In Proc. Fifth Intl. Conf. on Computer Vision, Cambridge MA, 1995, pp. 330-337.

Complete Scene Structure from Four Point Correspondences

Steven M. Seitz
seitz@cs.wisc.edu

Charles R. Dyer
dyer@cs.wisc.edu

Department of Computer Sciences
University of Wisconsin
Madison, WI 53706

Abstract

A new technique is presented for computing 3D scene structure from point and line features in monocular image sequences. Unlike previous methods, the technique guarantees the completeness of the recovered scene, ensuring that every scene feature that is detected in each image is reconstructed. The approach relies on the presence of four or more reference features whose correspondences are known in all the images. Under an orthographic or affine camera model, the parallax of the reference features provides constraints that simplify the recovery of the rest of the visible scene. An efficient recursive algorithm is described that uses a unified framework for point and line features. The algorithm integrates the tasks of feature correspondence and structure recovery, ensuring that all reconstructible features are tracked. In addition, the algorithm is immune to outliers and feature-drift, two weaknesses of existing structure-from-motion techniques. Experimental results are presented for real images.

1 Introduction

Many existing structure-from-motion algorithms generate optimal structure estimates but typically reconstruct very little of the visible scene. This shortcoming is due primarily to the reliance on error-prone, nearest-neighbor-based, feature tracking methods [1, 2] that can reliably track only a subset of the visible image features. Since the set of trackable image features is generally a small subset of the detected image features, the 3D reconstruction is necessarily incomplete.

In this paper we present a novel approach for 3D scene reconstruction that provides guarantees on the completeness, consistency, and optimality of the reconstruction. Furthermore, no limiting assumptions such as spatial or temporal smoothness are needed. Specifically, we guarantee

- Completeness: Every continuously-visible scene feature is reconstructed
- Consistency: Every reconstructed scene feature can be explained by a feature in each image.
- Optimality: The computed structure is the best least-squared approximation to the image measurements, subject to constraints provided by a set of *reference features*.

By *scene feature*, we mean any point or line that moves rigidly, such as a surface marking or surface orientation discontinuity. To be reconstructible, a scene feature must be detected in each image. We assume that the projection process is accurately modeled by an affine camera model. In addition, we require that the 2D positions of at least four non-coplanar corresponding *reference features* can be found in each image.

The fundamental insight used in this paper is that the features that *can* be reliably tracked provide geometric constraints that simplify the correspondence and reconstruction of the *rest* of the features in the image sequence. Hence, we can extend a small set of feature correspondences to a complete set covering all reconstructible features, and to a complete reconstruction.

The reference features are used to simplify structure recovery in two ways: First, the parallax of the reference features determines epipolar lines that constrain the feature correspondence process to a one-dimensional search. Subsequent images further constrain the possible matches. Second, the reference features determine a global affine reference frame in which structure recovery is straightforward. Completeness is ensured by maintaining a separate hypothesis for each set of possibly corresponding image features, subject to the epipolar constraints. Consistency is achieved by discarding any hypothesis that cannot be explained by a 3D scene feature.

Our approach has a number of useful properties that resolve outstanding problems in current structure-

The support of the National Science Foundation under Grant Nos. IRI-9220782 and CDA-9222948 is gratefully acknowledged. We would like to thank Rich Madison and Carlo Tomasi for providing the program used to track point-features for the cube sequence.

from-motion approaches. In particular, outlier detection and removal occur naturally in the process of structure recovery, avoiding the need for special outlier detection algorithms. Outliers pose a significant problem for structure recovery algorithms based on least-squares techniques [3, 4] because a few bad features will bias the entire reconstruction. In our approach, each feature is recovered *independently* so outliers do not bias the recovery of other features. Furthermore, the algorithm automatically eliminates any image feature that cannot be explained by a 3D scene feature, thereby virtually eliminating outliers altogether. A similar issue is *feature-drift*, which typically occurs in long image-sequences due to the propagation of tracking errors. Feature-drift does not arise in our approach because all possible sets of feature correspondences are retained as separate hypotheses. A hypothesized set of corresponding image features is discarded only when there is no scene feature that could explain it. Finally, the worst-case computational complexity of the algorithm is $O(nm^3)$ for n images and m reconstructible scene features, in contrast to the exponential growth generally exhibited by complete algorithms [2].

The rest of the paper is organized as follows: Section 2 reviews related work on structure-from-motion and feature tracking. Section 3 discusses the constraints provided by a set of reference features, including the derivation of an affine reference frame and the determination of epipolar lines. The framework for reconstructing scene features is presented in Section 4. Two optimizations are described in Section 5 that reduce the correspondence problem to a table lookup. The complete algorithm is presented in Section 6, and Section 7 presents experimental results on real images.

2 Related Work

Despite its importance, the completeness problem has been neglected in the structure-from-motion literature, although a few vision researchers have considered completeness in related problems. Kutulakos and Dyer [5] introduced a provable algorithm for global surface reconstruction using an active observer. Their approach provided guarantees on the completeness of the recovered scene, but required continuous and controlled camera motion. There has been some work on feature-tracking using multiple hypotheses to generate and maintain different sets of possible feature correspondences [2]. Unfortunately, these algorithms have exponential complexity so suboptimal approximations are used in practice. Moreover, the strategies for hypothesis pruning are based on assumptions such as motion continuity that are often violated in practical applications.

