(\widehat{x}_q, \widehat{y}_q)$ in the above two equations with (4) and (5), we eliminate the denominator terms. Then, we arrive at the two equations given at the bottom of the page for each line correspondence. Using the above two equations, if four or more line correspondences are given, we linearly compute the eight coefficients $a_1, \cdots, a_8$ with a_9 set to 1 since the nine coefficients can be determined only up to a scale factor. Then, using the algorithms presented in [4] and [8], we can noniteratively solve for the motion and plane normal.

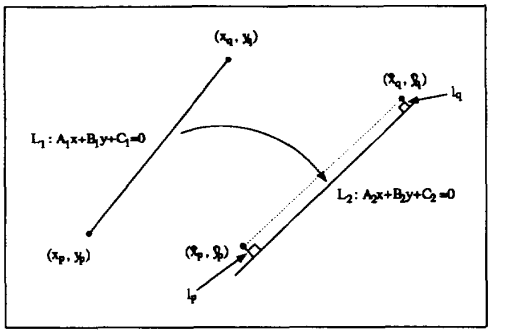


Fig. 2. Illustration of constraint used in line-based estimation.

To evaluate the motion and structure estimates obtained, we define the image error of a line i between t_k and t_{k+1} as

$$E_{k,i,L} \stackrel{\text{def}}{=} \sqrt{\frac{E_{p,k,i,L}^2 + E_{q,k,i,L}^2}{2}}, \tag{16}$$

where $E_{p,k,i,L}$ and $E_{q,k,i,L}$ are defined by the residual errors of Eqs. (12) and (13), respectively.

Estimation from Region Correspondences: Using (4), (5), and (7), the displacement vector (D_x, D_y) can be approximated by

$$D_x = a_3 + (a_1 - a_9)x + a_2y - a_8xy - a_7x^2 \tag{17}$$

$$D_y = a_6 + a_4x + (a_5 - a_9)y - a_7xy - a_8y^2, \tag{18}$$

if we assume that 1) $\frac{T_Z}{Z} \ll 1$, 2) the field of view of the camera is small, and 3) the rotation about X and Y axes is small [2], [12]. These assumptions, which are quite common in motion analysis are not very restrictive since the field of view of a camera is small in practice and the amount of motion is small if the time interval between two images is short. Note that the second-order polynomials for (D_x, D_y) can be derived without any approximation by using the instantaneous velocity formulation for the optical flow. Let M and N be the corresponding regions at two time instants. Then, using

$$\frac{A_2x_pa_1 + A_2y_pa_2 + A_2a_3 + B_2x_pa_4 + B_2y_pa_5 + B_2a_6 + C_2x_pa_7 + C_2y_pa_8 + C_2a_9}{\sqrt{A_2^2 + B_2^2}} = \tag{14}$$

$$\frac{A_2x_qa_1 + A_2y_qa_2 + A_2a_3 + B_2x_qa_4 + B_2y_qa_5 + B_2a_6 + C_2x_qa_7 + C_2y_qa_8 + C_2a_9}{\sqrt{A_2^2 + B_2^2}} = 0. \tag{15}$$

Jacobian, we can derive the following two equations for each region correspondence [12]:

$$\frac{N_{10}}{N_{00}} - \frac{M_{10}}{M_{00}} = a_3 + (a_1 - a_9)\frac{M_{10}}{M_{00}} + a_2\frac{M_{01}}{M_{00}} - a_8\frac{M_{11}}{M_{00}} - a_7\frac{M_{20}}{M_{00}} \quad (19)$$

$$\frac{N_{01}}{N_{00}} - \frac{M_{01}}{M_{00}} = a_6 + a_4\frac{M_{10}}{M_{00}} + (a_5 - a_9)\frac{M_{01}}{M_{00}} - a_7\frac{M_{11}}{M_{00}} - a_8\frac{M_{02}}{M_{00}}, \quad (20)$$

where

$$N_{ij} \stackrel{\text{def}}{=} \int\int_N x^i y^j dxdy$$
$$M_{ij} \stackrel{\text{def}}{=} \int\int_M x^i y^j dxdy. \quad (21)$$

Eqations (19) and (20) represent the constraints from which the motion and structure parameters are to be estimated using a sufficiently large number of region correspondences. Therefore, we can linearly compute 8 coefficients $a_1, \cdots, a_8$ from four or more region correspondences with a_9 set to one since the coefficient a_9 can have any value in the above two equations. Then, using the algorithms presented in [4] and [8], we can noniteratively solve for the motion and plane normal.

To evaluate the motion and structure estimates obtained, we define the image error of a region i between t_k and t_{k+1} as

$$E_{k,i,R} \stackrel{\text{def}}{=} \sqrt{E^2_{x,k,i,R} + E^2_{y,k,i,R}}\ , \quad (22)$$

where $E_{x,k,i,R}$ and $E_{y,k,i,R}$ are defined by the residual errors of (19) and (20), respectively.

Estimation from Texture Gradient: Consider the problem of estimating the orientation of a planar textured field from gradients of image texture properties. Blostein and Ahuja [17] extract texture elements while simultaneously recovering the orientation. A planar surface is characterized by the triple (A_C, S, T), where A_C is the texel area expected in the image center, S is the slant, and T is the tilt. Slant, ranging from to 90°, is the angle between the planar surface and the image plane. Tilt, ranging from 0° to 360°, is the direction in which the surface normally projects in the image. Given image coordinates (x, y),

$$\theta \stackrel{\text{def}}{=} \arctan((x\cos T + y\sin T)\text{FOV}), \quad (23)$$

where FOV is the field of view of the camera. A_i, the area of a texel at location (x, y) in the image is related to A_C, the area of a texel at the image center, by

$$A_i = A_C(1 - \tan\theta\tan S)^3. \quad (24)$$

Here, $\theta + S$ should be less than 90° since for 90° the plane becomes parallel to the line from the camera center to the point (x, y) in the image plane. Then, we have $(1-\tan\theta\tan S) > 0$. When S is equal to or greater than 90°, A_C is not defined. In order to estimate the best orientation, they use the gradient of texture element areas. For each (A_C, S, T), (24) gives the expected texel area at each image location. These expected areas are compared to the extracted region areas in the image, and a fit-rating is computed for the plane. The plane that receives the highest fit-rating is selected as the estimate of the textured surface. Then, the candidate texels that support the best planar fit are interpreted as true image texels. In this paper, we extend the method by allowing multiple texture patterns on a planar surface. We first detect regions or candidate texels. We make groups of regions such that the minimum distance between any region and the others in each group is below a threshold. Then, for each (S, T), we compute the support for different A_C values using (24). The voting is carried out for each group of detected regions where A_i is given by the area of the region. We can select the set of the highest voted values of A_C and then compute the total support for (S, T). The set of (S, T) that receives the highest support becomes the estimate of the planar orientation. Then, the candidate regions that support the best planar fit are interpreted as true image texels.

For a 3-D plane equation given by $aX + bY + cZ = 1$, we can easily convert the values of slant and tilt to (a, b, c) using the following equations:

$$S = \arccos c \quad (25)$$
$$T = \arctan\frac{(-b)}{(-a)}. \quad (26)$$

Orientation from Vanishing Line: Consider a 3-D plane given by $aX + bY + cZ = 1$. Its surface normal $\vec{n}_s$ is represented by the vector $[a, b, c]'$. Since the vanishing line is defined as the intersection of the image plane $(Z = 1)$ and $aX + bY + cZ = 0$, it is expressed as $ax + by + c = 0$ in terms of the image coordinates x and y. Therefore, if we know the surface normal, the vanishing line is determined in the image plane. Conversely, if we know the equation of the vanishing line in the image plane, the surface normal is determined up to a scale factor.

C. Integrated Estimation

This section presents approaches to motion and structure estimation that integrate the information from all cues discussed in Section III–B. First, we describe linear estimation using two frames. Then we present a nonlinear method using multiple frames of batch size N for robust estimation. Finally, we present a sequential-batch method.

