# Scalable Extrinsic Calibration
# of Omni-Directional Image Networks

MATTHEW ANTONE AND SETH TELLER

*Computer Graphics Group, MIT Lab for Computer Science, Technology Square, Cambridge, MA 02139*

tone@graphics.lcs.mit.edu      seth@graphics.lcs.mit.edu

**Abstract.**  We describe a linear-time algorithm that recovers absolute camera orientations and positions, along with uncertainty estimates, for networks of terrestrial image *nodes* spanning hundreds of meters in outdoor urban scenes. The algorithm produces pose estimates globally consistent to roughly $0.1°$ (2 milliradians) and 5 centimeters on average, or about four pixels of epipolar alignment.

We assume that adjacent nodes observe overlapping portions of the scene, and that at least two distinct vanishing points are observed by each node. The algorithm decouples registration into pure rotation and translation stages. The rotation stage aligns nodes to commonly observed scene line directions; the translation stage assigns node positions consistent with locally estimated motion directions, then registers the resulting network to absolute (Earth) coordinates.

The paper's principal contributions include: extension of classic registration methods to large scale and dimensional extent; a consistent probabilistic framework for modeling projective uncertainty; and a new hybrid of Hough transform and expectation maximization algorithms.

We assess the algorithm's performance on synthetic and real data, and draw several conclusions. First, by fusing thousands of observations the algorithm achieves accurate registration even in the face of significant lighting variations, low-level feature noise, and error in initial pose estimates. Second, the algorithm's robustness and accuracy increase with image field of view. Third, the algorithm surmounts the usual tradeoff between speed and accuracy; it is both faster and more accurate than manual bundle adjustment.

**Keywords:**  Exterior orientation, egomotion, structure from motion, panoramas

## 1   Introduction

Calibrated imagery is of fundamental interest in a variety of computer vision and graphics applications, including sensor fusion, 3D reconstruction for model capture, and image-based rendering for realistic visual simulation. In practice, image registration can require substantial manual effort, for example specification of matching tie points across multiple images as constraints for a bundle adjustment algorithm. Even for small datasets, this manual component can absorb tens or hundreds of hours of human effort, and is difficult or impossible to partition among several workers.

The algorithm in this paper was developed as part of a system for automated geometric model capture in urban environments [Tel97, Tel01]. In this system, a human operator moves a sensor [BdT99] to widely-separated vantage points in and around the scene of interest. At each position, the sensor acquires a high-resolution image, along with a rough estimate of the acquiring camera's 6-DOF pose, or position and orientation, in absolute (Earth) coordinates (Fig. 1).

Images are grouped by optical center into wide-FOV mosaics called "nodes" [CMT98]. Each node is subsequently treated as a rigid, super-hemispherical image with a single 6-DOF pose. The use of wide-FOV imagery provides significant
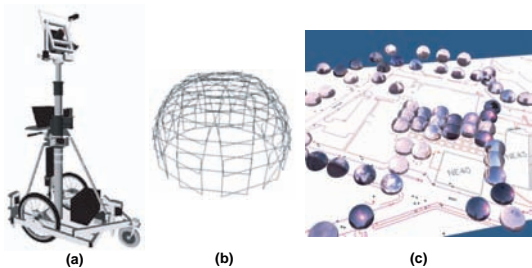
*Figure 1.* Pose-Image Acquisition. (a) Our prototype sensor, which acquires geo-referenced images. (b) A typical omni-directional image tiling. (c) Node locations registered with a ground map.

advantages in practice, both reducing the number of optimization parameters and eliminating classical bias [Kan93] and ambiguities [FA98] in camera motion estimation.

The sensor's initial pose estimates are accurate only to a few degrees and a few meters. Since detailed 3-D reconstruction requires accurate epipolar geometry, one critical component of our system is the refinement of the sensor's initial pose estimates to bring all nodes into registration. The dataset size and extent rules out interactive techniques; thus pose recovery must be automated, and its computational cost must scale well with the number of images.

Solving the general registration problem requires determining six parameters for each node: three of rotation and three of position. Our approach decouples the 6-DOF problem into a pure rotation (3-DOF) component followed by a pure translation (3-DOF) component. Our algorithm cannot assume that common scene structure is observed by all images; indeed, due to occlusion, most image pairs observe nothing in common. Instead, we use the (rough) initial pose estimates to associate nodes likely to have observed overlapping scene structure, then use an efficient local-to-global alignment strategy to register the entire network.

### 1.1  Algorithm Overview

The goal of our algorithm is to accurately register every node to a single, common coordinate system. Intuitively, the algorithm detects common scene structure observed by clusters of adjacent (nearby) nodes, exploiting the tendency of such nodes to have observed overlapping scene structure. Each node is aligned to this locally observed structure,

after which a global optimization brings all nodes into registration.

More formally, the algorithm operates in two sequential stages. The first, rotational alignment, classifies scene lines into parallel sets, and from these constructs position-invariant *vanishing points*. An expectation maximization (EM) algorithm, based on a projective mixture model and initialized by a Hough transform, estimates the vanishing points observed by each node. A subsequent EM formulation probabilistically aligns node vanishing points with scene-relative directions and recovers global orientation and uncertainty for each node.

The algorithm's second stage, position recovery, putatively couples point features across adjacent node pairs, then uses a Hough transform to extract a crude estimate of the inter-node motion direction or *baseline* for each pair. A Monte-Carlo expectation maximization (MCEM) technique based on a projective uncertainty model refines each baseline estimate. A global optimization phase assembles the local baseline estimates into a network-wide constraint set, propagates the constraints to produce globally consistent node positions, and rigidly transforms the resulting network to be maximally consistent with the initial sensor pose estimates.

### 1.2  Requirements and Assumptions

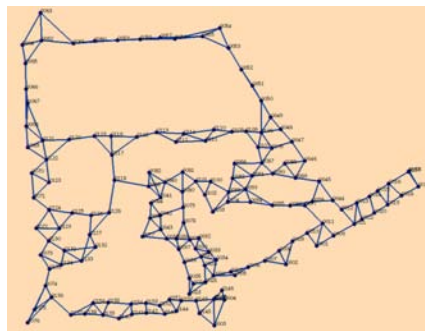To register a set of images, our algorithm requires the following inputs:



*Figure 2.* A Node Network. Points represent nodes; edges represent adjacency.

- **Accurate intrinsic calibration.** Images have been corrected for radial distortion, and pinhole camera parameters (i.e. focal length, principal point, skew) are given.

- **Rough extrinsic pose.** Approximate position and orientation estimates for each node are supplied by the sensor.