Other researchers have noted that the image motion of a few reference features provides useful geometric constraints. Koenderink and van Doorn [6] showed that four points determine a global, object-centered, affine reference frame in which 3D structure information can be recovered. Several researchers have noted that epipolar lines can be determined from a number of feature correspondences in uncalibrated images [4, 7, 8, 9, 10]. The epipolar lines constrain the set of possible feature correspondences and can be obtained from as few as four corresponding points under orthographic or affine projection. Other work has shown that additional images further constrain correspondences and this property has been used in trinocular stereopsis [10, 11].

3 Constraints from Four Features

Image motion of a rigid scene is known to be highly constrained; the projections of any scene feature are limited to a set of epipolar lines. The problem is that these epipolar lines are not generally known in advance so they are not used in feature tracking. Recent results [4, 9, 10], however, have shown that epipolar lines can be determined from as few as four feature correspondences, without the need for camera calibration. These results suggest a two stage solution to the feature correspondence problem where a few feature correspondences are used to simplify tracking for the rest of the scene.

Our algorithm for recovering scene structure depends upon the acquisition of a set of affine projection matrices, $\mathbf{\Pi}_i$, that allow inverse projection of image features into a global 3D affine space. Several methods have been proposed for determining these quantities from point correspondences [4, 6]. We use an adaptation of the method proposed by Tomasi and Kanade [3] for an orthographic camera and modified by Shapiro [4] for the affine case. This method was chosen because (1) more than four points may be used, and (2) it has been adapted for recursive estimation of the projection matrices [12] when the images are processed incrementally. The original (non-recursive) method is reviewed briefly below.

3.1 Projection Matrices

From four or more non-coplanar image feature correspondences it is possible to obtain the projection equations that map a point from an affine 3D space to each image. We use the factorization approach [3, 4] to determine the affine projection matrices. For simplicity, assume that the origin of each image I_i is chosen to be the centroid of the k reference features. The projection matrices are found by concatenating the image measurements, forming a $2n \times k$ matrix $\mathbf{M}$ and computing the singular value decomposition $\mathbf{M} = \mathbf{U\Sigma V}$. The projection matrices $\mathbf{\Pi}_1, \ldots, \mathbf{\Pi}_n$ are the successive 2×3 blocks of the first three columns of $\mathbf{U}$. As in [4], the projection matrices need not be orthogonal, in which case the reconstructed scene will be an affine transformation of the true scene. For further details of the algorithm, see [3, 4, 12].

It is useful to know the direction of the optical axis $\mathbf{K}_i$ of I_i, also known as the *direction of projection* for an orthographic camera. The optical axis of image I_i is the null-space of $\mathbf{\Pi}_i$, i.e., $\mathbf{K}_i = \{\boldsymbol{P} \mid \mathbf{\Pi}_i \boldsymbol{P} = 0\}$, and points in the direction of the eigenvector of the matrix $\begin{bmatrix} & \mathbf{\Pi}_i & \\ 0 & 0 & 0 \end{bmatrix}$ with eigenvalue 0.

3.2 Epipolar Lines

For a given point $\boldsymbol{p}$ in image I_1, the epipolar line in I_i is defined to be the projection of the line $L^p = \{\boldsymbol{P} \mid \mathbf{\Pi}_i \boldsymbol{P} = \boldsymbol{p}\}$ into image I_i. An implicit form of L^p can be found in terms of the *affine fundamental matrix* [4]. We present a different derivation which involves aligning the affine coordinate system with I_1. The advantage of the latter approach is that the epipolar geometry is given *directly* by the projection matrices.

Suppose that the X and Y axes of the global affine coordinate frame are aligned with the first image, i.e.,

$$\mathbf{\Pi}_1 = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \end{array} \right] \tag{1}$$

and $\boldsymbol{p} = [x\ y]^T$ is a point in I_1. Then

$$L^p = \{[x\ y\ Z]^T \mid Z \in \Re\}$$

If $\mathbf{\Pi}_i$ is partitioned as $\mathbf{\Pi}_i = [\mathbf{A}_i \mid \mathbf{d}_i]$, where $\mathbf{A}_i$ is 2×2 and $\mathbf{d}_i$ is 2×1, we get the following expression for the epipolar line of $\boldsymbol{p}$ in I_i

$$\begin{aligned} l_i^p &= \{\mathbf{\Pi}_i [x\ y\ Z]^T \mid Z \in \Re\} \\ &= \{\mathbf{A}_i \boldsymbol{p} + Z\mathbf{d}_i \mid Z \in \Re\} \end{aligned} \tag{2}$$

Therefore, $\boldsymbol{d}_i$ is the direction of the epipolar line and $\mathbf{A}_i\boldsymbol{p}$ is its orthogonal offset from the origin of I_i. This derivation also shows that all epipolar lines in I_i are parallel, since $\mathbf{d}_i$ does not depend on $\boldsymbol{p}$.

