Structure and Motion from Line Segments in Multiple Images

Camillo J. Taylor, David J. Kriegman

Abstract— **This paper presents a new method for recovering the three dimensional structure of a scene composed of straight line segments using the image data obtained from a moving camera. The recovery algorithm is formulated in terms of an objective function which measures the total squared distance in the image plane between the observed edge segments and the projections (perspective) of the reconstructed lines. This objective function is minimized with respect to the line parameters and the camera positions to obtain an estimate for the structure of the scene. The effectiveness of this approach is demonstrated quantitatively through extensive simulations and qualitatively with actual image sequences. The implementation is being made publicly available.**

Keywords— **Structure from motion, straight lines, three-dimensional reconstruction, perspective projection, numerical minimization.**

I. Introduction

Many applications such as vehicle navigation, robot motion planning, object recognition and automatic acquisition of CAD models and architectural drawings involve recovering a representation for the geometric structure of the world from sensor data. This paper presents a new method for recovering the three dimensional structure of a scene composed of straight line segments using the image data obtained from a moving camera.

Since first being considered by Ullman [1], the *structure from motion* (SFM) problem has received quite a bit of attention, and a number of algorithms have emerged. These algorithms differ in the type of input (e.g. dense gray scale images, feature points, lines, etc.), the number of required images (e.g. two, three, unlimited), the camera model (e.g. perspective or orthographic projection), and the assumed knowledge of the camera motion (e.g. completely known, constant velocity, unknown velocity, etc.).

Several researchers have looked at the problem of recovering the structure of a scene composed of point features from a set of images. Longuet-Higgins and others [2], [3], [4], [5], [6] have shown that the structure of a set of points can be recovered from two images even when the relative position of the two cameras is unknown. Other researchers have developed algorithms that use multiple images of a rigid scene to produce a more accurate reconstruction, and these fall into two classes: batch algorithms, which process all of the data simultaneously, and incremental algorithms, which maintain some notion of state that is updated with each new sensor measurement. Several incremental structure from motion algorithms based on the Extended Kalman Filter (EKF) have been proposed in the literature [7], [8], [9], [10].

Most batch algorithms formulate the SFM problem in terms of a nonlinear objective function that is then minimized to obtain an estimate for the structure of the environment. Since these algorithms avoid the linearizing assumptions inherent in the EKF they usually yield more accurate reconstruction results [6], [4]. Tomasi and Kanade [11] demonstrated a batch algorithm that can be used to reconstruct a set of points viewed under orthographic projection from a sequence of camera positions. By modelling orthographic projection as a linear map in baricentric coordinates, they showed how singular value decomposition could be used to minimize an objective function which measures the disparity between the projections of the reconstructed points and the measured point features in the images. Szeliski and Kang [12] and Hu and Ahuja [4] have also obtained good results by applying nonlinear minimization algorithms to point based SFM problems.

A few researchers have considered the problem of reconstructing scenes composed of straight line segments. Straight line features are prominent in most man-made environments, they can be detected and tracked relatively easily in image data, and they provide a great deal of information about the structure of the scene. Additionally, since edge features have more image support than point features, they can be localized more accurately. These properties make them ideal candidates for use in structure from motion algorithms.

Several approaches based on the Extended Kalman Filter (EKF) have been proposed. Faugeras et. al. [8] presented a method that solves for the motion of a camera and the structure of the scene from three perspective images. They applied an Extended Kalman Filter to minimize an objective function which represented a version of the epipolar constraint. Crowley et. al. [7] describe a method for reconstructing a rigid scene in which an EKF was employed to update the estimate for the structural parameters using the measurements obtained from a moving camera. In this work, the absolute position of the moving camera is known accurately. Both Jezouin and Ayache [9] and Vieville and Faugeras [10] have developed EKF based techniques that update the estimates for the scene structure and the camera position using the measurements obtained from a tracking

Camillo Taylor is currently with the Department of Electrical Engineering and Computer Science at U.C. Berkeley, Berkeley, CA 94720-1770. E-mail:camillo@robotics.eecs.berkeley.edu

David J. Kriegman is an Associate Professor with the Center for Systems Science in the Department of Electrical Engineering at Yale University, New Haven, CT 06520-8267. E-mail: kriegman@yale.edu

This research was performed while both authors were with the Center for Systems Science at Yale University.

Support for this work has been provided by a gift from the INMOS division of SGS-Thomson and by grants from the National Science Foundation, DDM-9112458 and NYI IRI-9257990.

system which monitors the positions of point and line features in the image stream.

Vieville [13] and Giai-Checa and Vieville [14] have investigated the feasibility of using image flow information to recover the geometry of a scene composed of straight line features. They relate the velocity of the edge features in the image to the line parameters and the camera motion; algorithms are presented for reconstructing a rigid scene using the information obtained from a tracking system. Since this problem is underconstrained additional assumptions are required in order to obtain a solution. Vieville assumes that the camera is moving with constant rotational and translational velocity while Giai-Checa et. al. explicitly search for junctions in the image and then use these intersections to induce extra constraints on the solution. Navab and Deriche [15] have developed a novel method for recovering the 3D position and velocity of the linear features from the image velocities and edge measurements obtained from a calibrated stereo pair.

Another approach to this structure from motion problem was inspired by the linear algorithm developed by Longuet-Higgins for point features [2]. Given a set of at least 13 linear features viewed in three frames, it is possible to derive a set of equations which represent an analog to the epipolar constraint that Longuet-Higgins exploited in his work. Liu and Huang [16], Spetsakis and Aloimonos [17], [18] and Faugeras [19] have all developed algorithms that recover the structure of the scene and the positions of the cameras based on these equations. One obvious advantage of these linear algorithms is that they provide a closed form solution. In practice, however, these methods tend to be very sensitive to errors in the image measurements. Section V presents a series of simulation experiments that compare the algorithm presented in this paper to the three frame linear algorithm proposed in [16]. These experiments demonstrate that our algorithm is much more robust to image noise than the linear methods and can be expected to provide much more accurate results.

In this paper, the reconstruction problem is formulated in terms of an objective function which measures the total squared distance in the image plane between the observed edge segments and the projections (perspective) of the reconstructed lines. This objective function is minimized with respect to the line parameters and camera positions in order to obtain an estimate for the structure of the scene. A minimum of six edge correspondences in three images is required to solve this structure from motion problem [8], but the algorithm can take advantage of any number of additional images or straight line features that may be available. To the best of our knowledge, this algorithm produces the most accurate results ever achieved on this type of structure from motion problem. In the special case where the line segments are vertical and the camera motion is confined to the horizontal plane, the presented algorithm degenerates to the least squares approach to recovering planar point locations and motion presented in [20].

Weng Huang and Ahuja [6] also describe an approach to this problem based on minimizing a non-linear objective function. The objective function that they propose, however, measures the Mahalanobis distance between the parameterization of the observed line and that of the predicted line. This objective function differs from the one advocated in this paper since it does not directly measure the retinal disparity in the image.

It has often been argued that a point based SFM algorithm could be used to directly recover the 3D coordinates of the endpoints of the line segments. Unfortunately, in practice it is very difficult to precisely locate the endpoints of an image edge for a number of reasons. Edge detection algorithms such as Canny's [21] are not designed to find junctions, and they often fail to accurately locate an isolated endpoint of a line. Various parts of the edge including the endpoints may be occluded in the images. The algorithm proposed in this paper avoids all of these problems by reconstructing the infinite straight line that supports the observed edge segments rather than the endpoints of the line. Consequently, the algorithm can be used even when multiple edges in a single image correspond to different portions of the same 3D line.

II. Problem Formulation

In this paper, the structure from motion problem is formulated in terms of an objective function $\mathcal{O}$ that measures the disparity between the actual image measurements and the image measurements that would be predicted from the recovered 3D structure and camera locations. This section describes how the objective function is constructed.