- **Node adjacency.** A list of the neighbors of each node is given, associating nodes likely to have viewed overlapping portions of the scene. The adjacency network is inferred from the sensor's GPS-based position estimates.

- **Line and point features.** Sub-pixel gradient-based line features are supplied for each image by a modified Canny edge detector [Can86]. Point features are inferred from intersections of nearby line pairs.

In practice, registration succeeds when the following conditions hold:

- **Visible vanishing points.** At least two distinct vanishing points (VPs) are visible in each wide-FOV node. These provide a local, translation-invariant coordinate frame for each node.

- **Overlapping scene geometry.** Nodes are acquired with sufficient density so that adjacent nodes observe overlapping scene geometry (namely 3-D lines and points). We do not make a small-baseline assumption; in practice, the inter-node baselines are relatively wide (tens of meters).

- **Wide-FOV nodes.** Our algorithm can be applied to images with any FOV. Wide-FOV images, however, are fundamentally more powerful than conventional images; they provide maximal observations of surrounding structure, disambiguate small rotations from small translations, reduce bias in inference, and in general enable more reliable convergence and higher accuracy.

### 1.3   Paper Overview

The remainder of the paper is structured as follows. Section 2 reviews projective feature representations and geometric probability. Section 3 describes orientation recovery. Section 4 describes metric position recovery. Section 5 reports results for several synthetic and real datasets. Section 6 reviews previous work on image registration. Section 7 summarizes the paper's contributions, and Section 8 concludes.

## 2   Preliminaries

This section reviews representations of coordinate transformations and uncertain projective features.

### 2.1   Extrinsic Pose

A rigid transformation, consisting of a $3 \times 1$ translation $\boldsymbol{t}$ and orthonormal $3 \times 3$ rotation $\boldsymbol{R}$ (or, equivalently, a unit quaternion $\boldsymbol{q}$), expresses points $\boldsymbol{p}^s$ in scene coordinates as points $\boldsymbol{p}^c$ in camera coordinates. Its inverse specifies the orientation and position of the camera with respect to the scene coordinate system. Formally,

$$
\begin{aligned}
\boldsymbol{p}^c &= \boldsymbol{R}^{\top}(\boldsymbol{p}^s - \boldsymbol{t}) \\
\boldsymbol{p}^s &= \boldsymbol{R}\boldsymbol{p}^c + \boldsymbol{t}
\end{aligned}
$$

where $\boldsymbol{t}$ is the position of the focal point, and the columns of $\boldsymbol{R}$ are the principal camera axes, both expressed in scene coordinates. Thus $\boldsymbol{t}$ and $\boldsymbol{R}$ summarize the external pose of the camera.

### 2.2   Projective Features

In the Euclidean image plane, we represent points as coordinate pairs $(u, v)$, and lines as $au + bv + c = 0$, or equivalently $\boldsymbol{p} \cdot \boldsymbol{l} = 0$, where $\boldsymbol{p} = (u, v, 1)^{\top}$ and $\boldsymbol{l} = (a, b, c)^{\top}$.

Although the Euclidean plane is convenient for feature detection, it cannot stably represent a full hemisphere of rays. Thus to represent line features we use the *projective plane* $\mathbb{P}^2$, a closed topological manifold containing all 3-D lines through the focal point. Excepting the focal point, points along any 3-D line constitute an equivalence class $\sim$:

$$
\boldsymbol{p} \sim \boldsymbol{r} \quad \Leftrightarrow \quad \boldsymbol{p} = \alpha \boldsymbol{r},
$$

where $\alpha$ is a real nonzero scalar value. We represent directions as points on the surface of the unit sphere $\mathbb{S}^2$. The sphere's surface is an ideal space for representation of projective features, just as it is an ideal space for image projection: it is closed, compact, and symmetric, and it provides uniform treatment of rays from all directions (Fig. 3).

### 2.3   Bingham's Distribution

Features viewed by a single camera have unknown depth; these projective features must be represented with suitable spherical probability distributions. Exponential distributions are useful for inference tasks [Ber79], but the most commonly used multi-variate Gaussian density is a
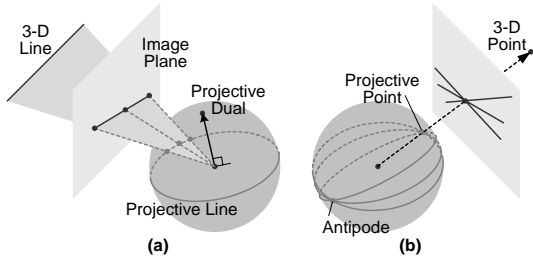
*Figure 3.* Projective Image Features. (a) A 3-D line can be represented by a 2-D line in planar projection or a great circle in spherical projection. Any point on the line must be orthogonal to the line's dual. (b) A 3-D point can be represented as a unit vector on the sphere, or as a pencil of lines passing through its projection.

Euclidean probability measure and is therefore not suitable for projective variables. Conditioning a zero-mean Gaussian variable $\boldsymbol{x} \in \mathcal{R}^3$ on $\|\boldsymbol{x}\| = 1$ results in *Bingham's distribution*, a flexible exponential density defined on the unit sphere [Bin74, JM79, Wat83].

This distribution can be generalized to arbitrary dimension $n$, and is parameterized by a symmetric $n \times n$ matrix $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\kappa}\boldsymbol{U}^\top$, analogous to the information matrix of a Gaussian [Riv84], where $\boldsymbol{U} \in \mathbb{R}^{n \times n}$ is a real unitary matrix whose columns $\boldsymbol{u}_i$ represent the principal directions of the distribution and $\boldsymbol{\kappa} \in \mathbb{R}^{n \times n}$ is a diagonal matrix of $n$ concentration parameters $\kappa_i$. The density is

$$
\begin{aligned}
p(\boldsymbol{x}) &= \frac{1}{c(\boldsymbol{\kappa})} \exp(\boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{x}) \\
&= \frac{1}{c(\boldsymbol{\kappa})} \exp\left(\sum_{i=1}^{n} \kappa_i (\boldsymbol{u}_i^\top \boldsymbol{x})^2\right)
\end{aligned}
$$

where $c(\boldsymbol{\kappa})$ is a normalizing coefficient depending only on the concentration parameters. We denote this density by $\mathcal{B}_n(\boldsymbol{x}; \boldsymbol{\kappa}, \boldsymbol{U})$ or simply $\mathcal{B}_n(\boldsymbol{x}; \boldsymbol{M})$.