To use formula (2) we must transform the global coordinate system so that it is aligned with I_1. This is accomplished by post-multiplying each projection matrix with any non-singular 3×3 matrix $\mathbf{S}$ satisfying:

$$\mathbf{\Pi}_1 \mathbf{S} = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \end{array} \right]$$

A simple solution for $\mathbf{S}$ is

$$\mathbf{S} = \left[\begin{array}{cc} \mathbf{A}_1^{-1} & -\mathbf{A}_1^{-1}\mathbf{d}_1 \\ 0\ \ 0 & 1 \end{array} \right]$$

Finally, note that the optical axis $\mathbf{K}_1$ of the first image in the aligned coordinate system is simply $[0\ 0\ 1]^T$.

4 Scene Reconstruction

In this section we describe a voting technique for recovering the 3D positions of image features in an affine reference frame. The technique is complete in the sense that every *reconstructible* feature is accounted for. A reconstructible feature is any point or line feature in the scene that is continuously-visible, i.e., is detected as a feature in each image. Our method makes use of an *implicit* representation of scene features that permits the representation of points and lines within a common parameter space.

The method uses epipolar constraints to find feature correspondences and a global affine reference frame in which to represent the recovered scene. The approach is incremental, providing updated optimal estimates of 3D structure as each image becomes available. The algorithm integrates the tasks of feature tracking and structure recovery into one process, ensuring that only the reconstructible features are tracked.

For simplicity of presentation, we assume that projection matrices $\mathbf{\Pi}_1, \ldots, \mathbf{\Pi}_n$ have been recovered.

4.1 Determining Correspondences

Any feature point $\boldsymbol{p}$ in I_i can be said to *vote for* the linear subspace L^p that projects to that feature. An explicit form for L^p is given by

$$L^p = \{\boldsymbol{O}^p + Z\mathbf{K}_i \mid Z \in \Re\} \tag{3}$$

where $\boldsymbol{O}^p$ is taken to be $\mathbf{\Pi}_i^{-1}\boldsymbol{p}$ with $\mathbf{\Pi}_i^{-1}$ the *pseudo-inverse* of $\mathbf{\Pi}_i$. Let $\boldsymbol{p}$ be a feature in I_1. The features that could correspond to $\boldsymbol{p}$ in I_i are the points $\boldsymbol{q}$ such that L^q intersects L^p. These features lie along l_i^p, the projection of L^p into I_i. In general, n features $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n$ correspond to the same scene point only if the lines $L^{p_1}, \ldots, L^{p_n}$ mutually intersect.

Due to measurement errors, the set of points that could project to $\boldsymbol{p}$ is more accurately modeled as a conic volume C^p with axis L^p and a constant eliptical cross-section given by the measurement covariance of $\boldsymbol{p}$. As before, $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n$ could correspond to the same scene feature only if $\bigcap C^{p_i} \neq \emptyset$.

The discussion so far suggests a voting technique such as a Hough transform to determine correspondences. Such an approach involves sampling the 3D world into a grid of bins. Each image feature, $\boldsymbol{p}$, votes for all the bins overlapping C^p. Any bin with a large number of votes supports the correspondence of the set of features voting for the bin. In addition, the position of a bin in the Hough space provides the rough 3D location of the corresponding feature in the scene.

The Hough approach is attractive but costly due to its extensive memory requirements. We present a similar, but less costly, approach in which bins are allocated only as needed. As described before, each feature $\boldsymbol{p}$ in I_1 votes for a 3D volume C^p. Initially, a bin $B^p_{-\infty,\infty}$ is allocated for each such volume. C^p can be broken into a set of segments, each parameterized by an interval $[Z_1, Z_2]$. If we define

$$L^p_{Z_1,Z_2} = \{\boldsymbol{O}^p + Z\mathbf{K}_1 \mid Z_1 \leq Z \leq Z_2\}$$

then $C^p_{Z_1,Z_2}$ is defined to be the segment of C^p with axis $L^p_{Z_1,Z_2}$. Any feature $\boldsymbol{q}$ in another image such that C^q and C^p intersect supports the hypothesis of a scene feature in the region of intersection. For simplicity, this region is approximated by the smallest segment $C^p_{Z_1,Z_2}$ that encloses $C^q \bigcap C^p$. For each such region, a new bin $B^p_{Z_1,Z_2}$ is allocated. The new set of bins replaces the old set. After n images, each segment $C^p_{Z_1,Z_2}$ that is supported by all n images will have it's own bin $B^p_{Z_1,Z_2}$.

This approach is also suited for matching corresponding image line segments. Specifically, a point $\boldsymbol{p}$ along each line segment in the first image is chosen to represent that feature. Any line segment l in a subsequent image votes for the volume $C^l = \bigcup\{C^q \mid \boldsymbol{q} \text{ on } l\}$. Every set of mutually intersecting volumes $C^p, C^{l_2}, \ldots, C^{l_n}$ will have a bin $B^p_{Z_1,Z_2}$ such that $L^p_{Z_1,Z_2} \subset \bigcap_{i=2}^n C^{l_i}$

Section 6 presents an efficient method that finds correspondences by traversing epipolar lines. The process is made more efficient by first rectifying the images so that all epipolar lines are horizontal and then caching proximity information so that corresponding features can be found by table lookup.