Integrated Linear Estimation Using Two Frames: For each pair of successive frames at t_k and t_{k+1}, we first solve for the intermediate parameters $a_1, \cdots, a_9$. To solve the six equations for points, lines, and regions simultaneously (see (9) and (10), (14) and (15), and (19) and (20)), we linearly compute the eight coefficients $a_1, \cdots, a_8$ with a_9 set to 1, since a_9 can have any value. Each equation is multiplied by a factor proportional to the significance of the corresponding cue. Then we noniteratively compute the parameters for motion and plane orientation.

Integrated Nonlinear Estimation Using Multiple Frames: The solution obtained by the method to be presented in this

section minimizes an image error between the observed cues and those corresponding to the motion and structure estimates.

To obtain a measure of inconsistency between the motion and structure estimates obtained and the different cues used, we first define the total image error between t_k and t_{k+1} as shown in (27) at the bottom of the page, where $E_{k,i,P}$, $E_{k,i,F}$, $E_{k,i,L}$, and $E_{k,i,R}$ are defined in (11), (16), and (22), and $n_P(k)$, $n_F(k)$, $n_L(k)$ and $n_R(k)$ are the numbers of point correspondences, flow vectors, line correspondences, and region correspondences between t_k and t_{k+1} for a planar patch, respectively. Note that each error term has the same unit (image plane error). Each error term $E_{k,i}^2$ is multiplied by a weight $\lambda_{k,i}$, which reflects the significance of the corresponding cue. Then, we define the average image error for one cue between t_{k_1} and t_{k_2} as

$$\overline{E_{k_1,k_2,I}} \stackrel{\text{def}}{=} \sqrt{\sum_{k=k_1}^{k_2-1} \frac{E_{k,I}^2}{\lambda_{k_1,k_2,I}}}, \tag{28}$$

where

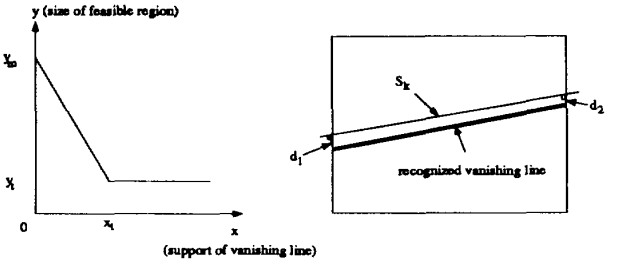
$$\lambda_{k_1,k_2,I} \stackrel{\text{def}}{=} \sum_{k=k_1}^{k_2-1} \left(\sum_{i=1}^{n_P(k)} \lambda_{k,i,P} + \sum_{i=1}^{n_F(k)} \lambda_{k,i,F} + \sum_{i=1}^{n_L(k)} \lambda_{k,i,L} + \sum_{i=1}^{n_R(k)} \lambda_{k,i,R} \right). \tag{29}$$

For each window of batch size N, we define the following objective function, which is to be minimized with respect to structure and motion parameters:

$$G(M_k, S_k) \stackrel{\text{def}}{=} \sum_{k=0}^{N-2} \frac{E_{k,I}^2}{\lambda_{0,N-1,I}} + \sum_{k=0}^{N-1} (\lambda_{k,V} E_{k,V}^2 + \lambda_{k,T} E_{k,T}^2), \tag{30}$$

where $E_{k,I}$ and $\lambda_{0,N-1,I}$ are defined in (27) and (29), respectively, and $E_{k,V}$ and $E_{k,T}$ are defined below.

The first term in (30) is normalized by $\lambda_{0,N-1,I}$ so that it represents the average image error for one cue. $E_{k,V}(S_k)$ is the penalty term that makes the orientation parameters stay within a certain range of the initial values computed from recognized vanishing lines. During the iterative optimization, the objective function involves large penalty when the iteration variables representing the unit surface normals leave the feasible regions. One simple way of defining the penalty terms at t_k and thus $E_{k,V}$ is as follows. The size of the feasible region, y, is determined by the support x computed from the vanishing line recognition stage as shown in Fig. 3(a), where



(a) (b)

Fig. 3. Illustration of the penalty term for vanishing line at t_k. (a) Size of feasible region versus support of the recognized vanishing line. (b) Definition of d_1 and d_2.

x_t, y_t, and y_m are the threshold values. Let d_1 (d_2) be the perpendicular distances in pixels from one (the other) end point of the recognized vanishing line to the vanishing line corresponding to the iteration variable S_k (Fig. 3(b)). If we let

$$d(x) \stackrel{\text{def}}{=} \max(y(x), y_t), \tag{31}$$
$$l_1 \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } d_1 \le d(x) \\ d_1 & \text{otherwise,} \end{cases}$$

and define l_2 in the same way, then the image error $E_{k,V}$ is defined as

$$E_{k,V} \stackrel{\text{def}}{=} \sqrt{l_1^2 + l_2^2}. \tag{32}$$

$E_{k,T}$ is the penalty term for texture that is defined similar to $E_{k,V}$. The size of the feasible region is determined by the support obtained from the texture gradient algorithm.

This objective function combines the contributions of multiple features to the scene characteristics to be estimated. Each contribution is weighted by a factor λ. Let M_k be set of motion parameters between t_k and t_{k+1}. Let $S_k = (a_k, b_k, c_k)$ be the unit surface normal $\vec{n}_{S,k}$ at t_k. For each overlapping batch of size N, we iteratively minimize the objective function ((30)) with respect to M_k and S_k where k runs from 0 to $N-2$ and $N-1$, respectively, without loss of generality. Since the number of the iteration variables is large, we use the motion and structure relationships to reduce the number of variables. First, we can relate the unit surface normals with the interframe rotations:

$$\vec{n}_{S,k+1} = \mathbf{R}_{k,k+1} \vec{n}_{S,k}. \tag{33}$$

Secondly, from the six equations ((9) and (10), (14) and (15), and (19) and (20)) with (6), the interframe translational velocities are linearly computed when the unit surface normals and interframe rotations are given. Therefore, once S_0 or $\vec{n}_{S,0}$ and interframe rotational velocities are given, the other unit surface normals and translations are linearly computed. S_0 is represented by two spherical angles, and each interframe

$$E_{k,I} \stackrel{\text{def}}{=} \sqrt{\sum_{i=1}^{n_P(k)} \lambda_{k,i,P} E_{k,i,P}^2 + \sum_{i=1}^{n_F(k)} \lambda_{k,i,F} E_{k,i,F}^2 + \sum_{i=1}^{n_L(k)} \lambda_{k,i,L} E_{k,i,L}^2 + \sum_{i=1}^{n_R(k)} \lambda_{k,i,R} E_{k,i,R}^2} \tag{27}$$

rotation is expressed by three variables for rotation axis and angle. We can thus reduce the number of search parameters of the objective function from $8(N-1)$ to $2+3(N-1)$:

$$\begin{aligned}&\min_{M_k,S_k} G(M_k,S_k)\\&\quad=\min_{S_k,k=0,\cdots,N-1,M_k,k=0,\cdots,N-2} G(M_k,S_k)\\&\quad=\min_{S_0,\mathbf{R}_{k,k+1},k=0,\cdots,N-2} G(S_0,\mathbf{R}_{k,k+1}).\end{aligned}\tag{34}$$

We note that (33) also enforces the consistency of structure parameters explained in Section II-C. Tracking of each feature is not necessary to enforce the structure consistency since all features are on the plane, whose structure is given by its orientation.