The Bingham density is antipodally symmetric, or *axial*: the probability of any point $\boldsymbol{x}$ is identical to that of $-\boldsymbol{x}$. It is closed under rotations: if $\boldsymbol{y} = \boldsymbol{R}\boldsymbol{x}$, where $\boldsymbol{R}$ is a rotation matrix and $\boldsymbol{x}$ has Bingham distribution $\mathcal{B}_n(\boldsymbol{x}; \boldsymbol{\kappa}, \boldsymbol{U})$, then $\boldsymbol{y}$ has a Bingham distribution $\mathcal{B}_n(\boldsymbol{y}; \boldsymbol{\kappa}, \boldsymbol{R}\boldsymbol{U})$. It is also closed under Bayesian inference: fusion of Bingham-distributed data in $\mathcal{R}^n$ produces a Bingham-distributed aggregate in $\mathcal{R}^n$. Finally, the Bingham representation is expressive: the concentration parameters can describe a wide variety of distributions (Fig. 4).
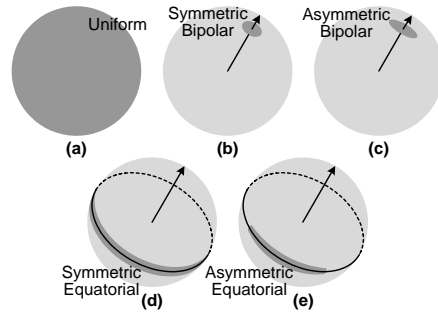


*Figure 4.* Bingham's Distribution on the Sphere. The shapes of iso-density contours on Bingham's distribution depend on the concentration parameters. For $\mathcal{B}_3$, the two shape parameters yield distributions which are (a) uniform ($\kappa_1 = \kappa_2 = 0$); (b) symmetric bipolar ($\kappa_1 = \kappa_2 \ll 0$); (c) asymmetric bipolar ($\kappa_1 < \kappa_2 \ll 0$); (d) symmetric equatorial ($\kappa_1 \ll \kappa_2 = 0$); and (e) asymmetric equatorial ($\kappa_1 \ll \kappa_2 < 0$).

The concentration parameters are unique only up to an additive shift; in other words, the density is unchanged if a single constant is added to all parameters. By convention, the parameters (along with their corresponding modal directions $\boldsymbol{u}_i$) are ordered from smallest to largest, and shifted by an additive constant so that

$$
\kappa_1 \leq \kappa_2 \leq \ldots \leq \kappa_n = 0.
$$

It is possible to transform the Euclidean sample covariance into a spherical Bingham parameter matrix and vice versa [JM79]. Given deterministic, unit-length data points $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k\}$, the maximum likelihood estimates of the underlying Bingham distribution parameters are related to the sample second moment matrix

$$
\boldsymbol{S}_{\boldsymbol{x}} = \frac{1}{k} \sum_{i=1}^{k} \boldsymbol{x}_i \boldsymbol{x}_i^\top. \tag{1}
$$

If the matrix is diagonalized into $\boldsymbol{S}_{\boldsymbol{x}} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix of the eigenvalues, then $\boldsymbol{U} = \boldsymbol{V}$ (that is, the principal directions of the Bingham distribution are exactly the eigenvectors of $\boldsymbol{S}_{\boldsymbol{x}}$), and the concentration matrix $\boldsymbol{\kappa}$ is an invertible function of $\boldsymbol{\Lambda}$.

# 3   Orientation Recovery

This section describes the orientation recovery algorithm. Section 3.1 reviews vanishing point geometry. Section 3.2 presents a novel method that robustly detects, then accurately estimates, multiple vanishing points in a single node. Section 3.3 extends a classical, deterministic algorithm for rotationally registering two nodes to account for (input) feature and (output) orientation uncertainty. Finally, Section 3.4 describes an EM algorithm to classify vanishing points, estimate scene-relative line directions, and refine rotations over a node network.

## 3.1   *Vanishing Point Geometry*

Parallel 3-D lines viewed under perspective converge to an apparent point of intersection known as a *vanishing point* (VP). Vanishing points have long been used in vision to extract information about scene geometry and egomotion.

Consider a 3-D line parallel to some unit direction $v$, and its 2-D projection on the image surface (Fig. 5). The two quantities are projectively equivalent; that is, any projective ray that intersects the image line also intersects the scene line. The set of all such rays thus forms a plane $\mathcal{P}$ that includes the focal point, the 2-D line, and the original 3-D line. Let $l$ represent the projective dual of the line, that is the direction on the sphere orthogonal to all rays through the image line. Since by construction $l$ is orthogonal to $\mathcal{P}$, it must also be orthogonal to the 3-D line; that is, $l \cdot v = 0$.
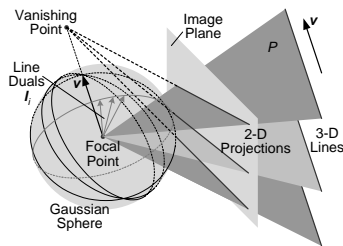


*Figure 5.* Vanishing Point Geometry. Projections of parallel 3-D lines converge to an apparent vanishing point in the plane. Projectively, the vanishing point represents the intersection of a pencil formed by observations of the lines.

Similarly, *any* 3-D line parallel to $v$ has a projective dual representation $l_i$ for which $l_i \cdot v = 0$.

The direction $v$ is thus the normal to a plane containing all such dual rays $l_i$ (Fig. 5). Because of the projective equivalence between scene lines and image lines, 2-D observations alone suffice for this construction; thus, $v$ can be recovered from a set of image lines if their associated 3-D lines are known to be parallel (Section 3.2.3).

Since vanishing points lie at infinity, they are invariant to local node translations. This implies that rotational error can be corrected independently of positional error by aligning locally-observed VPs, which represent scene-relative 3-D line directions.

## 3.2   *Detecting VPs in One Node*

Our VP detection algorithm takes image lines, represented by projective random variables $x_i$, as input. However, the collection of lines $\mathcal{X}$ in a given node is initially unclassified; that is, lines are not grouped into parallel sets, and outliers (arising from visual clutter such as foliage, cars, and people) are mixed with the lines of interest. The problem of vanishing point estimation thus has three components (Fig. 6). First, the number of groups $J$ (that is, the number of prominent 3-D line directions) must be established. Next, lines $x_i$ must be classified according to their corresponding 3-D direction or discarded as outliers. Finally, the vanishing point $v_j$ for each group must be estimated.

These three problems are tightly coupled, in that given a deterministic classification of all line features, the estimation problem reduces to a collection of $J$ isolated projective inference tasks, one for each line group. Similarly, given a set of $J$ 3-D directions $v_j$, line classification amounts to evaluating a similarity metric between lines and directions.