Since there is a one-to-one correspondence between bins and possible sets of corresponding features, **all possible interpretations of the scene are explicitly taken into account**. In addition, **only correspondences that are consistent with a 3D scene feature are considered**. These two properties ensure the completeness and consistency of the algorithm.

4.2 Determining 3D Position

In this section we make use of an *implicit* representation of three-dimensional subspaces such as points, lines, and planes, as solutions of 3×3 linear systems of equations of the form $\mathbf{A}\boldsymbol{X} = \mathbf{B}$. This representation is attractive because it offers a uniform treatment of point and line features. Furthermore, the implicit formulation generally results in a more compact (and hence more efficient) system of linear equations than does the more common explicit form.

Each image feature votes for a scene feature in the 3D subspace projecting to that feature. We represent an affine subspace S as a tuple $\langle \mathbf{A}, \boldsymbol{O} \rangle$ where $\boldsymbol{O}$ is the orthogonal offset of the subspace from the origin and $\mathbf{A}$ is a matrix whose columns span the linear subspace $S - \boldsymbol{O}$. In particular, a point $\boldsymbol{p}_i$ in I_i votes for the subspace $\langle \mathbf{K}_i, \boldsymbol{O}^{p_i} \rangle$. If $\boldsymbol{p}$ and $\boldsymbol{q}$ are two points along a line feature l_i in I_i then l_i votes for the subspace $\langle [\mathbf{K}_i \mid \boldsymbol{O}^q - \boldsymbol{O}^p], \boldsymbol{O}^p \rangle$.

In general, let $f_1, \ldots, f_n$ be a set of corresponding image point or line features and let $\langle \mathbf{A}_1, \boldsymbol{O}_1 \rangle, \ldots, \langle \mathbf{A}_n, \boldsymbol{O}_n \rangle$ be the respective subspaces for which they vote. The task is to determine the scene subspace that is as close as possible, in a least-squared sense, to these subspaces. Towards this end, the Mahalanobis distance of a point $\boldsymbol{P}$ to a subspace $\langle \mathbf{A}, \boldsymbol{O} \rangle$ is given by

$$(\boldsymbol{P} - \boldsymbol{O})^T \mathbf{HWH} (\boldsymbol{P} - \boldsymbol{O}) \tag{4}$$

where the 3×3 symmetric matrix $\mathbf{H}$ is defined by

$$\mathbf{H} = \mathbf{I} - \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$$

The 3×3 matrix $\mathbf{W}$ weights a vote according to its uncertainty and is typically chosen to be the inverse covariance matrix of the expected error. If $\mathbf{W}$ is the identity, Eq. (4) gives the orthogonal squared distance from the given subspace [13]. The weight matrix can be determined from the measurement covariance as follows: let $\mathbf{\Lambda}$ be the covariance of an image feature $\boldsymbol{p}$ in I_i. The subspace voted for by $\boldsymbol{p}$ is weighted by

$$\mathbf{W} = \mathbf{\Pi}_i^T \mathbf{\Lambda}^{-1} \mathbf{\Pi}_i$$

A scene feature is reconstructed by minimizing the following expression, which gives the weighted sum of squared distances of a point $\boldsymbol{P}$ to $\langle \mathbf{A}_1, \boldsymbol{O}_1 \rangle, \ldots, \langle \mathbf{A}_n, \boldsymbol{O}_n \rangle$

$$\begin{aligned} E_{point}(\boldsymbol{P}) &= \sum_{i=1}^{n} (\boldsymbol{P} - \boldsymbol{O}_i)^T \mathbf{H}_i \mathbf{W}_i \mathbf{H}_i (\boldsymbol{P} - \boldsymbol{O}_i) \\ &= \boldsymbol{P}^T \mathbf{H} \boldsymbol{P} - 2\mathbf{O}^T \boldsymbol{P} + c \end{aligned} \tag{5}$$

where

$$\mathbf{H} = \sum_{i=1}^{n} \mathbf{H}_i \mathbf{W}_i \mathbf{H}_i \tag{6}$$

$$\mathbf{O} = \sum_{i=1}^{n} \mathbf{H}_i \mathbf{W}_i \mathbf{H}_i \boldsymbol{O}_i \tag{7}$$

$$c = \sum_{i=1}^{n} \boldsymbol{O}_i^T \mathbf{H}_i \mathbf{W}_i \mathbf{H}_i \boldsymbol{O}_i \tag{8}$$

The optimal point $\bar{\boldsymbol{P}}$ is found by differentiating Eq. (5) and setting the result to 0. After slight rearrangement, this minimization yields the following 3×3 linear system whose solution is the optimal reconstructed scene point:

$$\mathbf{H}\bar{\boldsymbol{P}} = \mathbf{O} \tag{9}$$

In the case of line features and ideal measurements, Eq. (9) yields a one-parameter family of solutions that spans the reconstructed line. In general, the residual error of a 3D line segment $L = \{\boldsymbol{P} + t\mathbf{D} \mid t_1 \le t \le t_2\}$ from a set of image features is evaluated by integrating Eq. (5):

$$E_{line}(L, t_1, t_2) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} E_{point}(\boldsymbol{P} + t\mathbf{D})\, dt$$