Since we are concerned with monocular sequences, we have the problem of unknown scale for the estimated structure [10]. The scale factor of any two consecutive images depends on the scale factor of the first two images. Then, translations cannot be linearly computed even though S_0 and $\mathbf{R}_{k,k+1}$ for each k are given, resulting in an increase of the parameter space of iteration. This problem is also avoided by linking multiple frames through the unit surface normals in the objective function (30). Note that if we consider the pairs of frames between $t_0-t_1, t_0-t_2, \cdots, t_0-t_{N-1}$ in (30) instead of pairs of successive frames in order to avoid the unknown scale problem, we must track each feature from t_0 to t_{N-1}. For a given initial surface normal and interframe rotations, the other surface normals are determined through (33) (which assumes that scale_k in (8) is one). Then, the normalized translation parameters corresponding to the unit $\text{scale}'_k s$ are computed in the iteration process instead of true scaled translation. As we can see from the predefined equations, this does not affect the average image error given by the first term in the objective function G.

Then, by using the estimates of the initial surface normal and interframe rotations, we rescale structure and translation parameters sequentially starting from the initial frame by using any set of the points which are on the plane. Those points need not be the same throughout. We have three sources of initial guess for iteration variables (the initial surface normal and the interframe rotations). We can use the dual solutions obtained in closed form based on two successive frames, the detected vanishing lines, and the estimates obtained from the previous batch of frames. We try the three initial guesses and then select one that gives the minimum value of the objective function.

Integrated Sequential-Batch Estimation: It is desirable to use all the available frames as a batch and the interframe rotations and initial orientation as variables, and to iteratively minimize the objective function defined in (30). However, since the number of frames is very large, the above total batch method is impractical due to its enormous memory and computation requirement. Therefore, motion parameters obtained from the overlapping batches are sequentially updated. The size N of a batch is typically 3, where the number of iteration variables is 8.

After minimizing the objective function for a batch of size N starting at t_l, we determine the motion parameters $\mathbf{m}(l)$ between t_l and t_{l+1} as follows, though we can use any sequential updating algorithm available in the literature:

$$\mathbf{m}(l)=\sum_{k=l-(N-2)}^{l}\frac{\hat{\mathbf{m}}_k(l)}{\epsilon+\text{error}_k},\tag{35}$$

where $\hat{\mathbf{m}}_k(l)$ consist of the motion estimates between t_l and t_{l+1} obtained from the batch that runs from t_k to t_{k+N-1}, error_k is the square root of the minimum value of the objective function ((30)) from the batch computation between t_k and t_{k+N-1}, and ϵ represents a small positive number. Note that batch computations provide the normalized translation parameters with respect to the unit surface normal for each frame. Therefore, we do not need to be concerned about scale factors here even though we do not know the scales of the translation parameters.

Since vanishing lines are usually visible for flight images, it is important to compute the orientation parameters in such a way that the reconstructed vanishing lines change consistently with the estimated interframe rotation parameters over successive frames. Consider two overlapping batches that start at t_l and t_{l+1}, respectively. At t_l, the best orientation parameters are computed based on the frames from t_l to t_{l+N-1}. Then, at t_{l+1}, new orientation parameters are estimated based on the frames from t_{l+1} to t_{l+N}. Even though these two sets of estimated parameters minimize the objective functions within their batches, respectively, the surface normal values for the frame t_{l+1} estimated from two overlapping batch windows are not generally equal for noisy images. Further, a small difference between two estimated unit surface normal vectors for the same frame results in large errors in the image plane when they are viewed as vanishing lines. Therefore, we avoid using the updated orientation parameters as the final estimates for each frame as the new batch computations become available. Instead, at t_l, the ground orientations from t_0 to t_l are computed using $\mathbf{m}(0), \cdots, \mathbf{m}(l-1)$ and the reference ground orientation at t_r, which is the estimated value for the batch from t_r to t_{r+N-1}. In this paper, this batch is simply chosen by the criterion that it has the minimum value of the objective function (30) among the batches that start at t_0 through t_l. Since translation parameters are easily computed given rotation and surface orientation parameters, we basically need to update rotation parameters only.

Finally, we rescale translation and structure parameters starting from the initial frame by using any set of the points on the plane.

IV. Algorithm

This section describes more precisely the eight steps of the algorithm outlined in Section II–C. The algorithm uses points, flow, regions, lines, vanishing lines, and texture as the image cues. Given an image sequence $I_0, \cdots, I_{K-1}$, the main steps of the algorithm are as follows.

Step 1: Detection of Image Cues: Points, lines, and regions are detected independently in each frame. Optical flow is computed between each pair of adjacent frames.

Step 2: Estimation from Texture Gradient: Plane orientation and its support value are independently estimated from the detected regions for each frame using texture gradient.

Step 3: Integrated Segmentation and Matching Using the First-Order Model: This step groups points, flow, regions, and lines in each pair of adjacent images into subsets corresponding to the local planar surface patches of moving objects based on the similarity of six first-order (affine) image plane displacement coefficients [14]. This step also establishes the correspondences between points, regions, and lines in each pair of adjacent images.

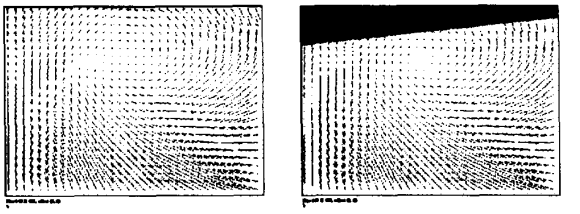
Step 4: Merging Local Solutions into One Global Solution: Since the features corresponding to different cues are oversegmented by the first-order model, a merging step is performed. For flight images, the largest plane segment corresponding to a ground is identified and is used to linearly estimate the dual solutions. One of the dual solutions, the one that is closer to the structure estimate from the previous frames, is selected. Since there remain the cues which were not grouped (and *therefore unmatched*) in Step 3, they are matched and acquired by the plane segment if they are within a distance IR from the plane segment and satisfy the same constraints on motion and structure as the plane given by the selected solution. After recomputing the solution, the outlier cues that are not on the ground are removed if this solution gives the estimate of plane orientation, which yields a vanishing line within the image. Then, motion and structure parameters are recomputed

Since points, flows, lines, and regions are often oversegmented by the first-order displacement model, a merging step is necessary. Any distinct first-order segments are merged into the different planes if they have compatible motion and structure parameters. Merging decision is based on the average image error in (28), which is computed using the nine coefficients $a_1, \cdots, a_9$ obtained from the two-view integrated linear estimation algorithm. We use merging criteria similar to those used in [2].

Then, the largest plane segment is identified under the assumption that it corresponds to the ground with respect to a moving observer. We have dual solutions for the planar case. The solution that is closer to the predicted plane orientation from the previous frames is selected.

Next, the remaining unmatched and ungrouped cues are matched and added to the largest plane if they are within a distance IR from the plane and satisfy the corresponding constraints on motion and structure defined by (4) and (5), (12) and (13), or (19) and (20), where $a_1, \cdots, a_9$ are given by the above solution.

Then, dual solutions are recomputed and one solution is selected in the same way. If this solution gives the estimate of plane orientation that yields a vanishing line within the image, the outlier cues that are not on the ground are removed. Note that we can not define outliers if the solution yields no vanishing line. In Fig. 4, we show two solutions corresponding to the same coefficients $a_1, \cdots, a_9$. Those coefficients are the two-view linear solution between t_4 and t_5 for the second real image sequence used. Note that two solutions give the same flow, but the flow is not defined for solution (b) in the dark part of the image plane, which corresponds to *sky*. Then, two sets of motion and structure parameters are linearly recomputed. Again, one solution is selected in the same way.