The expectation maximization (EM) algorithm [DLR77] is a powerful tool for parameter estimation from incomplete or unclassified data. This section presents an EM formulation for simultaneous line classification and vanishing point estimation as inference problems on the sphere. For the moment, it is assumed that the algorithm is appropriately initialized; that is, the number $J$ of prominent vanishing points is known, and an approximate direction is known for each. Section 3.2.4 describes an efficient Hough transform technique that determines these quantities.
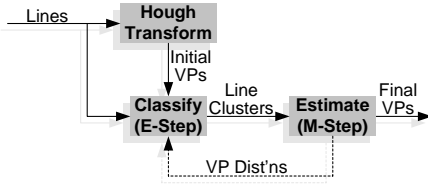
*Figure 6.* Vanishing Point Estimation. Stochastic line features from a single image are obtained from the fusion of gradient pixel distributions. These lines are used in a Hough transform, which finds prominent 3-D line directions to initialize an expectation maximization algorithm. The result is a set of accurate vanishing points and a line classification.

**3.2.1    Mixture Model.** Figure 5 shows that vanishing points are projective quantities constrained by the dual points of contributing projective lines. Thus each of the $J$ observed vanishing points $\boldsymbol{v}_j$ is modeled as a Bingham random variable with unknown parameter matrix $\boldsymbol{M}_j^{\boldsymbol{v}}$, formed by fusion of appropriate uncertain line features. The entire dataset $\mathcal{X}$ is a collection of unclassified samples from the set of random variables $\mathcal{V} = \{\boldsymbol{v}_0 \ldots , \boldsymbol{v}_J\}$, where $\boldsymbol{v}_0$ represents an unknown outlier distribution; thus, $\mathcal{X}$ is modeled by a mixture of $J + 1$ Bingham densities $p(\boldsymbol{x}_i|j, \mathcal{V})$, so that

$$p(\boldsymbol{x}_i|\mathcal{V}) = \sum_{j=0}^{J} p(\boldsymbol{x}_i|j, \mathcal{V})p(j|\mathcal{V}),$$

where $p(j|\mathcal{V})$ is a prior probability representing the fraction of observations generated by $\boldsymbol{v}_j$. Each observation $\boldsymbol{x}_i$ represents an uncertain line feature with known equatorial Bingham distribution $\mathcal{B}_3(\boldsymbol{x}_i; \boldsymbol{M}_i)$. The parameter matrices $\boldsymbol{M}_i$ are maximum likelihood estimates obtained from image gradients computed during line detection. Once the $\boldsymbol{M}_i$ are available, the algorithm's E-step and M-step can proceed in alternation.

**3.2.2    The E-Step.** In the E-step of the EM algorithm, a set of posterior probabilities $\alpha_{ij}$ is computed which effectively "weigh" each observation $\boldsymbol{x}_i$ during the estimation of the parameters $\boldsymbol{M}_j^{\boldsymbol{v}}$ for distribution $j$ in the subsequent M-step. The weights $\alpha_{ij}$ are given by

$$\alpha_{ij} = p(j|\boldsymbol{x}_i, \tilde{\mathcal{V}}) = \frac{p(\boldsymbol{x}_i|j, \tilde{\mathcal{V}})p(j|\mathcal{V})}{\sum_{m=1}^{J} p(\boldsymbol{x}_i|m, \tilde{\mathcal{V}})p(m|\mathcal{V})},$$

where $\tilde{\mathcal{V}}$ represents the vanishing point distributions as computed from the previous M-step. Assuming the prior probabilities $p(j|\mathcal{V})$ and current parameter estimates $\boldsymbol{M}_j^{\boldsymbol{v}}$ are known (either from the previous step or from initialization), all that remains is to calculate the mixture component probabilities $p(\boldsymbol{x}_i|j, \mathcal{V})$.

Intuitively, each $p(\boldsymbol{x}_i|j, \mathcal{V})$ represents the likelihood of the line $\boldsymbol{x}_i$ given that it belongs to vanishing point $\boldsymbol{v}_j$. If the line observation were deterministic, this likelihood would be simply $\mathcal{B}_3(\boldsymbol{x}_i; \boldsymbol{M}_j^{\boldsymbol{v}})$. However, $\boldsymbol{x}_i$ is a stochastic measurement itself represented by a probability distribution. Bayesian arguments can therefore be used to determine its likelihood. Let $\boldsymbol{x}_i^0$ represent a particular measurement from the distribution of $\boldsymbol{x}_i$; then

$$p(\boldsymbol{x}_i|j, \mathcal{V}, \boldsymbol{x}_i^0) = \frac{1}{c(\boldsymbol{M}_j^{\boldsymbol{v}})} \exp((\boldsymbol{x}_i^0)^\top \boldsymbol{M}_j^{\boldsymbol{v}}(\boldsymbol{x}_i^0)).$$

To eliminate the dependence on the particular value of $\boldsymbol{x}_i$, the joint likelihood is integrated over all possible measurement values:

$$
\begin{aligned}
p(\boldsymbol{x}_i|j, \mathcal{V}) &= \int p(\boldsymbol{x}_i|j, \mathcal{V}, \boldsymbol{x}_i^0)p(\boldsymbol{x}_i^0)\mathrm{d}\boldsymbol{x}_i^0 \\
&= \int \frac{1}{c(\boldsymbol{M}_j^{\boldsymbol{v}})} \exp\left[(\boldsymbol{x}_i^0)^\top \boldsymbol{M}_j^{\boldsymbol{v}}(\boldsymbol{x}_i^0)\right] \\
&\quad \cdot \frac{1}{c(\boldsymbol{M}_i)} \exp\left[(\boldsymbol{x}_i^0)^\top \boldsymbol{M}_i(\boldsymbol{x}_i^0)\right]\mathrm{d}\boldsymbol{x}_i^0 \\
&= \frac{1}{c(\boldsymbol{M}_j^{\boldsymbol{v}})c(\boldsymbol{M}_i)} \\
&\quad \cdot \int \exp[(\boldsymbol{x}_i^0)^\top(\boldsymbol{M}_j^{\boldsymbol{v}} + \boldsymbol{M}_i)(\boldsymbol{x}_i^0)]\mathrm{d}\boldsymbol{x}_i^0 \\
&= \frac{c(\boldsymbol{M}_j^{\boldsymbol{v}} + \boldsymbol{M}_i)}{c(\boldsymbol{M}_j^{\boldsymbol{v}})c(\boldsymbol{M}_i)}.
\end{aligned}
$$

Thus $p(\boldsymbol{x}_i|j, \mathcal{V})$ can be calculated as a ratio of normalizing coefficients from three different Bingham densities.