E_{line} is minimized when $\boldsymbol{P}$ satisfies Eq. (9) and $\mathbf{D}$ is the eigenvector of $\mathbf{H}$ with the smallest eigenvalue. If we denote λ_3 as the smallest eigenvalue of $\mathbf{H}$ and $\mathbf{V}_3$ as its associated eigenvector, the optimal reconstructed line in the scene and its expected error are given by

$$\begin{aligned} \bar{L} &= \{\bar{\boldsymbol{P}} + t\mathbf{V}_3 \mid t \in \Re\} \\ E_{line}(\bar{L}, t_1, t_2) &= E_{point}(\bar{\boldsymbol{P}}) + \alpha\lambda_3 \end{aligned} \tag{10}$$

where $\alpha = \frac{1}{3}(t_2^2 + t_2 t_1 + t_1^2)$. In practice, α can be estimated from the endpoints of one or more corresponding image features, or be fixed as a user-specified parameter. In our experiments, we chose $E_{line} = \lambda_3$ which effectively sets $\alpha = \infty$. This choice of E_{line} takes into account the error in the direction of $\bar{L}$ but not its offset.

The correspondence method of the last section generates a collection of image feature correspondences (bins), each of which votes for a possible feature in the scene. By using these bins to accumulate structure information as well as correspondence votes, the processes of structure recovery and feature correspondence are integrated. Each set of features determines a scene feature and a residual error. The information required to compute these quantities is stored in a bin:

BIN

H	– 3×3 projection matrix
O	– 3×1 offset vector
c	– auxiliary error term

This formulation works well in a recursive framework, where new measurements are incorporated incrementally into each bin, using Eqs. (6 - 8). In this respect, our approach is similar to methods based on the Kalman filter [4, 11] which is a popular tool for solving linear equations recursively. The primary advantage of our approach is the *implicit* formulation of structure which treats points and lines uniformly. Observe that each bin implicitly represents either a 3D point or a line. In contrast, previous approaches based on the Kalman filter employed an *explicit* description of the degrees of freedom of the reconstructible subspace, which differs for points and lines. In particular, points require a three parameter representational space and lines require a minimum of four parameters [11]. In contrast, an implicit formulation permits reconstruction of both points and lines within a common three-dimensional parameter space. Computationally this means that the implicit framework is more efficient both because of the reduced complexity and because the Kalman filter requires n matrix inversions whereas the implicit equations require only one.

5 Preprocessing Optimizations

Two optimizations are described that increase the efficiency of the algorithm by preprocessing the images. The first technique rectifies the images so that epipolar lines are horizontal. The second technique caches proximity information from the rectified images so that feature correspondences can be found by table lookup.

5.1 Image Rectification

The run time of determining feature correspondences is dominated by the cost of searching along epipolar lines. The task is ameliorated by appropriately transforming the images or edge maps so that epipolar lines are specially aligned. This technique, known as *image rectification*, has been previously applied to perspective imagery to simplify matching for binocular and trinocular stereo [11]. For perspective cameras in general position, image-rectification involves a non-linear image transform[1].

Under an affine or orthographic projection model, all epipolar lines in each image are parallel so the rectification process is simplified considerably. Each image is merely rotated so that epipolar lines are horizontal; skews and shears are unnecessary. The angle of rotation is determined by the direction of the epipolar lines in the image to be rectified. This direction is given by the vector $\boldsymbol{d}_i = [x\ y]^T$ in Eq. (2) and the rectification angle θ_i is

$$\theta_i = -arctan(\frac{y}{x})$$

Image I_i, $i = 2, \ldots, n$, is rectified by transforming each feature by the matrix

$$\mathbf{R}_i = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i) \\ sin(\theta_i) & cos(\theta_i) \end{bmatrix}$$

[1]The projective rectification transform is linear in projective-space but not in image-space.

Image rectification affects the projection matrices as well. The projection matrix of each rectified image is modified by $\mathbf{\Pi}_i = \mathbf{R}_i\mathbf{\Pi}_i$ after which it is of the form

$$\mathbf{\Pi}_i = \begin{bmatrix} a & b & c \\ d & e & 0 \end{bmatrix}$$

After rectification, epipolar computations are simplified as follows: The epipolar line in image I_i of a point $\boldsymbol{p} = [x_p\ y_p]^T$ in image I_1 is the horizontal scanline whose y-coordinate is given by:

$$y = dx_p + ey_p \tag{11}$$

Similarly, let $L^p = \{[x\ y\ Z]^T \mid Z \in \Re\}$ and let $\boldsymbol{q} = [x_q\ y_q]^T$ be a point along l_i^p. The line L^q voted for by $\boldsymbol{q}$ intersects L^p at the point $[x\ y\ Z_q]^T$ where

$$Z = \frac{x_q - ax_p - by_p}{c} \tag{12}$$

5.2 Interval Table

The search for feature correspondences can be reduced to a table lookup by precomputing the features that are sufficiently close to each scanline. Each rectified image is transformed to a set of interval lists, one per scanline. Each list of intervals specifies the set of all possible correspondences of every feature in I_1 having a given epipolar line in I_i.

The interval table is computed as follows: the set of features that are within a tolerated vertical distance of each scanline is found using a distance transform [14]. For each such feature, an interval is created for the region of the scanline within tolerance of that feature. A table is constructed that maps each scanline to the list of its intervals that are sufficiently close to an image feature. The result is a table that gives all the possible correspondences in an image for each feature in I_1. The table is indexed using Eq. (11). To reconstruct both point and line features, two interval tables are needed.