The image formation process can be modeled as a function $\mathcal{F}$ which takes the position and orientation of a camera $\mathbf{q}$ and the position of the three-dimensional line $\mathbf{p}$ and returns a representation for the corresponding edge in the image. Consider the case where there are m images of a static scene containing n straight line features, let $\mathbf{u}_{ij}$ be the measurement of the projection of feature i in image j. Let $Error()$ be a positive real valued function that measures the disparity between the observed image feature $\mathbf{u}_{ij}$ and the expected image feature $\mathcal{F}(\mathbf{p}_i, \mathbf{q}_j)$. The objective function $\mathcal{O}$ can now be defined as follows:

$$\mathcal{O} = \sum_{j=1}^{m} \sum_{i=1}^{n} Error(\mathcal{F}(\mathbf{p}_i, \mathbf{q}_j), \mathbf{u}_{ij}) \qquad (1)$$

The aim of the structure from motion algorithm is to find a choice of parameters, $\mathbf{p}_i$ and $\mathbf{q}_j$, that is most consistent with the image measurements by minimizing the objective function $\mathcal{O}$ with respect to these parameters using the techniques described in section III. In the remainder of this section, the elements of (1) are discussed in more detail.

The camera positions $\mathbf{q}_j$ are represented in the usual manner; by a translation vector, $\mathbf{t}_j \in \mathbb{R}^3$, and a rotation matrix, $R_j \in SO(3)$. These parameters represent the position and orientation of the camera with respect to some base frame of reference. In the sequel we will arbitrarily define the base frame of reference to be the first camera position in the sequence.

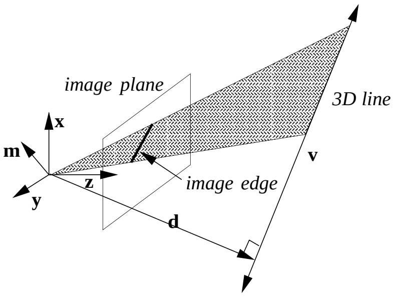

Fig. 1. A 3D line can be represented with respect to a coordinate system by two vectors $\hat{\mathbf{v}}$ and $\mathbf{d}$.

A. The Geometry of Straight Lines

Fig. 1 shows how a straight line can be represented in terms of a unit vector $\hat{\mathbf{v}}$ which indicates the direction of the line, and a vector $\mathbf{d}$ which designates the point on the line that is closest to the origin. In other words, we can represent a straight line by a tuple $\langle \hat{\mathbf{v}}, \mathbf{d} \rangle$ where $\hat{\mathbf{v}}^T\hat{\mathbf{v}} = 1$ and $\hat{\mathbf{v}}^T\mathbf{d} = 0$.[1] This set of tuples defines an algebraic set which is a 4 dimensional manifold embedded in $\mathbb{R}^6$. A particular straight line can be thought of as a point on this manifold.[2]

As shown in Fig. 1, the line and the camera center define a plane whose normal is denoted by the vector $\mathbf{m}$. Notice that under perspective projection the image of the line corresponds to the intersection of this plane and the image plane. Assuming unit focal length, the edge in the image will be defined by the equation:

$$m_x x + m_y y + m_z = 0 \tag{2}$$

where $\mathbf{m} = (m_x, m_y, m_z)^T$.

A measured edge segment in an image $\mathbf{u}_{ij}$ is represented by its endpoints, (x_1, y_1) and (x_2, y_2). As we noted in the introduction, the endpoints of these edges do not necessarily correspond to the endpoints of the three-dimensional line segments.

B. The Projection Function, $\mathcal{F}$

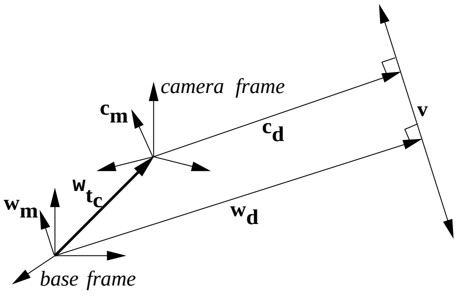

Fig. 2. The frames and vectors associated with two views of a line.

Fig. 2 shows a single infinite straight line viewed from two different positions. The rigid transformation between the base reference frame and the camera frame is defined by a translation vector ${}^w\mathbf{t}_c$ and a rotation matrix ${}^c_w\mathbf{R}$.[3] Given these definitions, we can write the following equations that relate the vectors representing the line in the camera frame (${}^c\hat{\mathbf{v}}$, ${}^c\hat{\mathbf{m}}$, ${}^c\mathbf{d}$) to the vectors representing the same line in the base frame (${}^w\hat{\mathbf{v}}$, ${}^w\hat{\mathbf{m}}$, ${}^w\mathbf{d}$).

$$\begin{aligned}
{}^c\hat{\mathbf{v}} &= {}^c_w\mathbf{R}\ {}^w\hat{\mathbf{v}} \\
{}^c\mathbf{d} &= {}^c_w\mathbf{R}\,(\,{}^w\mathbf{d} - {}^w\mathbf{t}_c + (\,{}^w\mathbf{t}_c \cdot {}^w\hat{\mathbf{v}}\,)\,{}^w\hat{\mathbf{v}}\,) \\
{}^c\mathbf{m} &= {}^c\hat{\mathbf{v}} \times {}^c\mathbf{d} \\
&= {}^c_w\mathbf{R}\,\{\,{}^w\hat{\mathbf{v}} \times (\,{}^w\mathbf{d} - {}^w\mathbf{t}_c\,)\} \\
{}^c\hat{\mathbf{m}} &= {}^c\mathbf{m}\,/||\,{}^c\mathbf{m}\,||
\end{aligned} \tag{3}$$

Equation 3 is particularly interesting since it determines how the normal vector $\mathbf{m}$ varies as a function of the camera position for a particular line. As shown in (2), the normal vector $\mathbf{m}$ determines where the line will appear in the image, so (3) actually represents the projection function, $\mathcal{F}$.

C. The Error Function, Error

Fig. 3 shows a typical situation in the image plane. The heavy line in this figure represents the observed edge segment $\mathbf{u}_{ij}$ while the lighter line represents the edge predicted from $\mathcal{F}(\mathbf{p}_i, \mathbf{q}_j)$.

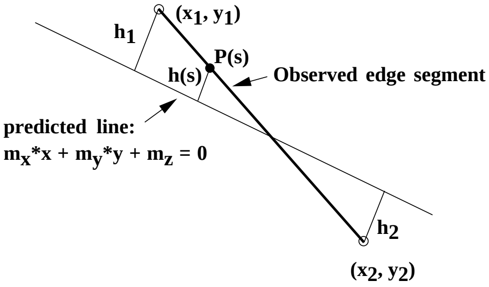

Fig. 3. The heavy line in this figure represents the observed edge segment in the image while the lighter line represents the predicted edge.

Points on the observed edge segment can be parameterized by a single scalar variable $s \in [0, l]$ where l is the length of the edge, $l = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. Let $h(s)$ be a function which measures the shortest distance from a point on the segment, $p(s)$, to the predicted edge as shown in Fig. 3.

$$h(s) = h_1 + s\frac{h_2 - h_1}{l} \tag{4}$$

where the scalar values h_1 and h_2 in (4) represent the shortest distances from the endpoints of the edge segment to the predicted edge, and are given by:

$$\begin{aligned}
h_1 &= \frac{m_x x_1 + m_y y_1 + m_z}{\sqrt{m_x^2 + m_y^2}} \\
h_2 &= \frac{m_x x_2 + m_y y_2 + m_z}{\sqrt{m_x^2 + m_y^2}}
\end{aligned} \tag{5}$$

[1] We could also have chosen to denote the lines by their Plücker coordinates $\langle \hat{\mathbf{v}}, \mathbf{M} \rangle$ where $\mathbf{M} = \hat{\mathbf{v}} \times \mathbf{d}$.

[2] A careful reader will notice that there is actually a two to one correspondence between points on this manifold and the set of infinite straight lines since $\langle \hat{\mathbf{v}}, \mathbf{d} \rangle$ and $\langle -\hat{\mathbf{v}}, \mathbf{d} \rangle$ denote the same line.

[3] To represent the coordinates of a vector, we follow the notation established by Craig [22]; the leading superscript indicates the frame in which the coordinates are expressed. Premultiplying the coordinates of a vector written in frame w by a rotation matrix ${}^c_w\mathbf{R}$ yields the coordinates in frame c.

With these definitions in place, we can define the total error between the observed edge segment and the predicted edge as:

$$\begin{aligned} Error &= \int_0^l h^2(s)ds = \frac{l}{3}(h_1^2 + h_1 h_2 + h_2^2) \\ &= \mathbf{m}^T(A^T B A)\mathbf{m} \end{aligned} \qquad (6)$$

where: $A = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{pmatrix}$, $B = \frac{l}{3(m_x^2+m_y^2)} \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$.