**3.2.3    The M-Step.** Once the weights are known, the Bingham parameter matrices $\boldsymbol{M}_j^{\boldsymbol{v}}$ of each vanishing point distribution can be estimated by maximizing the log likelihood function

$$\sum_{i=1}^{k} \sum_{j=1}^{J} \alpha_{ij} \log[p(\boldsymbol{x}_i|j, \mathcal{V})p(j|\mathcal{V})] + \log p(\mathcal{V}) \quad (2)$$

where $p(\mathcal{V})$ is a prior distribution on the vanishing points, and $k$ is the total number of line features.

The exponential form of the Bingham distribution facilitates calculation of the log likelihood. Every parameter matrix $\boldsymbol{M}_j^{\boldsymbol{v}}$ is computed independently by fusing all $k$ observations $\boldsymbol{x}_i$, each weighted by the $\alpha_{ij}$ from the E-step. Using Eq. (2) and pooling the $\boldsymbol{M}_i$ yields

$$\boldsymbol{M}_j^{\boldsymbol{v}} = \sum_{i=1}^k \alpha_{ij} \boldsymbol{M}_i + \boldsymbol{M}_j^{\boldsymbol{0}}$$

where $\boldsymbol{M}_j^{\boldsymbol{0}}$ represents the prior on $\boldsymbol{v}_j$ (Section 3.2.4).

***3.2.4   EM Initialization Using HT.*** Properly formulating and implementing the EM algorithm described above requires the number of vanishing points $J$ to be known. In addition, convergence to the correct solution (i.e. avoidance of local optima) requires reasonably accurate initial parameter estimates. Both quantities can be obtained using a Hough transform [Bar83]. We circumvent the practical difficulties of accuracy and parameterization involved in implementing the HT by using it only to initialize the EM algorithm and to generate a strong prior on the vanishing point estimates.

The HT parameter space is $\mathbb{S}^2$ (i.e. the space of all 3-D line directions), and constraints take the form $\boldsymbol{x}_i \cdot \boldsymbol{v}_j = 0$, where here the $\boldsymbol{x}_i$ are the polar (dual) directions of the input lines. The whole of $\mathbb{S}^2$ is discretized using a cubic parameterization [TPG97], with bin size chosen as a small multiple of the feature noise. Geometrically, each constraint represents a projective line (great circle) with normal $\boldsymbol{x}_i$; intersection of this line with three faces of the unit cube results in a set of at most three straight lines, which are easily discretized using standard clipping and drawing algorithms.

After accumulating the data, the algorithm identifies peaks in the accumulation space, each of which represents a likely vanishing point direction. The number of mixture components $J$ used in the EM algorithm is taken to be the number of statistically significant peak directions, and the initial VP estimates are the vectors from the origin through each peak.

Peak directions also serve as priors $p(\boldsymbol{v}_j)$, each of which is formulated as a bipolar Bingham density ($\kappa_1 \leq \kappa_2 \ll 0$) whose modal axis is aligned with the peak direction. The parameter matrix $\boldsymbol{M}_j^0$ for the prior density can be determined by



*Figure 7.*  Hough Transform for VP Detection. A Hough transform of real line data. The HT algorithm accumulates all lines in a given node, then finds peaks in the antipodally-symmetric accumulation space, implemented as three faces of a unit cube.

forming a scatter matrix (Eq. (1)) from accumulation values in a region around the peak.

## 3.3   Registering Node Pairs

Once vanishing points have been estimated for each node, the relative rotation bringing any pair of nodes into registration can be determined by aligning two or more distinct VPs viewed by both nodes. Section 3.3.1 reviews the classical, deterministic formulation for two-camera registration when VP correspondence is known. Section 3.3.2 extends this classical method: it models uncertainty in the resulting rotations as the fusion of deterministic samples from a Bingham distribution on $\mathbb{S}^3$, and considers how uncertainty in the vanishing points affects the distribution of each node's resulting orientation. Sections 3.3.3 and 3.3.4 address the correspondence problem for the two-node case, and ambiguities that arise in practice.

***3.3.1   Deterministic   Pair   Registration.*** Consider two nodes $\mathcal{A}$ and $\mathcal{B}$, each of which views a common set of $J$ vanishing points. Let $\boldsymbol{v}_j^{\mathcal{A}}$ and $\boldsymbol{v}_j^{\mathcal{B}}$ denote the directions of a particular line direction $\boldsymbol{d}_j$ as seen by each node, and further assume that $\mathcal{B}$ is free to rotate while $\mathcal{A}$ is held fixed. We wish to estimate a single quaternion $\boldsymbol{q}$ that, when applied to $\mathcal{B}$ and its vanishing points, best aligns $\boldsymbol{v}_j^{\mathcal{A}}$ with $\boldsymbol{v}_j^{\mathcal{B}}$. For $\boldsymbol{q}$ to be unique, two distinct VPs are needed (i.e., we require that $J \geq 2$).

In the classical derivation of the optimal $\boldsymbol{q}$

[Hor87], the objective is to determine

$$\text{argmin}_{\boldsymbol{q}} \sum_{j=1}^{J} \|\boldsymbol{v}_j^{\mathcal{A}} - \boldsymbol{R}(\boldsymbol{q})\boldsymbol{v}_j^{\mathcal{B}}\|^2 \qquad (3)$$

$$= \text{argmin}_{\boldsymbol{q}} \left[ \boldsymbol{q}^{\top} \sum_{j=1}^{J} \boldsymbol{A}_j^{\top} \boldsymbol{A}_j \boldsymbol{q} \right]$$

$$= \text{argmin}_{\boldsymbol{q}} \boldsymbol{q}^{\top} \boldsymbol{A} \boldsymbol{q} \qquad (4)$$

i.e. the $\boldsymbol{q}$ that minimizes a quadratic error function. Each $4 \times 4$ matrix $\boldsymbol{A}_j$ is constructed as a linear function of the vanishing points $\boldsymbol{v}_j^{\mathcal{A}}$ and $\boldsymbol{v}_j^{\mathcal{B}}$. The solution to Eq. (4) is the eigenvector corresponding to the minimum eigenvalue of the symmetric $4 \times 4$ matrix $\boldsymbol{A}$.

This method minimizes the error metric of Eq. (3) but, aside from the scalar error residual, produces no notion of uncertainty in the result $\boldsymbol{q}$; nor does it treat uncertainty in the VPs themselves. The next section shows how to incorporate uncertainty into the estimation of $\boldsymbol{q}$.