The method assumes that estimated measurement covariances are constant for features within each image. Let $\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_n$ be the 2×2 covariance matrices for images $I_1, \ldots, I_n$, respectively. These quantities are typically provided by feature detectors and affine calibration techniques.

Under predictable sources of error, corresponding points will stray from epipolar lines by a predictable amount. The confidence that $\boldsymbol{p}$ and $\boldsymbol{q}$ correspond is determined by estimating the covariance about l_i^p, defined by

$$\mathbf{\Lambda}_i^l = \mathbf{\Pi}_i\mathbf{\Pi}_1^{-1}\mathbf{\Lambda}_1\mathbf{\Pi}_1^{-1^T}\mathbf{\Pi}_i^T$$

The vertical variance of $\boldsymbol{q}$ from l_i^p is

$$\lambda_i^l = [0\ 1](\mathbf{\Lambda}_i^l + \mathbf{\Lambda}_i)[0\ 1]^T \tag{13}$$

If feature measurements are assumed to be Gaussian, λ_i^l has a χ^2 distribution with two degrees of freedom. A suitable tolerance threshold may be selected by consulting a χ^2 table.

6 The Algorithm

The complete algorithm for determining the structure of a scene from a sequence of two or more uncalibrated images is presented in this section. It is assumed that at least four image features have been detected and matched in each image.

Scene Reconstruction Algorithm

1. Preprocess the images: Detect point and line features, determine projection matrices, rectify, and compute interval tables. This step can be performed incrementally.

2. Choose features in image I_1. For each feature $\boldsymbol{p}$, create an initial bin $B^p_{-\infty,\infty}$ using the point or line formula, as appropriate.

3. Get the next image, I_i, and obtain the set of possible matches using the interval table. Transform the intervals to affine depth intervals (Z_1^q, Z_2^q) using Eq. (12).

4. For each feature $\boldsymbol{q}$ in I_i and bin $B^p_{Z_1,Z_2}$ such that $(Z_1^q, Z_2^q) \bigcap (Z_1, Z_2) \neq \emptyset$, create a new bin, $B^p_{Z_1',Z_2'}$, where $(Z_1', Z_2') = (Z_1^q, Z_2^q) \bigcap (Z_1, Z_2)$. The bin fields, $\mathbf{H}$, $\mathbf{O}$, and c, are set to the sums of the respective fields of bins $B^p_{Z_1,Z_2}$ and $B^q_{-\infty,\infty}$. The new bins replace the previous set.

5. For each feature $\boldsymbol{p}$ in image I_1 with at least one bin, choose the bin $B^p_{Z_1,Z_2}$ with minimum error, using Eq. (5) or (10), and compute the 3D coordinates of the corresponding scene feature.

6. Go to Step 3 or stop if the image sequence is exhausted.

Although the algorithm is dominated by Step 5, this step need not be performed at every iteration because the 3D structure is implicitly stored in the set of bins. Rather, the scene structure can be periodically updated after a block of images is processed. For instance, a *batch* version of the above algorithm would have Step 5 performed last.

To avoid a combinatorial increase in the number of bins, a monotonicity constraint is added to Step 4 requiring that each bin $B^p_{Z_1,Z_2}$, $-\infty < Z_1, Z_2 < \infty$, be replaced with at most one other bin. Specifically, if multiple features $\boldsymbol{q}^1, \ldots, \boldsymbol{q}^m$ in I_i all satisfy $(Z_1^{q^i}, Z_2^{q^i}) \bigcap (Z_1, Z_2) \neq \emptyset$, a single bin is created for the feature $\boldsymbol{q}^j$ with minimal error. The result is that a feature may have at most m bins, where m is the maximum number of features in an image.

To evaluate the complexity of the algorithm, we assume a constant number of image features, m. Each feature may have at most m bins and each bin requires $O(m)$ operations per image due to the monotonicity constraint. With n images, the worst-case complexity of the algorithm is therefore $O(nm^3)$, not including preprocessing. Computation of the interval table requires a series of simple image transformations such as rotations and distance transforms that can be performed efficiently in hardware. The only remaining cost is due to tracking the reference features and finding the projection matrices. The complexity of the latter task is $O(nk^2)$ for k reference features.

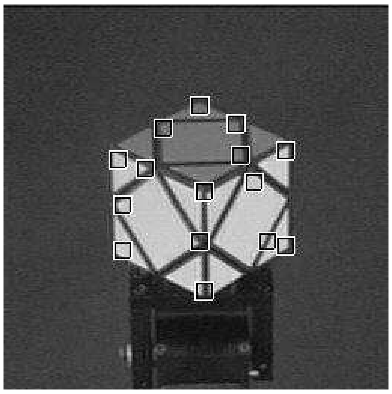

Figure 1: Set of features tracked in cube sequence.