(a) (b)

Fig. 4. Displacement fields of dual solutions for the same set of coefficients: $a_1 = 0.9159$, $a_2 = -0.0677$, $a_3 = 0.0062$, $a_4 = 0.0890$, $a_5 = 0.9515$, $a_6 = -0.0133$, $a_7 = -0.1972$, $a_8 = 0.0313$, and $a_9 = 1$. (a) First solution: $\vec{n}_S = [0.0723, -0.0758, 0.9945]'$, $\vec{T} = [-0.2085, 0.0048, 0.0696]'$, $\vec{n}_\omega = [0.1220, 0.9145, 0.3858]'$, $\omega = 13.44deg$. (b) Second solution: $\vec{n}_S = [-0.9711, 0.1066, 0.2135]'$, $\vec{T} = [0.0404, -0.0185, 0.2153]'$, $\vec{n}_\omega = [0.1303, -0.0327, 0.9909]'$, $\omega = 4.35deg$.

Step 5: Recognition of Vanishing Lines: The vanishing line is identified from the set of detected lines in each frame by using two-view estimates.

Successive two-view estimates of plane orientation are used for the recognition of a vanishing line from a set of detected lines in each frame. A simple way of identifying vanishing lines from sets of detected lines at t_k and t_{k+1}, given two-view estimates (that is, plane orientations or vanishing lines and rotation parameters), is as follows:

1) For each frame, compute the support value for each detected line based on distance and slope difference in the image plane between the predicted vanishing line and each detected line. Only those lines whose lengths are longer than a threshold TL are considered.
2) For each pair of lines in two frames, compute the total support based on support values from step 1 and (33).
3) Find the pairs of lines that give the maximum and the second maximum of the total support values. The pair of lines with the maximum value corresponds to the first candidates of recognized vanishing lines for two successive frames. The difference in the total support values represents the final support that is normalized between 0 and 1. Here, we get a low value of confidence if there are several candidate vanishing lines.

The confidence in the recognized vanishing lines improves if the results from the successive pairs of frames are consistent. We also note that vanishing lines are not always recognized.

Step 6: Integrated Nonlinear Batch Estimation and Sequential Update: Multiple frames in a batch are used to iteratively estimate motion and structure parameters. This step also updates motion parameters derived from each overlapping batch. Then, these motion parameters are used to compute the globally compatible structure parameters.

This step minimizes iteratively the objective function (30) with respect to iteration variables, S_0, and the interframe rotation parameters for each overlapping batch of frames. Two-view linear solutions, the recognized vanishing lines, and the estimates from the previous batch are used as initial guesses for this nonlinear minimization. The solution which gives the minimum value of the objective function is selected.

After minimizing the objective function for each batch, we update the motion parameters using (35).

At t_l, the ground orientations from t_0 to t_l are easily computed by using $\mathbf{m}(0), \cdots, \mathbf{m}(l-1)$ and the reference ground orientation at t_r, which is the estimated value of the batch from t_r to t_{r+N-1}. This batch is chosen using the criterion that it has the minimum value of the objective function given in (30) among the batches which start at t_0 through t_l. Since translation parameters are easily computed given rotations and surface orientations, we need to update rotation parameters only.

Then we rescale translation and structure parameters starting from the initial frame using any set of the points on the plane.

Step 7: Synthesis: This step synthesizes the input sequence from the image attributes used to obtain the motion and structure estimates as well as from the artificial image attributes that are consistent with the motion and structure estimates but not present in the original image.

Two methods are used in this paper, as described below:

1) The visualization sequence is synthesized by displaying a) those image attributes whose correspondences are used for motion and structure estimation, and b) the vanishing line derived from the estimated surface orientation. Now an attribute may not be present throughout an image sequence because, for example, it may not be detected in each image. Each such attribute is introduced in each image where it is missing. This is done by extrapolating from the nearest frame where it is detected, using the estimated motion and structure values.

2) We use artificial features not present in the scene such as a homogeneous disc pattern. The depiction at t_k shows how the ground will look if it had a unform disc pattern on it and it were viewed at the orientation estimated at t_k during recovery. The sequence of these depictions then comprises a visualization sequence that artificially depicts the estimated motion and structure parameters.

For display, real and/or artificial attributes are shown as a monocular as well as a binocular (stereo) sequence, thus further highlighting the recovered motion and structure parameters.

Step 8: Evaluation of 3-D Analysis: For performance evaluation, we compute alignment error between estimated vanishing lines and the actual vanishing lines, compute image plane differences between observed and 3-D predicted image cues, and perceptually compare the visualization sequence with the original sequence.

First, since the vanishing lines are usually visible for flight images, we compare estimated vanishing lines (surface orientations) with the actual vanishing lines and then measure the image errors defined in (32). Second, we check the average image error of the multiple cues defined in (28). Third, we visually compare the visualization sequence with the original sequence side by side using SUN or SGI monitors. The closer the two sequences are perceived to be, the better the estimates are judged to be.

V. Experimental Results

We conducted experiments to test the performance of the integrated analysis as well as synthesis. Two types of experiments were conducted. First, we evaluated the impact of integration on estimation by applying the algorithms to synthetic images showing known motion and structure and computing estimation errors. Second, we applied the algorithms to real image sequences with unknown ground truth. The performance was evaluated by comparison of the perceived motion and structure from the original and the synthesized image sequences. Sections V-B and V-C describe these experiments. Section V-A first presents some implementation details for the algorithm described in the previous section.

A. Implementation Details

In all the experiments, we use a region detection algorithm based on multiple intensity thresholds. The images are first segmented using multiple thresholds. We experimentally chose three threshold values. This is followed by connected component labeling, small region elimination, and region merging. Then, regions that border on the boundary of the image plane are removed. We detect lines using a modified version of the method described in [15]. To compute the optical flow, we use a method based on grey level correlation values. For each pixel at t_k, we move the window around its neighborhood in the image at t_{k+1} and compute the sum of absolute difference of intensity values. Then, we select the location where the correlation value is at its maximum. We use the flow obtained at the pixels where the image variation is high. In the two real image sequences used in our experiments, it is very difficult to extract the same point features between any two consecutive images. Therefore, we do not use point features for estimation. Then, the matching and segmentation algorithm presented in [14] is applied to the successive pairs of the images.

In the recognition of vanishing line, only lines whose lengths are longer than 170 pixels (TL) are considered. To iteratively minimize (30), we use a modified Levenberg–Marquardt algorithm (IMSL routine *dunlsf*). The batch size N used in our experiments is 3. The values of $\lambda_{k,i,F}$, $\lambda_{k,i,R}$, $\lambda_{k,i,L}$, $\lambda_{k,V}$, and $\lambda_{k,T}$ are 1, 4, 25, 1, and 1, respectively. The values of x_t, y_t, and y_m used in the penalty term for the vanishing lines are set to 0.014, 7 and SIZEV/2 pixels, respectively, where SIZEV is the vertical size of the image plane.

The compression ratio is defined as the ratio of the memory size for the original data to the size for compressed data. (Recall from Section I that compression is with respect to retention of 3-D characteristics, not the usual photometric properties). Based on the premise that the display of the attributes used during analysis will be the most cost effective way of communicating to the observer the same motion and structure characteristics as perceived from the original image sequence, we compute the compression ratio below. We consider only such 3-D scenes as can be approximated by piecewise planar surfaces. Since the amount of data for the camera parameters are negligible, we ignore the camera parameters in computing compression. Assume that one coordinate of a pixel can be represented by 10 b. A region is represented by the average intensity value (1 byte) and its boundary. The boundary is encoded by the Freeman chain code [6] and the coordinates of the starting pixel, where 0.5 bytes and 3 bytes are necessary for one chain code and the coordinates of one pixel, respectively. A line is represented by the pixel coordinates of two end points, and 5 bytes are necessary for