Although the projection of the recovered line onto the image plane has infinite extent, only the portion nearest to the observed image edge contributes to the error measure. This can be contrasted with other approaches where the observed edge is treated as though it were infinite [8].

It should also be noted that in this formulation of the error function, the contributions from various edges are explicitly weighted by their lengths, l. This is a desirable property since longer edges can be localized more accurately in the image than shorter ones. If a line does not appear in a particular image, this error term is set to zero. Note that in order to reconstruct a particular straight line, it must appear in at least two images in the sequence.

If each edge was viewed as a collection of edgels (x_i, y_i) rather than as a continuous line segment, we could measure the disparity between the expected edge and the observed edgels by summing the squared distances between these points and our reprojected line, $\mathbf{m} = (m_x, m_y, m_z)$. The resulting error function could be expressed as follows: $Error = \mathbf{m}^T A \mathbf{m}/(m_x^2 + m_y^2)$ where $A = \sum_i (x_i, y_i, 1)(x_i, y_i, 1)^T$. This error function has exactly the same form as the one given in equation (6) so the same minimization algorithm could be used to recover the unknown parameters.

III. Recovery Algorithm

To obtain an estimate for the structure of the scene and the motion of the camera, the objective function $\mathcal{O}$, described in the previous section, must be minimized. Like most non-linear cost functions $\mathcal{O}$ can exhibit multiple local minima. We propose to use a hybrid optimization technique that generates a series of random initial estimates for the unknown parameters; these estimates are then used as starting points for a gradient descent minimization procedure that locates local minima of the cost function $\mathcal{O}$. In this implementation, the global minimization algorithm only needs to generate initial estimates for the camera orientations, R_j, since initial estimates for the other parameters $(\mathbf{t}_j, \hat{\mathbf{v}}_i, \mathbf{d}_i)$ are obtained as part of the local minimization algorithm described in section III-B. The effectiveness of this global minimization strategy will be demonstrated experimentally in section IV.

A. Global Minimization

The basic idea behind the global minimization strategy is to randomly generate initial estimates for the unknown parameters and then apply a standard gradient decent minimization procedure to locate a local minimum of the cost function. If this local minimum satisfies certain conditions it will be returned as the final answer, otherwise, the algorithm tries again with a new set of random initial estimates. This is the same technique that Horn employed to recover the relative orientation of two cameras from point correspondences [5].

The first stage of the global minimization process involves generating a set of random initial estimates for the camera orientations, R_j. The reconstruction algorithm should be provided with an indication for the orientation of each camera in the form of a rotation interval $\langle R_0, \theta \rangle : R_0 \in SO(3), \theta_{max} \in \mathbb{R}$. R_0 represents an estimate of the camera orientation while θ_{max} represents the maximum amount of angular error in the estimate in radians. On a mobile robot, for example, initial estimates for the camera orientations could be obtained from odometry while the parameter θ_{max} would reflect the uncertainty in these measurements. The rotation interval actually denotes a closed set of rotation matrices defined as follows: $\{R \mid R = R_0 \exp\{J(\mathbf{w})\}, \sqrt{\mathbf{w}^T\mathbf{w}} \leq \theta_{max}\}$ where $J(\mathbf{w})$ denotes the skew symmetric matrix associated with the vector $\mathbf{w}$, and exp is the matrix exponential operator. Note that if initial estimates for the camera orientation are unavailable, then R_0 is simply the identity matrix and θ_{max} is set to π.

Initial estimates for the camera orientations are generated by choosing random elements from the rotation intervals associated with each camera position. In each case a random element is selected from the closed ball $\{\mathbf{w} \mid \sqrt{\mathbf{w}^T\mathbf{w}} \leq \theta_{max}\}$, and the corresponding rotation matrix is taken as an initial estimate. These initial estimate are used as a starting point for the gradient descent minimization procedure described in section III-B which converges to a local minimum of the objective function.

At the end of the local minimization step, the disparity between the predicted edges and the observed edges is compared to some preset threshold. If it is below, the minimum is accepted as a feasible estimate; otherwise, a new set of initial estimates for the camera orientations is generated, and the local minimization process is restarted. Since $\mathcal{O}$ is based on image error, the value of this threshold is determined by considering the maximum amount of error expected in the image measurements.

The number of local minimizations performed before a feasible estimate is obtained will depend upon the shape of the error surface and on the size of the rotation intervals provided to the algorithm. A set of simulation experiments was carried out to evaluate the convergence of this global minimization technique, and the results are presented in section IV-A.6. In practice, on the configurations that were used in the simulations and in the real data experiments, a feasible answer was usually obtained on the first try, although some of the experiments took as many as twenty or thirty passes before the global minimum was found.

B. Local Minimization

This subsection describes the procedure used to obtain a local minimum of the objective function $\mathcal{O}$ from a set of

TABLE I

An overview of the four stages in the local minimization strategy.

A Construct initial estimates for the line orientations $\hat{\mathbf{v}}_i$ by minimizing $\mathcal{C}_1$.
Free parameters : $\hat{\mathbf{v}}_i$
Number of degrees of freedom : $2n$

B Improve the estimates of R_j, $\hat{\mathbf{v}}_i$ by minimizing $\mathcal{C}_1$.
Free parameters : $\hat{\mathbf{v}}_i, R_j$
Number of degrees of freedom : $2n + 3(m-1)$

C Construct initial estimates for $\mathbf{d}_i$ and $\mathbf{t}_j$ by minimizing the quadratic functional $\mathcal{C}_2$. One of the non-zero translation parameters is held constant to fix the scale.
Free parameters : $\hat{\mathbf{d}}_i, \mathbf{t}_j$
Number of degrees of freedom : $2n + 3(m-1) - 1$

D Use the initial estimates for R_j, $\hat{\mathbf{v}}_i$, $\mathbf{d}_i$ and $\mathbf{t}_j$ provided by the previous stages as initial estimates and minimize the main objective function $\mathcal{O}$ with respect to all of the free parameters to obtain the final estimates for R_j, $\hat{\mathbf{v}}_i$, $\mathbf{d}_i$ and $\mathbf{t}_j$
Free parameters : $\hat{\mathbf{v}}_i, R_j, \mathbf{d}_i, \mathbf{t}_j$
Number of degrees of freedom : $4n + 6(m-1) - 1$

initial estimates of the camera orientations, R_j. The reader will notice that the objective function given in (1) is actually expressed in terms of four different types of parameters; camera orientations, R_j, camera translations, $\mathbf{t}_j$, line directions, $\hat{\mathbf{v}}_i$, and line positions, $\mathbf{d}_i$. Before carrying out the minimization of $\mathcal{O}$, initial estimates $(\mathbf{t}_j, \mathbf{v}_i, \mathbf{d}_i)$ are generated from the image measurements $\mathbf{u}_{ij}$ and the random initial estimate of R_j provided by the global optimization procedure.

The local minimization process is decomposed into four stages which are summarized in table I and now described in detail. A multi-stage method was employed in order to reduce the overall time taken by the algorithm. The simpler initial stages were designed to provide good initial estimates for the final optimization stage which minimizes the non-linear objective function with respect to all the unknown parameters simultaneously. Each iteration at the final stage is relatively expensive from a computational standpoint. Computing these initial estimates helps to reduce the number of iterations required at this stage which reduces the overall compute time.

In stage A the initial estimates for R_j are used to obtain estimates for $\hat{\mathbf{v}}_i$. In stage B, better estimates for R_j and $\hat{\mathbf{v}}_i$ are determined. In stage C, we obtain initial estimates for $\mathbf{d}_i$ and $\mathbf{t}_j$, and in stage D the objective function $\mathcal{O}$ is minimized with respect to all of the free parameters to produce the final estimates for the structure of the scene and the positions of the cameras.