7 Experiments

7.1 Rotating Cube Sequence

A variant of a Rubik's Cube was placed on a pantilt device and filmed while it was undergoing a 23 degree rotation about one axis and a 45 degree rotation about another. Several point-features were selected and automatically tracked through a sequence of 83 images. A few features that did not lie on the cube were manually deleted from the feature set. The first image is shown in Figure 1 with reference features marked. Images 26 through 50 were set aside as a test set and not used in the structure recovery procedure. This sequence contains several sets of closely-spaced parallel lines, making feature-tracking difficult using traditional correlation-based methods.

Edges were detected in every image and the algorithm was run on the line segments in images 1-25 and 51-83. To verify that the reconstruction was accurate, we reprojected the recovered lines and compared the reconstructed and original images. To determine line endpoints, each line was clipped so that its projection matched the corresponding edge in image 1 as closely as possible.

Fig. 2 shows the results. Qualitatively, it is apparent that the reconstructed images match the original images quite well. The figure illustrates the ability of the method to reconstruct views (top) and to predict novel views (bottom). In addition, the ability to eliminate outliers is demonstrated. Notice that the detected line segments (left) each contain several lines that are not on the cube. These are features on the pantilt head that did not rotate with the cube and were automatically filtered out as outliers. Also note that certain lines were filtered out that should not have been. The loss of lines can be attributed to edges that were not detected in one or more images. Each im-

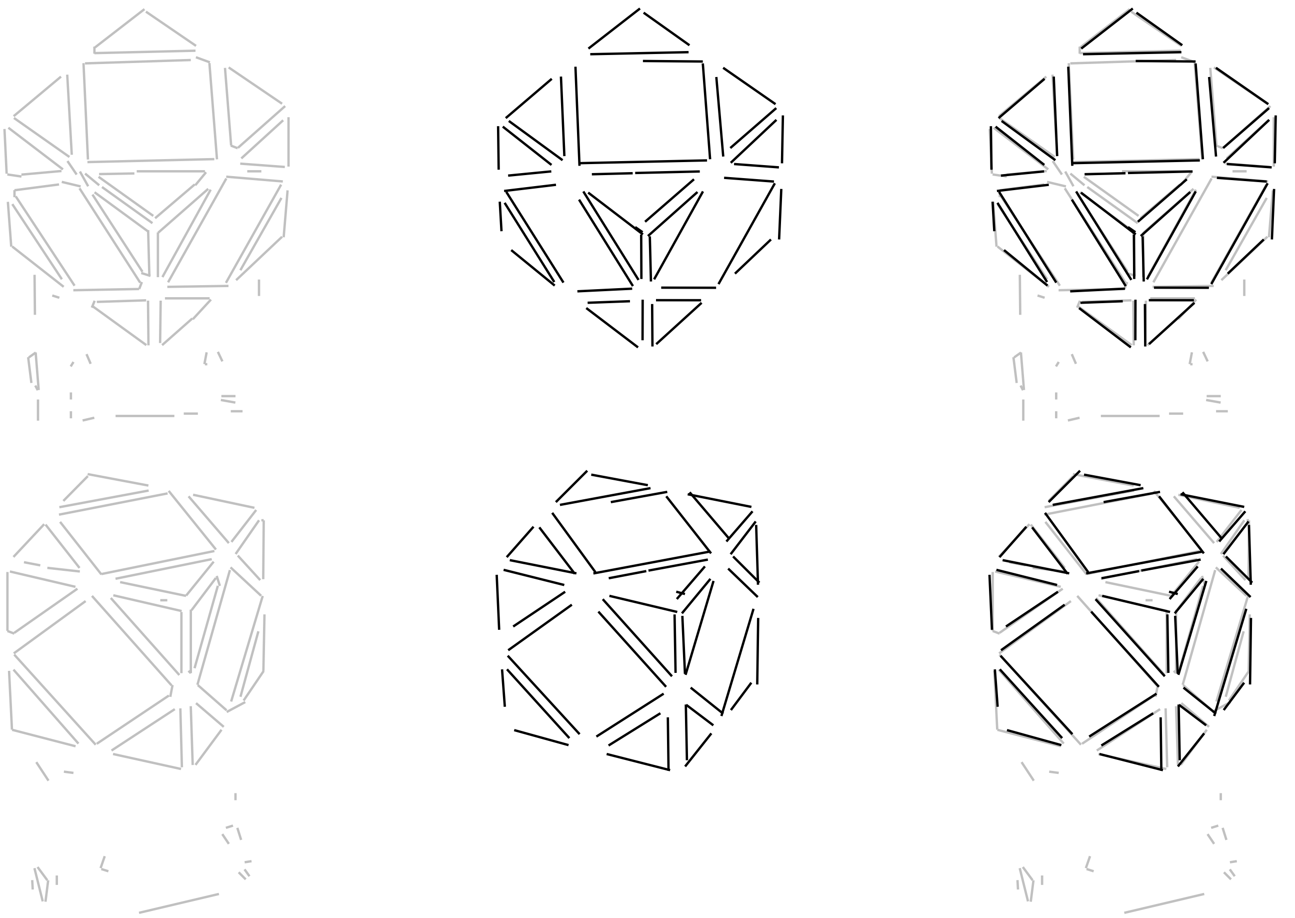

Figure 2: Reconstructed projections of a cube. Left: Detected line segments for image 20 (top) and 40 (bottom). Center: Reconstructed images created by reprojecting the recovered lines. Right: Overlay of the original and reconstructed lines. Note that outliers were automatically removed (center).