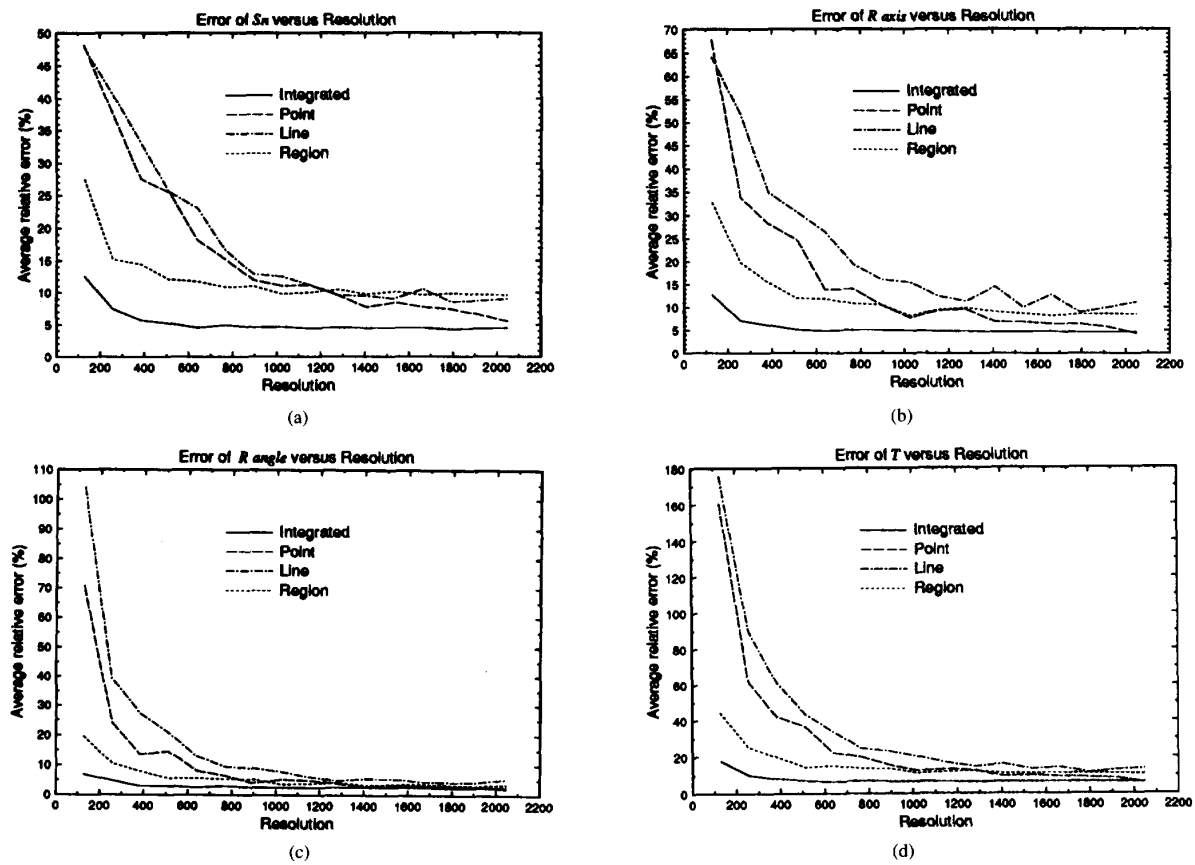


Fig. 5. Performance comparison. Resolutions are varied from 64 × 64 to 2024 × 2024 for a set of actual parameters ($\vec{n}_\omega = [0.5774, 0.5774, 0.5774]'$, $\omega = 4°$ and $\vec{T} = [0.3, 0.3, 0.3]'$). (a) Error of surface normal ($\vec{n}_S$). (b) Error of rotation axis ($\vec{n}_\omega$). (c) Error of rotation angle (ω). (d) Error of translation ($\vec{T}$).

a line. Rotation, translation, and orientation parameters are given by 3, 3, and 2 real numbers (4 bytes per real number), respectively. The optical flow vectors used during analysis are not included since they only describe motion of other points and the flow field can be reconstructed if motion and structure parameters are given. Consider a sequence of K frames where the size of each frame is SIZEV by SIZEH and each pixel is represented by 256 grey levels. Then, we have the following formula for compression ratio:

$$\text{compression ratio} = \frac{\text{SIZEV} \times \text{SIZEH} \times K}{\text{TOL} \times 5 + \text{TOR} \times 4 + \text{LORB} \times 0.5 + 24 \times (K-1) + 8} \tag{36}$$

where TOL and TOR are the total numbers of lines and regions, respectively, and LORB are total length of the chain codes for regions. If a 3-D model for lines and regions is used, the compression ratio can be further reduced.

B. Quantitative Evaluation from Synthetic Images

Experiments were conducted to compare the estimation errors obtained using different cues individually as well as together.

Average Estimation Error from Two Views: Performance of the integrated estimation method using multiple cues (point, line, and region) was compared with the methods that use single cues such as point (Section III-B), line (Section III-B), and region (Section III-B), respectively. In the simulations, the integrated linear two-view estimation method described in Section III-C was used for motion and structure estimation.

The size of the simulated image plane is 1 × 1. Twelve feature correspondences are used for each type of feature (points, lines, and regions). At each trial, a plane passing through the point at (0, 0, 10) is randomly generated. Then, 3-D coordinates of points on the given plane are randomly generated. A line is obtained by a pair of randomly chosen end points, where we consider only those lines whose projected length into the image plane is longer than 0.03. A region boundary is generated by four random variables: two for the center point, one for radius (from 0.01 to 0.08), and one for the number of points (from 16 to 64) on a region boundary. To demonstrate the improvement obtained by using multiple features, points, lines, and regions are generated only in the first, second, and third quadrants of the image plane, respectively. Only features that are within the visual field in both frames are generated. The image coordinates of the points are quantized to the nearest integer for each resolution. The

(a) (b) (c)

Fig. 6. Synthetic image sequence (a) t_0, (b) t_1, (c) t_2.

TABLE I
TRUE VALUES OF SURFACE NORMAL ($\vec{n}_{S,k}$) AT t_k, ROTATION AXIS ($\vec{n}_\omega$), ROTATION ANGLE (ω), AND TRANSLATION ($\vec{T}$) BETWEEN t_k AND t_{k+1}

$[t_k, t_{k+1}]$	[0,1]	[1,2]
$\vec{n}_{S,k}$	[−0.9150, 0.3624, 0.1772]	[−0.9536, 0.2431, 0.1775]
$\vec{n}_\omega$	[0.1222, −0.0368, 0.9918]	[0.1197, −0.0340, 0.9922]
ω (deg)	7.2000	5.4000
$\vec{T}$	[0.0460, −0.0160, 0.2300]	[0.0470, −0.0150, 0.2000]

relative error of a vector is defined by the Euclidean norm of the error vector divided by the Euclidean norm of the correct vector. The surface normal $\vec{n}_S$ is scaled to the unit vector. All errors represent average errors over 50 random trials.

Fig. 5 shows the average relative errors of surface normal $\vec{n}_S$, rotation axis $\vec{n}_\omega$, rotation angle ω, and translation $\vec{T}$, for a set of known motion parameters ($\vec{n}_\omega = [0.5774, 0.5774, 0.5774]'$, $\omega = 4°$, and $\vec{T} = [0.3, 0.3, 0.3]'$). First, we see that the use of multiple features gives robust estimates at all resolutions. For points and lines, the estimates become more accurate as the resolution increases. Regions give more reliable estimates from low to mid resolutions, though the accuracy of the estimates does not improve at higher resolutions. This is expected, since the approximations are made under the three assumptions when the region-based equations are derived. (It is not difficult to derive the exact nonlinear method for regions.) The good performance of regions at lower resolutions is due to the robustness of lower order moments used in (19) and (20) with respect to quantization of region boundaries. (Stable detection of region boundaries may be difficult in some images.) Increasing the number of features gives better estimates. We note here that estimation of translation is the most noise-sensitive, which is also observed for the nonplanar case in [9].

From these simulations, we see that integrated estimation can increase the accuracy of the resulting estimates because the detected features are large in number and spatially better distributed, if there are no outliers present. Note that these simulations are based on the assumption of perfect extraction and matching of features up to the quantization errors.