We begin our discussion with a careful examination of (3) which determines how a particular line projects onto the image plane at a particular camera location.

$$\begin{aligned} {}^c\mathbf{m} &= {}^c\hat{\mathbf{v}} \times {}^c\mathbf{d} \\ &= {}^c_w\mathbf{R}\,\{ {}^w\hat{\mathbf{v}} \times ({}^w\mathbf{d} - {}^w\mathbf{t}_c)\} \end{aligned} \qquad (7)$$

From this equation the following constraints can be deduced.

$${}^c\mathbf{m}^T ({}^c_w\mathbf{R}\ {}^w\hat{\mathbf{v}}) = 0 \qquad (8)$$

$${}^c\mathbf{m}^T ({}^c_w\mathbf{R}\,({}^w\mathbf{d} - {}^w\mathbf{t}_c)) = 0 \qquad (9)$$

Given an observed edge $\mathbf{u}_{ij}$, the measured normal $\hat{\mathbf{m}}'$ to the plane passing through the camera center and the observed edge can be obtained as follows:

$$\begin{aligned} {}^c\mathbf{m}' &= (x_1,\ y_1,\ -1)^T \times (x_2,\ y_2,\ -1)^T \\ {}^c\hat{\mathbf{m}}' &= {}^c\mathbf{m}'/||{}^c\mathbf{m}'|| \end{aligned} \qquad (10)$$

An objective function can be devised which indicates how well the estimates for R_j and $\hat{\mathbf{v}}_i$ satisfy the constraint given in (8).

$$\mathcal{C}_1 = \sum_{j=1}^{m}\sum_{i=1}^{n} (\hat{\mathbf{m}'}^T_{ij} R_j \hat{\mathbf{v}}_i)^2 \qquad (11)$$

Notice that this objective function involves only, R_j and $\hat{\mathbf{v}}_i$, which means that it can be used to estimate the line directions and camera orientation independently of the camera position and line position.

In stage A the rotation matrices R_j are held constant, and the line directions are estimated by minimizing $\mathcal{C}_1$ with respect to the $\hat{\mathbf{v}}_i$ parameters. Notice that this function can be decomposed into $\mathcal{C}_1 = \sum_{i=1}^{n} \mathcal{C}_{A_i}$ where

$$\mathcal{C}_{A_i} = \sum_{j=1}^{m} (\hat{\mathbf{m}'}^T_{ij} R_j \hat{\mathbf{v}}_i)^2. \qquad (12)$$

The global minimum of $\mathcal{C}_1$ with respect to the $\hat{\mathbf{v}}_i$ parameters can actually be determined by minimizing each $\mathcal{C}_{A_i}$ term independently since each of these terms involves a different $\hat{\mathbf{v}}_i$ vector. Each $\mathcal{C}_{A_i}$ term is a simple homogeneous quadratic form which can be readily minimized using standard techniques like singular value decomposition.

In stage B the nonlinear objective function $\mathcal{C}_1$ is minimized with respect to all of the free parameters, R_j and $\hat{\mathbf{v}}_i$, to yield better estimates for both the camera orientations and the line directions. Once estimates for R_j and $\hat{\mathbf{v}}_i$ have been obtained, the constraint described in (9) can be used to obtain initial estimates for the $\mathbf{d}_i$ and $\mathbf{t}_j$ parameters. The following objective function reflects how well our estimates obey this constraint:

$$\mathcal{C}_2 = \sum_{j=1}^{m}\sum_{i=1}^{n} (\hat{\mathbf{m}'}^T_{ij} R_j (\mathbf{d}_i - \mathbf{t}_j))^2 \qquad (13)$$

From each line direction, $\hat{\mathbf{v}}_i$, two unit vectors, $\hat{\mathbf{v}}_i^x$, and $\hat{\mathbf{v}}_i^y$, can be constructed such that $\hat{\mathbf{v}} \cdot \hat{\mathbf{v}}_i^x = 0$, $\hat{\mathbf{v}} \cdot \hat{\mathbf{v}}_i^y = 0$, and $\hat{\mathbf{v}}_i^x \cdot \hat{\mathbf{v}}_i^y = 0$. Since $\mathbf{v}_i$ is orthogonal to $\mathbf{d}_i$, the line position vector $\mathbf{d}_i$ can be expressed in terms of these two vectors as

$\mathbf{d}_i = \alpha\hat{\mathbf{v}}_i^x + \beta\hat{\mathbf{v}}_i^y$. This allows us to rewrite the objective function given in (13) as follows:

$$\mathcal{C}_2 = \sum_{j=1}^{m}\sum_{i=1}^{n}(\hat{\mathbf{m}'}^T_{ij} R_j(\alpha\hat{\mathbf{v}}_i^x + \beta\hat{\mathbf{v}}_i^y - t_x\hat{x} - t_y\hat{y} - t_z\hat{z}))^2 \quad (14)$$

Notice that this cost function is actually a simple quadratic form with respect to the parameters α, β, t_x, t_y, and t_z. This means that the standard, closed form linear least squares equations can be applied to obtain initial estimates for these parameters.

Other researchers have used the constraints described in equations (8) and (9) to recover the position of an observer with respect to a known constellation of straight line features from image data. Liu, Huang and Faugeras [24] presented an algorithm that solves for the camera orientation first and then the camera translation. Kumar and Hanson [25] proposed a related technique that solves for the rotational and translational parameters simultaneously. In this case, these constraints are being used to estimate both the camera positions and the structure of the scene.

Once we have initial estimates for R_j, $\hat{\mathbf{v}}_i$, $\hat{\mathbf{d}}_i$ and $\mathbf{t}_j$, the main objective function $\mathcal{O}$ given in (1) can be minimized directly using the technique described in [26] to obtain the final estimate for the structure of the scene and the positions of the camera. This minimization involves a total of $4n+6(m-1)-1$ independent parameters. The optimization method used at this stage is very similar to the approach advanced by Steven Smith in his dissertation [27]. Smith showed how to carry out a version of the Newton minimization algorithm on Riemannian manifolds and proved that this method shared the quadratic convergence properties of its Euclidean counterpart. The final results produced after this stage are generally twice as accurate as the estimates provided by stage C.

In an earlier version of this work [20] the unknown parameters were divided into two sets: the structural parameters $\hat{\mathbf{v}}_i, \mathbf{d}_i$ and the camera position parameters $R_j, \mathbf{t}_j$. On every iteration of the optimization algorithm, the objective function was minimized with respect to each set of parameters independently as described in [28] in order to reduce the computational complexity of the overall procedure. More recently, Szeliski and Kang [12] have published work which indicates that a more direct approach to the optimization problem can actually yield a performance improvement. The argument against minimizing the objective function with respect to all the parameters simultaneously using a variant of Newton's method is that it would involve inverting a very large Hessian matrix at every iteration. Szeliski and Kang observed that this Hessian matrix was actually sparse which makes the problem much simpler. They also observed that the direct methods required very few iterations to converge to the final minima so the overall time taken to produce a result was quite reasonable.

It is a well known fact that SFM algorithms can only recover the structure of the environment up to a scale factor. In our implementation, the scale factor is set by holding one of the non-zero translation parameters constant during the last two stages.

Stage D has a much greater computational complexity than stages A or B since it involves a larger number of parameters. One technique for reducing the overall time required by the algorithm involves storing the initial estimates for the camera orientation provided to stage D by stage B in a database. On each global iteration the algorithm can then check whether the current set of initial estimates has been investigated previously before invoking the final minimization stage.

Once the infinite straight lines have been recovered, the endpoints of the lines can be reconstructed by projecting the endpoints in the images back onto the infinite straight lines. Note that these reconstructed endpoints need not correspond to the physical endpoints of the line due to occlusions or edge detector failures, and so we make no claims about the accuracy of the endpoints.

IV. Experimental Results

This section describes a series of experiments that were carried out on both real and synthetic image data in order to evaluate the effectiveness of the proposed algorithm.

Thousands of simulation trials were carried out to determine how the accuracy of the algorithm changed as different parameters of the simulation were varied. These parameters included the amount of error in the image measurements, the number of camera positions, the number of straight lines in the scene, calibration errors, and the effective baseline. The algorithm was also applied to a number of data sets obtained from actual image sequences. These experiments qualitatively demonstrate that the method can be used to successfully reconstruct a variety of scenes.

Section V presents the results from a series of simulation experiments that compare the proposed algorithm to the three frame linear method presented in [16].