### 3.3.2  *Stochastic Pair Registration.* 

Recall from Section 2.3 that rotational uncertainty can be described as a Bingham distribution on $\mathbb{S}^3$ characterized by a $4 \times 4$ matrix of parameters $\boldsymbol{M_q}$. The matrix $\boldsymbol{A}$ obtained in Eq. (4), when properly normalized, is analogous to a sample second moment matrix: it is symmetric and positive semidefinite, and its eigenvalues sum to unity. $\boldsymbol{A}$ is a sum of $J$ matrices $\boldsymbol{A}_j^{\top} \boldsymbol{A}_j$ that also possess these properties. Each of these constituent matrices can be viewed as the squared contribution of a "sample" $\boldsymbol{q}_j$ formed from an individual vanishing point correspondence. Thus, a parameter matrix $\boldsymbol{M_q}$ for the distribution on the resulting quaternion $\boldsymbol{q}$ can be obtained directly from $\boldsymbol{A}$ using the ML method mentioned in Section 2.3.

This method can be used only for measurements of equal weight. Extension to weighted data is straightforward, involving a normalized, weighted sum of constituent sample matrices [AT00]. In the general case, however, where vanishing points are described as Bingham variables, the distribution on $\mathbb{S}^3$ induced by each correspondence must be computed.

Every matrix $\boldsymbol{A}_j$ is a function of the vanishing point directions in its underlying correspondence. Thus, the parameters of the Bingham distribution

associated with $\boldsymbol{A}_j$ can also be expressed as a function of these directions. Given particular sample values of vanishing point distributions $\boldsymbol{v}_j^{\mathcal{A}}$ and $\boldsymbol{v}_j^{\mathcal{B}}$, define $\boldsymbol{M}(\boldsymbol{v}_j^{\mathcal{A}}, \boldsymbol{v}_j^{\mathcal{B}})$ as the parameter matrix of the associated distribution. Then the contribution of correspondence $j$ can be obtained by Bayesian integration over all possible sample values of the two constituent vanishing points:

$$p(\boldsymbol{q}_j) = \int_{\mathbb{S}^2} \int_{\mathbb{S}^2} p(\boldsymbol{q}_j | \boldsymbol{v}_j^{\mathcal{A}}, \boldsymbol{v}_j^{\mathcal{B}}) p(\boldsymbol{v}_j^{\mathcal{A}}) p(\boldsymbol{v}_j^{\mathcal{B}}) \mathrm{d}\boldsymbol{v}_j^{\mathcal{A}} \mathrm{d}\boldsymbol{v}_j^{\mathcal{B}}$$

$$= \int_{\mathbb{S}^2} \int_{\mathbb{S}^2} \mathcal{B}_4(\boldsymbol{q}_j; \boldsymbol{M}(\boldsymbol{v}_j^{\mathcal{A}}, \boldsymbol{v}_j^{\mathcal{B}}))$$
$$\cdot \mathcal{B}_3(\boldsymbol{v}_j^{\mathcal{A}}; \boldsymbol{M}^{\mathcal{A}}) \mathcal{B}_3(\boldsymbol{v}_j^{\mathcal{B}}; \boldsymbol{M}^{\mathcal{B}}) \mathrm{d}\boldsymbol{v}_j^{\mathcal{A}} \mathrm{d}\boldsymbol{v}_j^{\mathcal{B}}.$$

This quantity can be approximated by a Bingham distribution on $\mathbb{S}^3$ with parameter matrix $\boldsymbol{M}_j$. (In general, $\boldsymbol{M}_j$ is not exact due to the nonlinear dependence of $\boldsymbol{q}_j \in \mathcal{S}^3$ on $\boldsymbol{v}_j^{\mathcal{A}}$ and $\boldsymbol{v}_j^{\mathcal{B}} \in \mathcal{S}^2$.) Once distributions have been determined for each correspondence $\boldsymbol{q}_j$, the final aggregate distribution is described simply by

$$\boldsymbol{M_q} = \sum_{j=1}^{J} \boldsymbol{M}_j. \qquad (5)$$

### 3.3.3  *Matching VPs Across a Node Pair.* 

The registration methods above assume that one-to-one correspondence has been established between vanishing points detected in a given pair of nodes. In general, determining correspondence is difficult without additional information. However, if the two relevant nodes view sufficient common scene geometry, then approximate initial pose is typically sufficient to establish consistent correspondence. This section presents heuristic methods to determine local (i.e. two-node) correspondence.

If two nodes view overlapping scene geometry, then the sets of VPs detected in each node are likely to contain common members. In this case nodes $\mathcal{A}$ and $\mathcal{B}$ have in common a set of VPs related by a single rigid rotation $\boldsymbol{q}$.

Since at least two correspondences are needed to find a unique rotation relating the two nodes, relative angles between pairs of vanishing points in each node can be used as matching criteria. For example, if the angle between $\boldsymbol{v}_1^{\mathcal{A}}$ and $\boldsymbol{v}_2^{\mathcal{A}}$ differs significantly from that between $\boldsymbol{v}_1^{\mathcal{B}}$ and $\boldsymbol{v}_2^{\mathcal{B}}$, then this *pair couplet* cannot possibly match. Thus,

only those pair couplets are considered whose relative angles are within a small threshold of each other. Angular thresholds are related to the Bingham parameters of the respective vanishing point distributions; highly concentrated distributions thus have tighter thresholds than do distributions with more spread. Since vanishing points are axial, there are two angles to consider (summing to 180°); the minimum of the two is used for comparison (Fig. 8a).

For each pair couplet meeting the relative angle criterion above, the algorithm computes a score $s_i$ as follows. First, the VP pair from node $\mathcal{B}$ is rotated to the VP pair from node $\mathcal{A}$ by $\boldsymbol{q}$ using the deterministic pair registration technique from Section 3.3.1; the direction of each VP is taken as the major axis of its associated Bingham distribution. The positive angle of rotation $\theta_i$ required to align the two pairs is noted, and the remaining vanishing points from $\mathcal{B}$ are then rotated by $\boldsymbol{q}$ and compared with each vanishing point from $\mathcal{A}$. The total number $N_i$ of vanishing points that align to counterparts in $\mathcal{A}$ within a threshold angle, including the original pair, is tabulated.
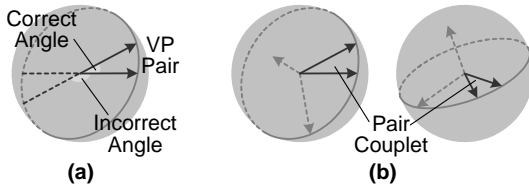


*Figure 8*. Pair Couplets. (a) There are two possible choices when comparing relative angles in axial quantities. By convention, the smaller of the two is chosen. (b) A matching pair couplet is depicted. Relative angles between the pairs are identical despite the fact that the nodes are not rotationally aligned.

Each score is then computed as $s_i = N_i/\theta_i$. This emphasizes correspondence sets containing many matches, while preserving the assumption that the relative rotations are already known to reasonable accuracy. The correspondence set with the highest score is chosen as the "correct" set, for later use in global rotational alignment (Section 3.4.3).