Estimation Errors from Multiple Views: These simulations use three frames of a synthetic sequence as shown in Fig. 6. The first and second frames are synthesized from the third frame using bilinear interpolation. The focal length is 8 mm. The field of view of the camera is 40° by 34.4°, corresponding to the image resolution of 560 by 480. The true values of motion and structure parameters are shown in Table I.

We first detect points, lines, and regions. Then the matching and segmentation algorithm is applied to two pairs of consecutive images. The numbers of matched features for points, lines, and regions in the ground segment are 63, 59, and 38 between t_0 and t_1, and 64, 78, and 33 between t_1 and t_2, respectively.

TABLE II
ESTIMATES OF SURFACE NORMAL, ROTATION AXIS, ROTATION ANGLE, AND TRANSLATION FROM THE INTEGRATED NONLINEAR BATCH ALGORITHM

$[t_k, t_{k+1}]$	[0, 1]	[1, 2]
$\vec{n}_{S,k}$	[−0.9292, 0.3274, 0.1716]	[−0.9632, 0.2073, 0.1711]
$\vec{n}_\omega$	[0.1218, −0.0385, 0.9918]	[0.1230, −0.0356, 0.9918]
ω (deg)	7.1638	5.3217
$\vec{T}$	[0.0476, −0.0160, 0.2243]	[0.0495, −0.01639, 0.1953]

In Table II, we show the estimates that result from the integrated nonlinear batch algorithm using multiple cues (point, line, and region) described in Section III-C. The estimates can be seen to be good.

In Tables III and IV we show percentage errors in the estimates derived using the integrated approach against those derived using individual cues, for both linear two-view estimation and nonlinear batch estimation.

The following observations can be made from these results. First, the nonlinear batch method using multiple features and frames in Section III-C yields satisfactory estimates despite the fact that outliers are present (in this case, points between t_0 and t_1). The point outliers are caused by the difficulty of extracting and matching the same point features between t_0 and t_1. This experiment shows clearly the nonlinear integrated method's capability of reducing the effect of outliers by using the large number of available features and the structure consistency constraint. If the ultimate accuracy is important, an existing robust regression method [18] can be used for minimization of the image errors in (30) at the expense of increased computational cost. Further, the results demonstrate that while the estimates derived by integration are not always more accurate than those based on the best features, there are features that lead to large errors because the features have outliers. Since it is not known *a priori* which features are the most reliable or error prone in a given scenario, integration gives robust results without using scene-specific information. Individual features may happen to be configured so as to yield better estimates than integrated analysis, but it is not known that this is the case and, if so, from which features. Integration therefore provides estimates whose variance computed over many scenes is smaller than if individual features were used separately.

Second, the nonlinear batch methods usually give more robust estimates than the linear two-view solutions, especially for noisy images (for example, t_0 and t_1). Third, the penalty term $E_{k,V}$ in (30) for the candidates of a vanishing line increases the robustness of the estimates. This term is useful especially for the noisy images if the candidates of the vanishing line are available. The term $E_{k,V}$ is better constrained by d_1 and d_2 (see Fig. 3) than by the error of the surface normal vector. For example, the nonlinear estimation error for $\vec{n}_{S,0}$ in Table III is 3.82%, which is relatively low. However, d_1 and d_2 are 4.74 and 18.31 pixels, respectively. This shows that the reliable estimation of a vanishing line in the image plane is difficult. Fourth, in general, the image error decreases if the accuracy of

TABLE III
PERCENTAGE ESTIMATION ERRORS IN SURFACE NORMAL ($\vec{n}_{S,0}$), ROTATION AXIS ($\vec{n}_\omega$), ROTATION ANGLE (ω), TRANSLATION ($\vec{T}$) BETWEEN t_0 AND t_1 FOR VARIOUS METHODS

	Linear two-view				Nonlinear Batch			
Error in %	$\vec{n}_{S,0}$	$\vec{n}_\omega$	ω	$\vec{T}$	$\vec{n}_{S,0}$	$\vec{n}_\omega$	ω	$\vec{T}$
Point	57.13	2.30	8.14	23.39	0.41	3.75	23.57	88.41
Line	3.00	0.18	0.27	1.29	0.24	0.13	0.14	0.98
Region	4.39	0.56	2.09	10.48	0.94	0.52	3.08	9.10
Integrated	41.26	1.34	6.23	23.65	3.82	0.18	0.50	2.51

TABLE IV
PERCENTAGE ESTIMATION ERRORS IN SURFACE NORMAL ($\vec{n}_{S,1}$), ROTATION AXIS ($\vec{n}_\omega$), ROTATION ANGLE (ω), TRANSLATION ($\vec{T}$) BETWEEN t_1 AND t_2 FOR VARIOUS METHODS

	Linear two-view				Nonlinear Batch			
Error in %	$\vec{n}_{S,1}$	$\vec{n}_\omega$	ω	$\vec{T}$	$\vec{n}_{S,1}$	$\vec{n}_\omega$	ω	$\vec{T}$
Point	1.73	0.47	0.46	2.40	3.05	0.67	0.49	1.70
Line	1.80	0.38	0.72	1.95	0.23	0.30	0.36	1.26
Region	3.99	0.48	2.29	9.60	1.19	0.30	2.59	8.63
Integrated	0.34	0.07	0.34	3.19	3.76	0.37	1.45	2.65

the estimates increases. Therefore, the image error is a good quantitative criterion for evaluating the estimates when the actual values are not available. (Experiments with real images demonstrate this.) The average image error $\overline{E_{0,2,I}}$ in (28) is 1.09, which is acceptable. If we do not use point features, we could obtain lower average image error.

C. Perceptual Evaluation from Real Images

We conducted experiments with two real image sequences.

Desert Sequence: We derived a sequence of 29 frames from a commercially available VHS videotape of a film shot from a flying aircraft. The focal length was assumed to be 1 mm. The digitization was done with a resolution of 600 by 464. In Fig. 7(a) and (b), we show two frames at t_5 and t_6. Since the commercial VHS tape is far from having the quality of the master tape, digitized images are very noisy. There is also blurring of images since they were taken from a camera mounted on a flying aircraft.

Next, we extract regions, lines, and flow as image cues. Examples of these detected features at t_5 and t_6 are shown in Fig. 7(c) through (g). Flow vectors are used only at those locations where a point feature detector responds.

The result of segmentation, matching, and merging for two frames are shown in Fig. 7(h) and (i). This is the result of integrated interpretation. The attributes shown here are diverse but mutually compatible cues. Note that the parts corresponding to the sky and bottom of the aircraft were successfully segmented out. Vanishing lines were successfully recognized at all frames.

The estimated values of motion parameters shows that the camera on the aircraft moves in the direction of the optical axis over the ground. For this image sequence, the vanishing line was an important cue for reliable estimates of the ground orientation. Estimated vanishing lines (surface orientations) are in good agreement with the actual vanishing lines in the image plane. The average image error $\overline{E_{0,28,I}}$ computed by (28) is 1.7709 pixels. Although the error is larger than one pixel, it is not bad if we consider the poor image quality. The original sequence and the resulting visualization sequence by using the first method of Step 7 of the algorithm are presented in Fig. 8(a) and (b) for t_0, t_{14}, and t_{28}. In Fig. 8(c), synthetic discs are used to enhance the perception of motion and structure, as explained in the second method. If we watch the two visualization sequences as they are played on a SUN workstation monitor, we perceive the same motion and structure from them in an informal viewing as from the original image sequence.[1] We also display the synthesized sequence in stereo on a SGI monitor. The compression ratio achieved by using (36) is 502.

Runway Sequence: We derived a sequence of 34 frames from a commercially available CAV laserdisc of a film shot from a flying aircraft. The focal length was assumed to be 8 mm. The digitization was done with a resolution of 640 $\times$ 480. This is a challenging sequence to our algorithm since the images contain partially or completely occluded vanishing lines and there is reflection of the ground on the bottom of the airplane. The quality of the images is a little better than the desert sequence images obtained from a VHS tape. In Fig. 9(a), we show two frames at t_4 and t_5.