A. Simulation Experiments

In all of the simulation experiments a similar arrangement of camera positions and straight line features was used. The cameras were arranged in a circular stereo configuration as shown in Fig. 4. The first camera position is located at the origin of the coordinate system and the other $(m-1)$ positions are equally spaced along the circumference of a circle of radius 250 mm. centered at the origin. This arrangement of camera positions ensures that the maximum stereo baseline between any two positions remains the same regardless of the number of camera positions used in the simulation.

The reference object is composed of a set of randomly rotated wire frame cubes 200 mm. on a side placed 1 meter away from the origin along the z-axis as shown in Fig. 4. Experiments were also carried out on configurations of random line segments, but we found that it was much easier to visually interpret the results of the reconstruction when polygonal structures were used.

Before describing the results of these experiments, we need to have some means for evaluating the accuracy of the recovery process. Below, we define three metrics that

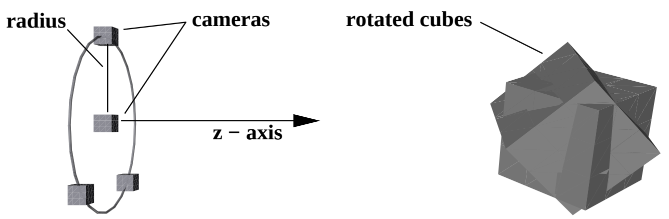

Fig. 4. A circular stereo camera configuration.

characterize the accuracy of the reconstructed lines and camera locations and orientations.

A.1 Measuring Camera Rotational Error

Any rotation matrix $R \in SO(3)$ can be written in the form $R = \exp\{\theta J(\hat{w})\}$ where $\theta \in [0, \pi]$, $\hat{w}^T \hat{w} = 1$. The magnitude of a rotation can be defined as $||R|| = ||\exp\{\theta J(\hat{w})\}|| = \theta$. Given two rotation matrices R_1 and R_2 we can define a metric function $\langle R_1, R_2 \rangle$, as $\langle R_1, R_2 \rangle = ||R_1^T R_2||$. It is relatively straightforward to show that this function is actually a metric on the Lie Group $SO(3)$. This function effectively measures the "distance" between two rotation matrices in radians.

A.2 Measuring Structural Error

In order to evaluate the accuracy of the reconstructed lines, we need to define an error measure that reflects the difference between the recovered scene structure and the actual scene. This can be accomplished by calculating the mean squared distance between each of the recovered infinite 3-D lines and the actual 3-D line segments. For every point on the line segment, the square of the distance to the closest point on the infinite straight line can be determined in closed form. This value can be integrated along the extent of the segment to obtain the total squared distance between the segment and the line. This integral is then divided by the length of the segment to yield the final result.

A.3 Experiment 1: Image Noise

The first set of simulation experiments was designed to determine how the accuracy of the reconstruction would vary as the amount of error in the image measurements was increased. A uniformly distributed random image error was added to the endpoints of the image segments. In these simulations the virtual camera had a focal length of 8 mm with a 30 degree field of view and the pixels on the image plane were assumed to be 8.37e-3 mm. on side. These parameters closely model the camera system that was used for the real data experiments.

Fig. 5 shows how the accuracy of the reconstruction varied as a function of the measurement error. Each point on these graphs represents the average of 50 trials, each of these trials involved 6 camera positions and 36 straight line features. As expected, the reconstruction error increased as the random image errors were increased.

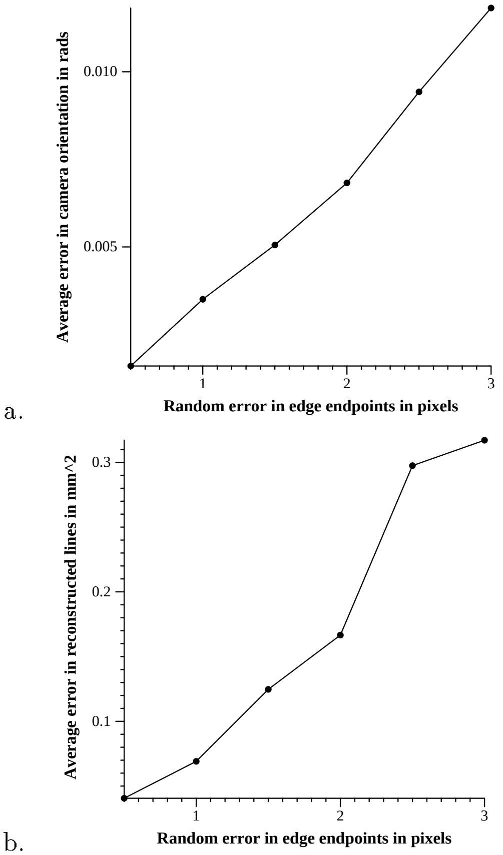

Fig. 5. How the accuracy in the reconstruction varies as a function of the magnitude of the random errors in the edge endpoints: a. Camera orientation error; b. Structure error.

A.4 Experiment 2: Calibration Errors in Camera Center

As is well known, it is very difficult to accurately and reliably determine the center of projection of a camera through calibration. This experiment was designed to explore the impact of this type of calibration error on the SFM algorithm by adding a randomly chosen camera center bias to all of the edges in the sequence. The simulation parameters were the same as in experiment 1, however no random error was added to the endpoints.

The results indicated that the algorithm is much more sensitive to random errors in the edge endpoints than it is to camera center bias, even with a camera center bias of 25 pixels the average error in the recovered camera orientations was only 0.0045 rads. while the error in the reconstructed lines was only 0.1 sq. mm..

A.5 Experiment 3: Number of Images and Number of Lines

The third set of simulation experiments was designed to determine how the accuracy of the reconstruction changed as a function of the number of camera positions m and the number of straight line features n. The random image errors were on the order of 0.5 pixels while the camera center errors were on the order of 5.0 pixels. For each combination of n and m, 100 random scenes were generated. The graphs in Fig. 6 clearly demonstrate that the accuracy of the method improves as the number of camera positions and the number of features is increased.

Note that since these experiments were carried out with the cameras arranged in a circular stereo configuration, increasing the number of camera positions does not increase the effective stereo baseline of the image sequence. Nonetheless, this experiment shows that the additional constraints obtained from the extra images can be used to improve the accuracy of the reconstruction. Similarly, each additional line in the scene contributes a set of image measurements which help to constrain the structure of the scene.

These graphs also show a pattern of diminishing returns, that is increasing the number of images used beyond 6 or the number of straight line features beyond 50 does not improve the accuracy of the method significantly.

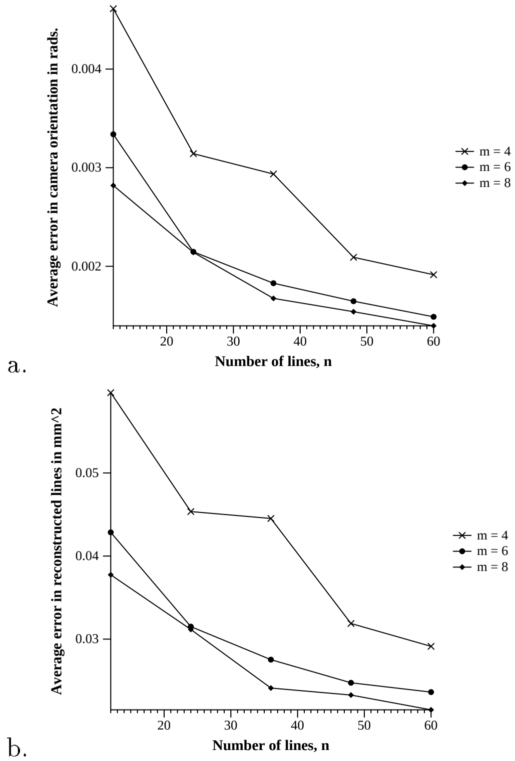

Fig. 6. Variation of reconstruction accuracy as a function of the number of images m and the number of lines n: a. Camera orientation error; b. Structure error.

A.6 Experiment 4: Global Convergence

This experiment was designed to investigate the efficiency of the global minimization procedure outlined in section III-A. In these experiments the accuracy of the initial estimates for the camera orientations provided to the algorithm, θ_{max} was varied between 20 degrees and 180 degrees. Each experiment involved 36 line segments and 4 camera positions. The random image errors were on the order of 0.5 pixels while the camera center errors were on the order of 5.0 pixels. For each value of θ_{max} 50 experiments were run and the number of global iterations performed before a feasible minimum was obtained was recorded (see table II). Note that the term *global iterations* refers to the number of times that the algorithm generates a new set of random initial estimates for the unknown parameters and *not* the number of iterations required to converge from a random initial estimate to a local minima.