Let $J$ represent the number of VPs viewed by each node. Then enumeration of all possible VP pairs per node is $\mathcal{O}(J^2)$, and enumeration of all possible pair couplets is $\mathcal{O}(J^4)$. Computation of correspondence sets for each couplet is $\mathcal{O}(J^2)$, so the work required overall is $\mathcal{O}(J^6)$. In practice $J$ is small—typically less than 6.

### 3.3.4 *Correspondence Ambiguities.*

Rotation as presented in Section 3.3 requires correspondence between signed directions, but vanishing points are axial (i.e. undirected) quantities. Thus for each pair couplet meeting the relative angle criterion, *two* different rotations, differing by 180°, must be computed along with their scores, one for each combination of VP sign.

Other ambiguities can arise that are not so easily resolved, especially in urban scenes consisting of mutually orthogonal lines. Since relative angles between multiple pairs of vanishing points can be identical (e.g. 90°) within a single node, there may exist several plausible match configurations. If there is significant error in initial rotational pose and if correspondence ambiguities exist, the matching algorithm can fail, finding a plausible but incorrect match assignment. In our system, the sensor's initial orientation estimates are usually accurate enough to avoid this problem.

### 3.4 Registering Node Clusters

The above treatment of rotational registration is deficient in two respects. First, it determines explicit or "hard" correspondence among vanishing points rather than stochastic correspondence; second, it considers only two nodes at a time. This section presents a multi-node extension for rotational registration which addresses the above concerns and produces a globally optimal set of node orientations along with the associated uncertainty of each.
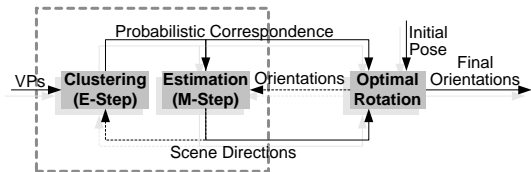


*Figure 9*. Global Orientation Recovery. Vanishing points are aligned with one another to determine node orientations. A two-level feedback hierarchy is used: the high level estimates rotations and scene-relative line directions in alternation; the low level (outlined) classifies VPs and estimates line directions.

As is typical in vision problems, pose recovery consists of two coupled sub-problems: correspondence and registration. In our framework, given a grouping of vanishing points into sets, where each set represents observations of a true scene-relative line direction, estimation of relative rotations be-

comes simpler. Conversely, given a set of accurate node orientations, determining correspondence is simplified. This suggests an iterative bundle-adjustment scheme that alternately estimates orientations given correspondence, then establishes correspondence given orientations (Fig. 9).

Rotations and correspondence are initially produced by exhaustive search (Section 3.4.3). Global (scene-relative) directions are estimated based on vanishing point clusters; each node is then rotated until its vanishing points optimally align with these global directions, and the process repeats. There are two levels of feedback in the process, one at the high level of rotational bundle adjustment and the other in the estimation of global line directions, which alternates between determination of probabilistic correspondence and estimation of directional distributions.

**3.4.1  EM for Multi-Node Registration.** This alternation between classification and estimation suggests application of an EM algorithm, which would circumvent the need for explicit correspondence and provide an adequate probabilistic estimation framework. At its core, the problem is to determine the probability distributions of a set of rotations in the form of quaternions, $\mathcal{Q} = \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_M\}$, based solely on the data $\mathcal{V} = \{\mathcal{V}^1, \ldots, \mathcal{V}^M\}$, where $\mathcal{V}^i$ is the set of vanishing points $\boldsymbol{v}_j^i$ detected in node $i$. Probabilistically, this can be written as

$$\operatorname{argmax}_{\mathcal{Q}}[p(\mathcal{Q}|\mathcal{V})].$$

However, the rotations depend on scene-relative line directions $\mathcal{D}$, as well as correspondence $\mathcal{C}$ between these directions and the vanishing points in each node. Using Bayes' rule, the likelihood to be maximized can thus be rewritten as

$$p(\mathcal{Q}|\mathcal{V}) = \int_{\mathcal{D}} p(\mathcal{Q}|\mathcal{D}, \mathcal{V}) p(\mathcal{D}|\mathcal{V}) \mathrm{d}\mathcal{D}$$
$$= \int_{\mathcal{D}} \sum_{\mathcal{C}} p(\mathcal{Q}|\mathcal{C}, \mathcal{D}, \mathcal{V}) p(\mathcal{C}|\mathcal{D}, \mathcal{V}) p(\mathcal{D}|\mathcal{V}) \mathrm{d}\mathcal{D}.$$

Note that a sum is taken over $\mathcal{C}$ rather than an integral, because the set of correspondence configurations is discrete. The quantity $p(\mathcal{D}|\mathcal{V})$ represents the prior distribution on global line directions given only the vanishing point data, and is taken to be uniform, since in the absence of rotational pose, nothing is known about this distribu-

tion. The quantity $p(\mathcal{C}|\mathcal{D}, \mathcal{V})$ is the prior distribution on correspondence given only global line directions and vanishing points (not rotations). This distribution can be approximated from the pairwise correspondences established in Section 3.3.3.

The high-level EM algorithm alternates between two steps. First, it computes the likelihoods $p(\mathcal{C}, \mathcal{D}|\mathcal{Q}, \mathcal{V})$. Next, it maximizes the expression

$$\int_{\mathcal{D}} \sum_{\mathcal{C}} p(\mathcal{C}, \mathcal{D}|\mathcal{Q}, \mathcal{V}) \log p(\mathcal{Q}|\mathcal{C}, \mathcal{D}, \mathcal{V}) \mathrm{d}\mathcal{D}.$$

The likelihoods computed in the E-step serve as weights on the conditional log-likelihood maximized in the M-step. Conditioned on line directions and correspondence, the quaternions are independent of one another because vanishing points in each node can be rotated in isolation to optimally align with the global line directions. Thus,

$$\log p(\mathcal{Q}|\mathcal{C}, \mathcal{D}, \mathcal{V}) = \log \prod_{i=1}^{M} p(\boldsymbol{q}_i|\mathcal{C}, \mathcal{D}, \mathcal{V})$$
$$= \sum_{i=1}^{M} \log p(\boldsymbol{q}_i|\mathcal{C}, \mathcal{D}, \mathcal{V})$$

and each quaternion can be estimated independently. Maximization proceeds as described in Section 3.3.2, with the Bingham distribution of orientation $\boldsymbol{q}_i$ specified by $\boldsymbol{M}_i^{\boldsymbol{q}}$, which represents a weighted sum of correspondence matrices of the form in Eq. (5).