Next, we extract regions, lines, and flow as image cues. Examples of these detected features at t_4 and t_5 are shown in Fig. 9(c) through (g). Flow vectors are used at only those locations where a point feature detector responds.

The result of segmentation, matching and merging for two frames are shown in Fig. 9(h) and (i). We can see the parts corresponding to the bottom of the airplane are successfully segmented out. Vanishing lines were not identified at $t = (4, 6, 10, 11, 12, 14, 15, 18, 22, 23, 24, 26, 27)$ due to occlusion and several candidates from the actual vanishing line and the bottom parts of the airplane, as we can see in Fig. 10(a).

The estimated motion parameters show the camera on the aircraft moves in the direction of the optical axis over the

[1] A videotape showing the original image sequence and the visualization sequences is available.

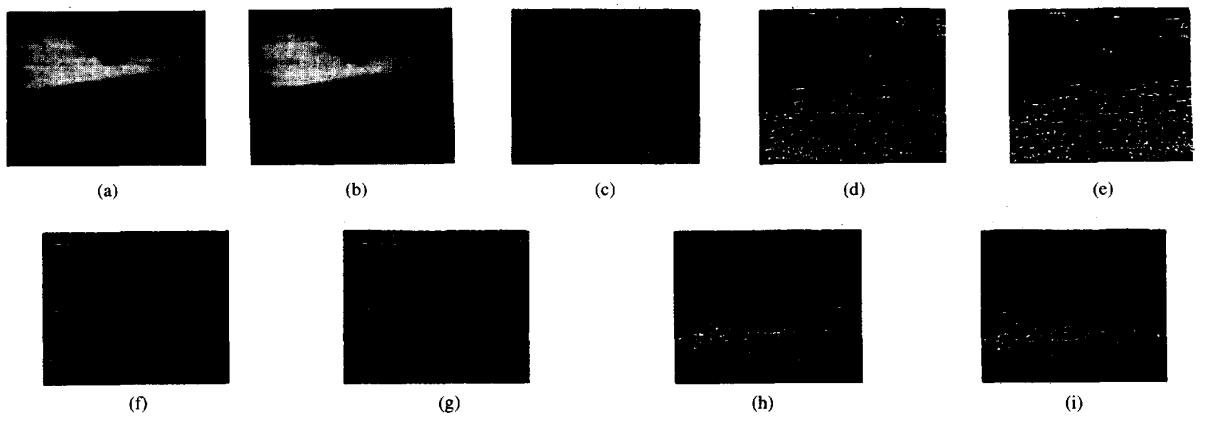


Fig. 7. Extracted features and segmentation results at t_5 and t_6 for the desert sequence. (a) Input image at t_5. (b) Input image at t_6. (c) Computed flow between t_5 and t_6. (d) Extracted regions at t_5. (e) Extracted regions at t_6. (f) Extracted lines at t_5. (g) Extracted lines at t_6. (h) Segmentation result at t_5. (i) Segmentation result at t_6.

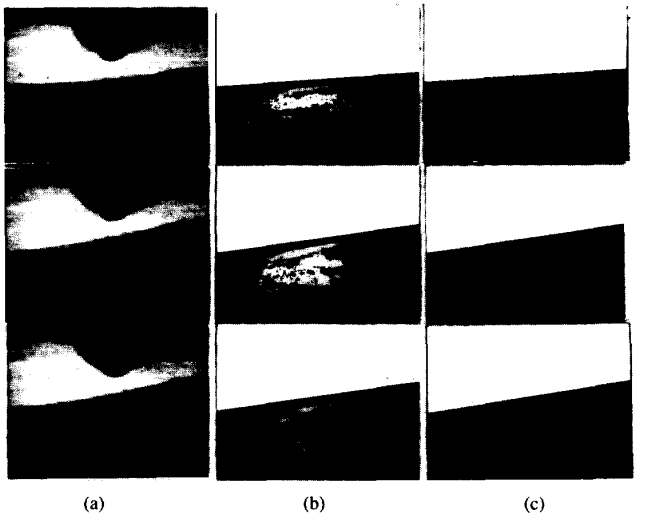


Fig. 8. Experiments with the desert sequence: (a) Input image sequence. (b) Visualization image sequence. (c) Visualization image sequence with synthetic pattern.

ground. For this image sequence, lines were an important cue for reliable estimation since several good line features exist in the image sequence. Estimated vanishing lines are in a good agreement with the actual vanishing lines in the image plane. The average image error $\overline{E_{0,33,I}}$ computed by (28) is 0.858679 pixels. The error is less than one pixel, which is satisfactory. This value is lower compared to what we achieved for the desert sequence since more cues are used during analysis. The original sequence and the resulting visualization sequence using the first method of Step 7 of the algorithm are presented in Fig. 10(a) and (b) for t_0, t_{16}, and t_{32}. If we watch the visualization sequences as they are played on a SUN workstation monitor, we perceive the same motion and structure from them in an informal viewing as in the original image sequence[1]. We also display the synthesized sequence in stereo on a SGI monitor. The compression ratio achieved by using (36) is 367.

VI. Conclusion and Extensions

We have presented an approach for motion and structure estimation from a monocular sequence of images that makes integrated use of multiple image cues such as points, lines, regions, and optical flow for piecewise planar surface. To increase the reliability of the result further, we used a sequential-batch method to compute motion and plane orientation. In a batch minimization, the objective function links multiple frames through the unit surface normal, yielding two results simultaneously: First, it enforces the structure consistency, and second, it avoids the problem of unknown scale, thus reducing the iteration space. The iteration space is further reduced using motion and structure relationships. Note that tracking of the features is not necessary to enforce the structure consistency for planar surface. According to the experiments we conducted, the integrated approach using

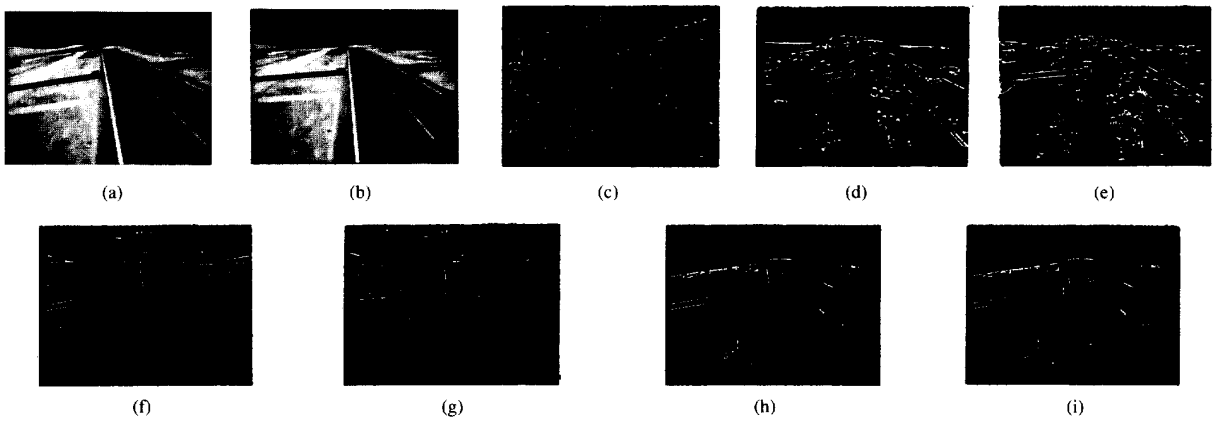


Fig. 9. Extracted features and segmentation results at t_4 andt_5 for the runway sequence. (a) Input image at t_4. (b) Input image at t_5. (c) Computed flow between t_4 and t_5. (d) Extracted regions at t_4. (e) Extracted regions at t_5. (f) Extracted lines at t_4. (g) Extracted lines at t_5. (h) Segmentation result at t_4. (i) Segmentation result at t_5.