TABLE II

This table shows how the number of global iterations required for convergence varies as a function of the amount of uncertainty in the initial estimates, θ_{max}.

θ_{max} in degrees	median no. of global iterations	Percentage of experiments converging within 500 iterations
20	2	100
40	4	100
60	9	98
90	14	98
180	159	90

As the amount of error in the original estimates increases, the median number of iterations required to locate a feasible global minimum rises since the algorithm is forced to investigate a larger portion of the parameter space. Even when the algorithm has no information about the camera orientations ($\theta_{max} = 180$ degrees), it still manages to locate the global minimum when given enough trials.

A.7 Experiment 5: Baseline and Number of Images

This experiment was designed to show how the accuracy of the method changes with the radius of the circular stereo configuration and the number of camera positions. The experiment was designed to determine whether the accuracy of the method was more sensitive to the number of images used or the stereo baseline of the system.

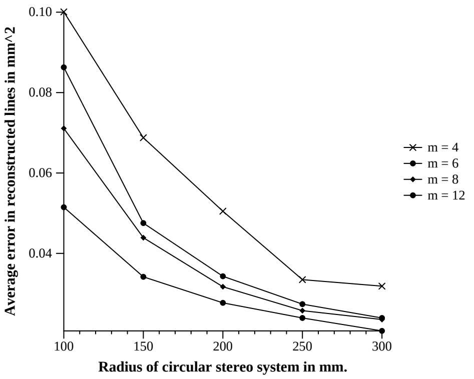

Fig. 7. Variation of reconstruction accuracy as a function of the number of images m and the radius of the circular stereo configuration.

Fig. 7 clearly shows that the accuracy of the method improves as the effective stereo baseline of the camera arrangement increases. The results also suggest that a small number of widely spaced camera positions can actually provide better structural estimates than a large number of closely spaced positions. Each point on the graph represents the average results obtained over 100 experiments. The random image errors in these experiments were on the order of 0.5 pixels while the camera center errors were on the order of 5.0 pixels, each experiment involved 36 line segments.

B. Real Data Experiments

Experiments were carried out on a variety of image sequences taken in and around our laboratory. Images were digitized to 464 by 572 pixels using a CCD camera with an 8 mm lens. In each of these experiments the image edges were obtained using a variation of the Canny edge detector [21]. The line correspondences were determined manually, and initial estimates for the camera orientations were obtained by taking eyeball estimates. The intrinsic parameters (aspect ratio, camera center and quadratic radial distortion) of the camera system were obtained from a set of calibration images.

B.1 Scene 1: Building Blocks

Fig. 8 shows two of the eight images taken of a stack of building blocks, and Fig. 9 shows various views of the reconstruction of that scene. These scenes are rendered assuming perspective projection with the same focal length as the actual camera.

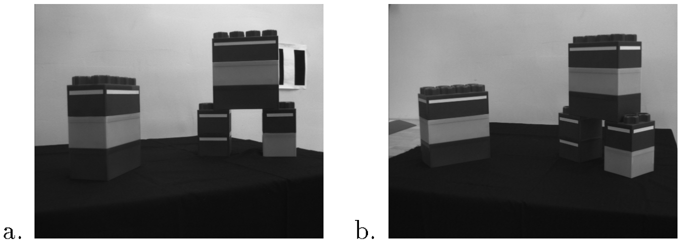

Fig. 8. Two images from a sequence of eight that were used for reconstruction.

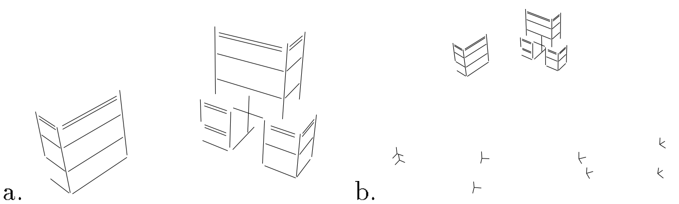

Fig. 9. Two views of the reconstructed scene. The small coordinate axes in figure b represent the reconstructed camera positions.

Qualitatively, the lines appear to be well reconstructed. The figures show that the relationships between lines in the reconstruction reflect the relationships between the corresponding lines in the actual structure; parallel, perpendicular or coplanar lines on the actual object appear in the same configuration in the reconstruction. Notice that the algorithm could not provide estimates for the positions of the line segments on the far side of the structure since none of these lines were visible in any of the images. Note that we make no claims about the locations of the endpoints since these are simply taken to be the extrema of the projections of the measured image endpoints onto the reconstructed lines.

B.2 Scene 2: The Eli Cube

Fig. 10 shows two of the ten images taken of a marked box, while Fig. 11 shows various views of the reconstruction of that scene including the location of the cameras.

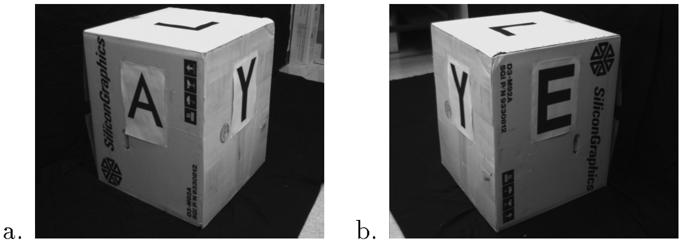

Fig. 10. Two images taken from a sequence of ten.

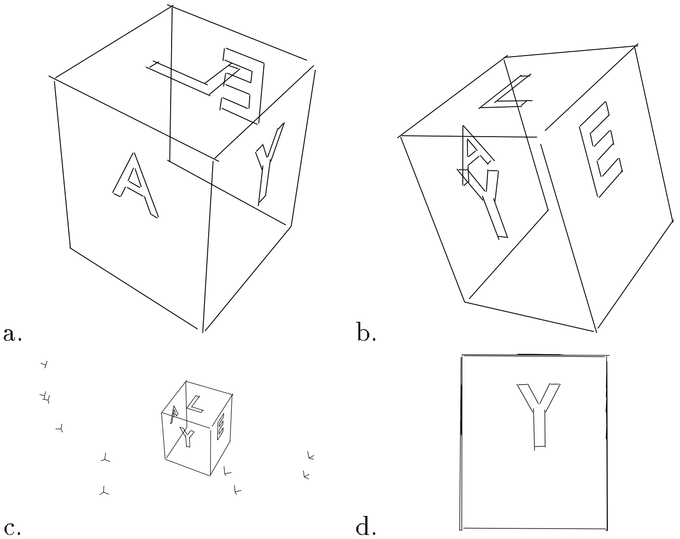

Fig. 11. A variety of views of the reconstructed scene. The small coordinate axes in figure c indicate the reconstructed camera positions. Figure d is rendered under orthographic projection.

In this experiment, the camera was moved around the object in order to obtain views of different sides of the cube; note that opposite sides of the cube (A and E) cannot be seen simultaneously. The algorithm was able to take advantage of all available image data to reconstruct four of the six faces of the cube. From the side view under orthographic projection shown in Fig. 11.d, one can see that the individual letters in the reconstructed scene are nearly coplanar with the faces of the cube.

B.3 Scene 3: Inside a building

Fig. 12 shows four of the twenty four images taken from an area inside our office complex. A large number of views were taken to cover a large area of the scene; in contrast to the two previous examples where the objects being viewed were relatively small, here the robot is immersed within the scene being reconstructed. Several panoramic views of the scene (Figs. 12.a–12.b) were used in this experiment along with more focused views (Figs. 12.c–12.d) which captured various details.