Figure 3: Reconstructed affine structure of a building. Left: Frontal view from below. Right: Top view.

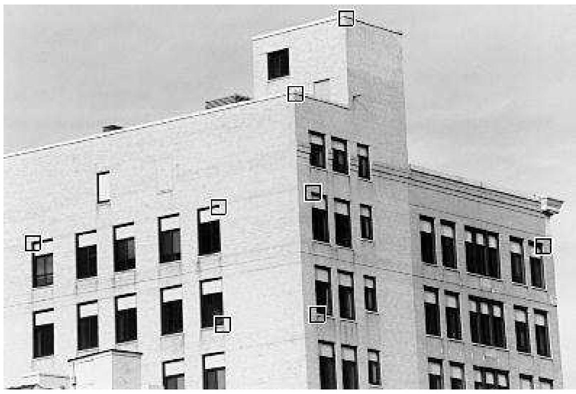

Figure 4: Set of features tracked in building sequence.

age contained roughly 70-100 line features. The algorithm found correspondences and computed reconstructed lines at a rate of approximately 12 frames per second on a Sun SPARC 10, not including image preprocessing and file I/O.

7.2 Images of Outdoor Building Scene

In this section we present a case where conventional structure-from-motion techniques are not viable due to the extreme difficulty of obtaining many feature correspondences. Eight photographs of a bank building were taken from varying viewpoints. The images were processed in an arbitrary order and there is no coherence from one image to another that could be used to ameliorate feature correspondence. Eight corresponding reference features were manually selected in each image (shown in Fig. 4). The correspondences of all the reconstructible line segments were found by our algorithm, which took roughly 20 seconds for 8 frames, each containing 300-450 line segments. The slower runtime reflects a five-fold increase in the number of features, with respect to the cube sequence.

Fig. 3 shows the reconstructed affine structure of the building, shown in appropriate reference frames. Notice that the reconstructed walls are not perpendicular, whereas the walls of the actual building meet at right angles. This is not an error on the part of the algorithm, but rather an artifact of working in an affine reference frame.

Note that only lines that were detected in every image were reconstructed, so several edges were dropped, particularly those near image borders. This indicates that the algorithm is sensitive to occlusions and features missed during edge detection. Handling of temporary occlusions is possible in this framework by slightly generalizing the notion of *reconstructible feature* so that representation in every image is not required. This is a topic of future work.

8 Conclusions

We have presented a unified approach for solving the problems of feature correspondence and robust structure recovery of point and line features. The algorithm departs from previous work by providing guarantees on the completeness and consistency of the recovered scene. By solving for feature correspondences and structure simultaneously, all possible explanations of the scene can be accounted for. This framework eliminates the need for smooth camera motion and provides automatic removal of outliers based on incompatibility with a rigid motion. The algorithm is recursive and its accuracy is demonstrated with real image sequences.

Future plans include a real-time implementation of the algorithm to permit complete structure recovery online. We also plan to extend the framework to allow reconstruction of features that become occluded and to investigate generalizing the method to work with a projective camera model.

References

[1] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th International Joint Conference on Artificial Intelligence*, 1981.

[2] I. J. Cox, "A review of statistical data association techniques for motion correspondence," *Intl. Journal of Computer Vision*, vol. 10, no. 1, pp. 53–66, 1993.

[3] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *Intl. Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[4] L. Shapiro, A. Zisserman, and M. Brady, "Motion from point matches using affine epipolar geometry," in *Proc. Third European Conference on Computer Vision*, pp. 73–84, 1994.

[5] K. N. Kutulakos and C. R. Dyer, "Global surface reconstruction by purposive control of observer motion," in *Proc. Computer Vision and Pattern Recognition*, pp. 331–338, 1994.

[6] J. J. Koenderink and A. J. van Doorn, "Affine structure from motion," *Opt. Soc. Am. A*, vol. 8, pp. 377–385, 1991.

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

[8] O. D. Faugeras, Q.-T. Luong, and S. J. Maybank, "Camera self-calibration: Theory and experiments," in *Proc. European Conference on Computer Vision*, pp. 321–334, 1992.

[9] C.-H. Lee and T. Huang, "Finding point correspondences and determining motion of a rigid object from two weak perspective views," *Computer Vision, Graphics, and Image Processing*, vol. 52, pp. 309–327, 1990.

[10] R. Basri, "On the uniqueness of correspondence under orthographic and perspective projections," in *Proc. Image Understanding Workshop*, pp. 875–884, 1992.

[11] N. Ayache and F. Lustman, "Trinocular stereo vision for robotics," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 1, pp. 73–85, 1991.

[12] T. Morita and T. Kanade, "A sequential factorization method for recovering shape and motion from image streams," Tech. Rep. CMU-CS-94-158, Carnegie Mellon University, Pittsburgh, PA, June 1994.

[13] G. Strang, *Linear Algebra and its Applications*. San Diego, CA: Harcourt Brace Jovanovich Inc., 1988.

[14] G. Borgefors, "Distance transformations in arbitrary dimensions," *Computer Vision, Graphics, and Image Processing*, vol. 27, pp. 321–345, 1984.