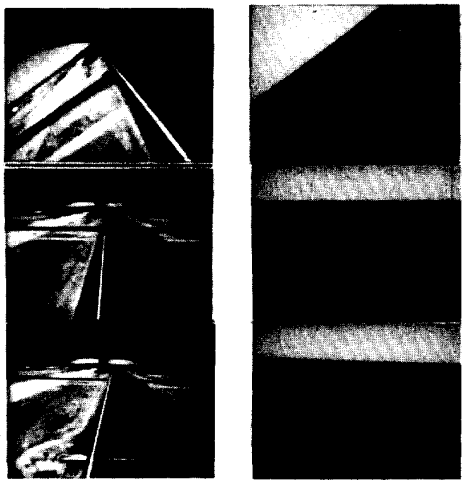


Fig. 10. Experiments with the runway sequence: (a) Input image sequence. (b) Visualization image sequence.

multiple frames gives satisfactory results even with outliers present. If precision is important, we can use one of the existing robust regression methods [18] at the expense of increased computational cost while still following the basic approach presented.

We used the intermediate results of the estimation process to synthesize the original image sequence in two ways. We also used the vanishing line, which can be seen in real flight images. Performance evaluation was done by conducting experiments with one synthetic and two real image sequences to demonstrate the feasibility of our approach.

Texture gradient was not present as a useful cue in the two image sequences we used in our experiments. We plan to experiment with images that have textures. We also plan to extend this integrated approach to image sequences that contain both planar and nonplanar surfaces.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their valuable suggestions. Also, we thank Dr. Y. Liu, X. Hu, S. Thirumalai, D. Hougen, and T. Cho for letting us use their programs for feature detection.

REFERENCES

[1] K. Kanatani, "Detecting the motion of a planar surface by line and surface integrals," *Comput. Vision, Graphics, Image Processing*, vol. 29, pp. 13–22, 1985.

[2] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, no. 4, July 1985.

[3] M. Subbarao and A. Waxman, "Closed form solutions to image flow equations for planar surfaces in motion," *Comput. Vision, Graphics, Image Processing*, vol. 36, pp. 208–228, 1986.

[4] R. Y. Tsai and T. S. Huang, "Estimating three dimensional motion parameters of a rigid planar patch, II: Singular value decomposition," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-30, no. 4, pp. 525–534, Aug. 1982.

[5] J. Rolfe and K. Staples, *Flight Simulation*. Cambridge, England: Cambridge University Press, 1986.

[6] T. Pavlidis, *Structural Pattern Recognition*. Berlin: Springer-Verlag, 1977.

[7] D. Hearn and M. Baker, *Computer Graphics*. London: Prentice-Hall International, 1986.

[8] J. Weng, N. Ahuja, and T. S. Huang, "Motion and structure from point correspondences: A robust algorithm for planar case with error estimation," in *Proc. Int. Conf. Pattern Recognition*, Rome, pp. 247–251, 1988.

[9] J. Weng, T. S. Huang, and N. Ahuja, "Motion and structure from two perspective views: Algorithms, error analysis, and error estimation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-11, no. 5, pp. 451–476, May 1989.

[10] N. Cui, J. Weng, and P. Cohen, "Extended structure and motion analysis from monocular image sequences," in *Proc. 3rd Int. Conf. on Computer Vision*, Osaka, Japan, pp. 222–229, 1990.

[11] R. Kumar, A. Tirumalai, and R. Jain, "A nonlinear optimization algorithm for the estimation of structure and motion parameters," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, San Diego, CA, pp. 136–143, 1989.

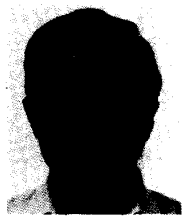
[12] S. Sull and N. Ahuja, "Segmentation, matching and estimation of structure and motion of textured piecewise planar surfaces," in *Proc. IEEE Workshop on Visual Motion*, Princeton, NJ, pp. 274–279, Oct. 1991.

[13] ——, "Integrated 3-D recovery and visualization of flight image sequences," in *Proc. Image Understanding Workshop*, DARPA, pp. 21–28, Jan. 1992.

[14] S. Sull, "Integrated 3-D analysis and analysis-guided syntheses," Ph.D dissertation, University of Illinois, Urbana-Champaign, 1993.
[15] Y. Liu and T. Huang, "Determining straight line correspondences from intensity images," *Pattern Recogn.*, vol. 24, pp. 489–504, 1991.
[16] ——, "Estimation of rigid body motion using straight line correspondences," *Comput. Vision, Graphics, Image Processing*, vol. 43, no. 1, pp. 37–52, Jul. 1988.
[17] D. Blostein and N. Ahuja, "Shape from texture: Integrating texture-element extraction and surface estimation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 12, pp. 1233–1250, Dec. 1989.
[18] P. Meer, D. Mints, A. Rosenfeld, and D. Kim, "Robust regression methods for computer vision: A review," *Int. J. Comput. Vision*, vol. 6, no. 1, pp. 59–70, 1991.

Sanghoon Sull (S'78–M'93) was born in Seoul, Korea, on September 22, 1958. He received the B.S. degree with honors in electronics engineering from the Seoul National University, Seoul, Korea, in 1981, the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, Korea, in 1983, and the Ph.D. degree in electrical and computer engineering from the University of Illinois, Urbana-Champaign, in 1993.

From March 1983 to July 1986, he worked for the Korea Broadcasting Systems as a Research and Development Engineer. From October 1986 to December 1987, he worked at the Center for Robotic Systems in Microelectronics, University of California, Santa Barbara, as a Research Assistant. From January 1988 to January 1993, he was a Research Assistant at the Coordinated Science Laboratory and the Beckman Institute, University of Illinois, Urbana-Champaign. From January 1993 to February 1994, he was a postdoctoral Research Associate at the Beckman Institute. Since March 1994 he has been a National Research Council Research Associate at NASA, Ames Research Center, Moffet Field, CA. His current research interests include computer vision, graphics, and image processing.



Narendra Ahuja (S'79–M'79–SM'85–F'92) received the B.E. degree with honors in electronics engineering from the Birla Institute of Technology and Science, Pilani, India, in 1972, the M.E. degree with distinction in electrical communication engineering from the Indian Institute of Science, Bangalore, India, in 1974, and the Ph.D. degree in computer science from the University of Maryland, College Park, in 1979.

From 1974 to 1975 he was Scientific Officer in the Department of Electronics, Government of India, New Delhi. From 1975 to 1979 he was at the Computer Vision Laboratory, University of Maryland, College Park. Since 1979 he has been with the University of Illinois, Urbana-Champaign, where he is currently (1988–) a Professor in the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory, and the Beckman Institute. His interests are in computer vision, robotics, image processing, image synthesis, and parallel algorithms. He has been involved in teaching, research, consulting, and organizing conferences in these areas. His current research emphasizes integrated use of multiple image sources of scene information to construct three-dimensional descriptions of scenes, the use of integrated image analysis for realistic image synthesis, the use of the acquired three-dimensional information for navigation, and multiprocessor architectures for computer vision.

Dr. Ahuja was selected as a Beckman Associate in the University of Illinois Center for Advanced Study for 1990–91. He received University Scholar Award (1985), Presidential Young Investigator Award (1984), National Scholarship (1967–72), and President's Merit Award (1966). He has coauthored the books *Pattern Models* (Wiley, 1983) with Bruce Schachter, and *Motion and Structure from Image Sequences* (Springer-Verlag, to appear) with Juyang Weng and Thomas Huang. He has been Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *Computer Vision, Graphics, and Image Processing*, *Journal of Mathematical Imaging and Vision*, and *Journal of Information Science and Technology*. He is a fellow of the American Association for Artificial Intelligence, and a member of the Association for Computing Machinery and the Optical Society of America.