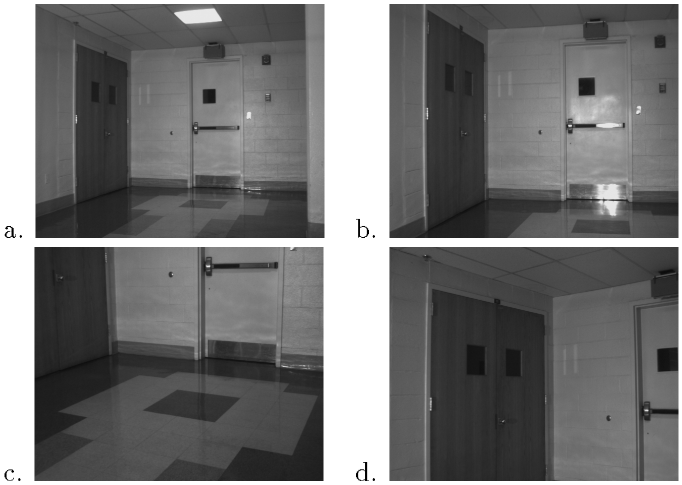

Fig. 12. Four images taken from a sequence of 24 taken from a section of our office complex.

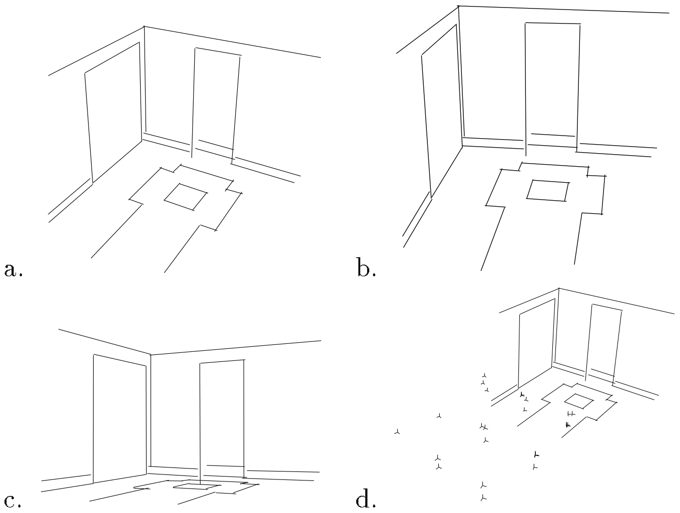

Fig. 13. A set of views of the reconstruction of the scene. The small coordinate axes in figure d represent the reconstructed camera positions.

Fig. 13 shows various views of the reconstruction of that scene. The algorithm successfully reconstructed the walls, the doors and the markings on the floor. Because of the limited field of view, more images were needed to capture the structure of the scene than in the previous two examples.

B.4 Summary

Note that each of these image sequences contained edges that were visible in some images but not in others. The presented algorithm can handle this situation quite naturally and still provide acceptable reconstruction results.

The reconstruction times are given in table III. The algorithm was implemented in C and all the computations were performed on a Silicon Graphics R4000 Indigo.

TABLE III

Compute time required for each scene.

Scene	number of lines, n	number of camera positions, m	Total compute time in secs.
1	45	8	437.09
2	48	10	553.40
3	33	24	790.47

V. Comparison with Linear Algorithm

A series of simulation experiments were carried out in order to compare the algorithm presented in this paper to the three frame linear techniques proposed in [16], [29], [17]. We chose to implement the algorithm described in [16] because it was considered to be one of the best linear structure from motion methods.

These experiments simulated a trinocular stereo configuration viewing a set of three rotated cubes (a total of 36 line segments). In this configuration the three cameras were placed at regular intervals along a circle of radius 250 mm., the rotated cubes were 200 mm. on side and were positioned 1 meter away from the center of the trinocular system. This arrangement is almost identical to the one shown in figure 4 the only difference being that the camera at the center of the circle was omitted.

These experiments were designed to determine how the accuracy of the reconstruction would vary as the amount of error in the image measurements was increased. A uniformly distributed random image error was added to the endpoints of the image segments. The simulated cameras had a focal length of 8mm with a 30 degree field of view, and the pixels in the image plane were assumed to be 8.373e-3 mm. on a side. The same input data was supplied to the three frame linear algorithm (WLHA88) and the non-linear algorithm (TK93).

Fig. 14 shows a side by side comparison of the results obtained from the WLHA88 algorithm and the TK93 algorithm for the same data set. The image error in the input data set was 0.5 pixels. Fig. 15 shows how the accuracy of the reconstruction varied as a function of image error. Each point in these graphs represents the average of 50 trials.

These experiments demonstrate that the linear method does not perform well in the presence of image error. In fact, the results become unusable as the image error is increased beyond 0.5 pixels. When the errors in the image measurements are relatively large, 1.5 pixels or greater, the estimate for the camera orientations provided by the WLHA88 algorithm is essentially random since the average error in these estimates approaches and exceeds $\pi/2$. This also means that no real advantage would be gained by using the results from the linear algorithm as initial estimates for a more sophisticated iterative technique. Fig. 15 also indicates that the TK93 algorithm produced results that were at least an order of magnitude more accurate than

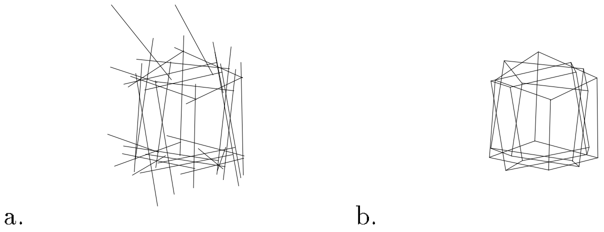

Fig. 14. Figure a. shows the reconstruction produced by the WLHA88 three frame linear algorithm while Figure b. shows the results obtained by the TK93 non-linear algorithm on the same data set. In this experiment, 0.5 pixels of random image error were added to the endpoints of the simulated image edges.

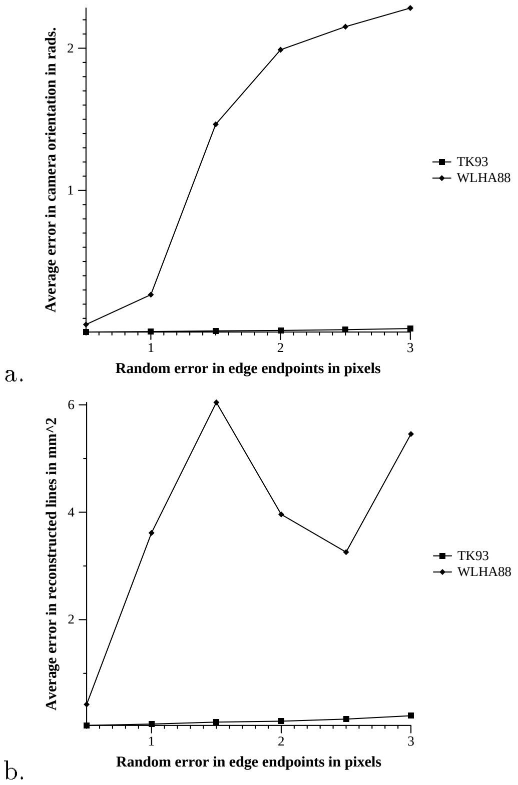

Fig. 15. A comparison of how the accuracy of the reconstruction varies as a function of the magnitude of the random errors in the edge endpoints: a. Camera orientation error; b. Structure error.

those produced by the WLHA88 algorithm.

Unlike the WLHA88 algorithm, the TK93 algorithm is not limited to three frames of image data. It can take advantage of any additional frames that may be available to improve the accuracy of its estimates. The WLHA88 algorithm also requires a minimum of thirteen line correspondences in three frames while the TK93 algorithm only requires 6 line correspondences in three images to produce a result. This is the minimum number of correspondences required to solve this particular structure from motion problem [8].

VI. Conclusion

This paper presented a novel algorithm for recovering the structure of a constellation of straight line features and the motion of an observer from a set of edge correspondences derived from an image sequence. The structure from motion problem is formulated in terms of a non-linear objective function that measures the total squared disparity between the actual image measurements and the predicted image measurements. The algorithm obtains an estimate for the structure of the environment and the positions of the camera by minimizing this non-linear objective function with respect to the unknown parameters. It avoids the linearizing assumptions inherent in the Extended Kalman Filter approaches and can, therefore, be expected to yield more accurate results. New techniques were developed in order to carry out the numerical minimization procedures over the non-linear parameter manifolds and to explore the critical points of this cost function. These techniques have proven to be quite effective for this class of problems.

Thousands of simulation experiments were carried out in order to investigate how the accuracy of the algorithm would be affected as different parameters were varied. These experiments show that the algorithm can produce very accurate results even in the presence of significant amounts of random error in the edge endpoints and large calibration errors in the image center parameters. The algorithm has been shown to be far more accurate than a well regarded linear one. It was also applied to a number of data sets obtained from actual image sequences. These experiments demonstrate that the method can be used to successfully reconstruct a variety of scenes.

The proposed algorithm also offers several advantages over other approaches: it can be applied to image sequences where various structural features do not appear in every image, it can be applied in situations where no initial estimates are available for the camera orientations and the algorithm's computations can be performed in a reasonable amount of time on a standard workstation.

The experiments also indicate that the accuracy of the method is critically dependent upon the positions of the camera with respect to the scene being observed. This is a fundamental property of any structure from motion algorithm. At present, there is no systematic technique for choosing an appropriate set of viewpoints for a structure from motion algorithm. This problem should definitely be addressed in future research.

The real data experiments were performed by applying a Canny edge detector to each of the images in the sequence and then determining the correspondences between the extracted features manually. We plan to automate this process by developing algorithms to track the edge features throughout a video sequence.

Finally, the world around us is composed of more than simply lines floating in space; these features typically lie on planar surfaces. While the presented approach is entirely a bottom-up process, more accurate estimates can be obtained if model constraints are imposed in a top-down fashion. For example, if corresponding straight lines form an L-junction or Y-junction in two or more images, then the lines must intersect at a point in the 3D world. This observation can be used to constrain the solution during the minimization. Other equality constraints such as coplanarity and parallelism could also be used to achieve greater accuracy. One further improvement would be to explicitly satisfy the inequality constraints imposed by T-junctions as is done in Sugihara's work on line drawing interpretation [30].

Implementation: The implementation of this algorithm, our implementation of the algorithm presented in [16], and the data sets used during our experiments are available for noncommercial use. They can be accessed via anonymous ftp at daneel.eng.yale.edu or by contacting either of the authors.

Acknowledgments: We would like to thank P. Anandan for his contributions to this research.

References

[1] S. Ullman, *The Interpretation of Visual Motion*, The MIT Press, Cambridge, MA, 1979.

[2] H.C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections", *Nature*, vol. 293, pp. 133–135, 1981.

[3] 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. 11, no. 5, pp. 451–476, May 89.

[4] X. Hu and N. Ahuja, "Motion and structure estimation using long sequence motion models", *Image and Vision Computing*, vol. 11, no. 9, pp. 549–570, November 1993.

[5] B.K.P. Horn, "Relative orientation", *Int. J. Computer Vision*, vol. 4, pp. 59–78, 1990.

[6] J. Weng, T. S. Huang, and N. Ahuja, *Motion and Structure from Image Sequences*, Springer Series on Information Sciences. Springer-Verlag, Berlin, 1993.

[7] James L. Crowley, Patrick Stelmaszyk, Thomas Skordas, and Pierre Puget, "Measurement and integration of 3-D structures by tracking edge lines", *International Journal of Computer Vision*, vol. 8, no. 1, pp. 29–52, July 1992.

[8] Olivier D. Faugeras, Francis Lustaman, and Giorgio Toscani, "Motion and structure from point and line matches", in *International Conference on Computer Vision*, June 1987, pp. 25–33.

[9] J. L. Jezouin and N. Ayache, "3d structure from a monocular sequence of images", in *International Conference on Computer Vision*. IEEE, December 1990, p. 441.

[10] T. Vieville and O. Faugeras, "Feed-forward recovery of motion and structure from a sequence of 2d-lines matches", in *International Conference on Computer Vision*. IEEE, December 1990, p. 517.

[11] Carlo Tomasi and Takeo Kanade, "Shape and motion from image streams under orthography: a factorization method", *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, November 1992.

[12] Richard Szeliski and Sing Bing Kang, "Recovering 3D shape and motion from image streams using non-linear least squares", *Journal of Visual Communication and Image Representation*, vol. 5, no. 1, pp. 10–28, March 1994.

[13] T. Vieville, "Estimation of 3d-motion and structure from tracking 2d-lines in a sequence of images", in *European Conference on Computer Vision*, April 1990, p. 281.

[14] B. Giai-Checa and T. Vieville, "3d-vision for active visual loops using locally rectilinear edges", in *Proceedings of the 1992 IEEE Int. Symp. on Intelligent Control*, August 1992, p. 341.

[15] Nassir Navab, Rachid Deriche, and Olivier D. Faugeras, "Recovering 3d motion and structure from stereo and 2d token tracking", in *International Conference on Computer Vision*. IEEE, December 1990, p. 513.

[16] J. Weng, Y. Liu, T. S. Huang, and N. Ahuja, "Estimating motion/structure from line correspondences: A robust linear algorithm and uniqueness theorems", in *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, 1988, pp. 387–392.

[17] M. E. Spetsakis and J. Aloimonos, "Structure from motion using line correspondences", *Int. J. Computer Vision*, vol. 4, no. 3, pp. 171–184, June 1990.

[18] Minas Spetsakis, "A linear algorithm for point and line based structure from motion", *CVGIP:Image Understanding*, vol. 56, no. 2, September 1992.

[19] Olivier Faugeras, *Three-Dimensional Computer Vision*, MIT Press, 1993.

[20] C.J. Taylor, D.J. Kriegman, and P. Anandan, "Structure and motion in two dimensions from multiple images: A least squares approach", in *IEEE Workshop on Visual Motion*, Oct. 1991, pp. 242–248.

[21] John Canny, "A computational approach to edge detection", *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 679–98, Nov. 1986.

[22] John Craig, *Introduction to Robotics: Mechanics and Control*, Addison-Wesley, New York, 1989.

[23] G.H. Golub and C.F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 1983.

[24] Yuncan Liu, Thomas S. Huang, and O. D. Faugeras, "Determination of camera location from 2D to 3D line and point correspondences", in *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, 1988, pp. 82–88.

[25] Rakesh Kumar and Allen R. Hanson, "Robust estimation of camera location and orientation from noisy data having outliers", in *Proceedings of the Workshop on the Interpretation of 3D Scenes*, November 1989, pp. 52–60.

[26] Camillo J. Taylor and David J. Kriegman, "Minimization on the lie group $SO(3)$ and related manifolds", Tech. Rep. 9405, Center for Systems Science, Dept. of Electrical Engineering, Yale University, New Haven, CT, April 1994.

[27] Steven Smith, *Geometric Optimization Methods for Adaptive Filtering*, PhD thesis, Harvard University, Division of Applied Sciences, Cambridge MA, September 1993.

[28] Morris M. Thompson, *Manual of Photogrammetry*, American Society of Photogrammetry, 1966.

[29] Yuncan Liu and Thomas S. Huang, "A linear algorithm for motion estimation using straight line correspondences", *Comp. Vision, Graphics, and Image Proc.*, vol. 44, no. 1, pp. 35–57, October 1988.

[30] Kokichi Sugihara, "An algebraic approach to the shape-from-image problem", *Artificial Intelligence*, vol. 23, pp. 59–95, 1984.

Camillo J. Taylor received an A.B. in Electrical, Computer and Systems Engineering *magna cum laude with highest honors* from Harvard University in 1988. He received the M.S. and Ph.D. degrees in Electrical Engineering from Yale University in 1990 and 1994 respectively. Dr. Taylor is a member of the Harvard chapter of *Phi Beta Kappa* and was the recipient of the Jamaica Scholarship in 1985. His research interests include structure from motion, mobile robots and advanced recognition systems. He is currently a postdoctoral researcher and lecturer with the Department of Electrical Engineering and Computer Science at U.C. Berkeley.

David J. Kriegman was born in Englewood, New Jersey on October 24, 1961. He graduated *summa cum laude* from Princeton University with a B.S.E. degree in Electrical Engineering and Computer Science in 1983 where he was awarded the Charles Ira Young Award for electrical engineering research. He received the M.S. degree in 1984 and Ph.D. in 1989 in electrical engineering from Stanford University where he studied under a Hertz Foundation Fellowship.

Currently, he is an Associate Professor at the Center for Systems Science in the Departments of Electrical Engineering and Computer Science at Yale University and was awarded a National Science Foundation Young Investigator Award in 1992. Dr. Kriegman's research interests include mobile robotics and computer vision.