**3.4.2  EM for Multi-Node Correspondence.** The algorithm above solves the M-step of the bundle adjustment, but the E-step still remains—the likelihoods $p(\mathcal{C}, \mathcal{D}|\mathcal{Q}, \mathcal{V})$ must be computed. Intuitively, these likelihoods represent distributions on correspondence $\mathcal{C}$ and scene-relative line directions $\mathcal{D}$ given $\mathcal{Q}$, the current set of orientation estimates. However, $\mathcal{C}$ and $\mathcal{D}$ are coupled; knowledge of the line directions influences the groupings, and vice versa.

Let $\tilde{\boldsymbol{v}}_j^i$ represent vanishing point $j$ in node $i$ after rotation by $\boldsymbol{q}_i$; the set of all such directions serves as the pool of data to be grouped. Further, let $\boldsymbol{d}_k$ represent a scene-relative 3-D line direction. The problem then becomes to simultaneously estimate the $\boldsymbol{d}_k$ and classify the $\tilde{\boldsymbol{v}}_j^i$.

This formulation is identical to the vanishing point estimation problem of Section 3.2. The collective dataset $\tilde{\mathcal{V}}$ is drawn from a weighted mixture

of Bingham distributions of $d_k$; the only difference is that the underlying samples are now bipolar rather than equatorial. Applying the lower-level EM algorithm estimates the line direction distributions $d_k$ and produces a probabilistic assignment of individual vanishing points to each $d_k$. After convergence, the resulting assignment weights are fed back into the M-step of Section 3.4.1.

***3.4.3 Initialization.*** As noted above, EM algorithms are effective only when properly initialized. This requires that the number of mixtures be known (in this case, the number $J$ of 3-D line directions), and that reasonable initial values be supplied (in this case, rotations and correspondences). This section outlines the initialization of the EM technique.

The initialization stage takes the node adjacency graph as input and proceeds as follows. First, it applies the two-node correspondence technique of Section 3.3.3 to each adjacent node pair, extracting unique vanishing point matches. It then combines multiple matching VPs into single, global line directions. The algorithm proceeds as follows:

    Clear the list of global line directions (GLDs)
    For each node pair in adjacency graph
      Apply two-node VP correspondence
        For each VP pair matched
          If neither VP exists in any GLD
            Create new GLD and add to list
            Link both constituent VPs to new GLD
          Else if one VP exists
            Find its GLD
            Link other VP to this direction
          Else if both VPs exist
            If associated with different GLDs
              Merge the two GLDs

This algorithm produces a list of vanishing point clusters, each of which represents observations of a single scene-relative 3-D line direction. The mixture model components of Section 3.4.2 are initialized to these VP clusters, and the correspondence weights (i.e., the probabilities associating VPs with global line directions) are initialized to binary values according to the grouping above. Any node with fewer than two VPs is tagged as unalignable. The algorithm can produce separate VP clusters representing the same 3-D direction. The EM algorithm (Section 3.4.1) combats this by merging all clusters that overlap with at least 95% probability.

# 4  Position Recovery

Recovery of structure and motion from image information encompasses several coupled problems: camera registration, feature correspondence, and scene structure. Rotational registration of the cameras simplifies the epipolar geometry and reduces the dimension of the search space, but the coupling between correspondence and node positions remains. Our approach is to estimate both correspondence and position simultaneously as probability densities, deferring commitment to deterministic values until global information is assembled and propagated throughout the node network.

## 4.1  Overview

The position recovery algorithm proceeds as follows (Fig. 10). First, translation directions (*baselines*) are estimated for every adjacent node pair in the network (Sections 4.2, 4.3), using a Hough transform on all possible feature matches. This approximate direction initializes an EM method whose E-step samples from a high-dimensional distribution using a Markov chain Monte-Carlo algorithm. This MCEM algorithm averages over all possible correspondence sets to determine the best motion direction (Section 4.4).

The position recovery algorithm next assembles all pairwise baselines into a global optimization that estimates the camera positions most consistent with the baselines (Section 4.5). A final step rigidly transforms the resulting network to be maximally consistent with the sensor's (Earth-relative) position estimates (Section 4.6).
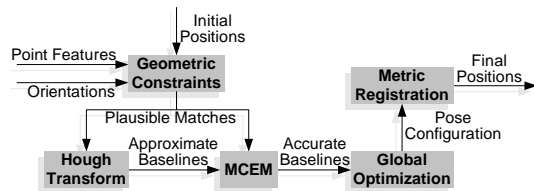


*Figure 10.* Translational Registration.

## 4.2  Two-Node Baseline Geometry

Given two rotationally registered nodes $\mathcal{A}$ and $\mathcal{B}$, and their respective feature sets $\mathcal{X}$ =

$\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$ and $\mathcal{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$, we wish to determine the motion direction (or baseline) $\boldsymbol{b}$ from $\mathcal{A}$ to $\mathcal{B}$ most consistent with the available data. This section describes the simplified epipolar geometry that arises when the rotation relating $\mathcal{A}$ and $\mathcal{B}$ is known, and presents geometric constraints that may be used to reject unlikely point matches. Given a set of explicit matches, baseline estimation reduces to a projective inference problem similar to that of vanishing point estimation.

**4.2.1   Epipolar Geometry.** An epipolar plane $\mathcal{P}$ contains two node centers and a 3-D point observed in both nodes (Fig. 11). Projections of the 3-D point onto each of the images, $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ respectively, must therefore also lie in $\mathcal{P}$.   For ro-
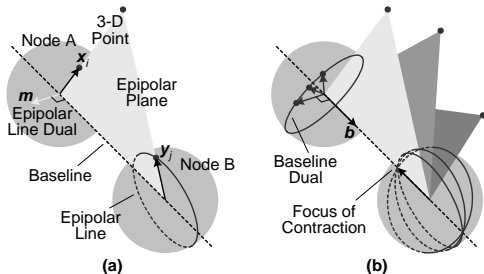


*Figure 11.* Pair Translation Geometry for Rotationally Aligned Nodes. (a) A single 3-D point lies in an epipolar plane containing the baseline and any projective observations of the point. The epipolar line is analogous to an image line feature. (b) The epipolar planes induced by a set of 3-D points forms a pencil coincident with the baseline. The normals of these planes thus lie on a great circle orthogonal to the baseline direction.

tationally registered nodes, the following relation holds:

$$(\boldsymbol{x}_i \times \boldsymbol{y}_j) \cdot \boldsymbol{b} = 0. \qquad (6)$$

Intuitively, the cross product of $\boldsymbol{x}_i$ with $\boldsymbol{y}_j$ is orthogonal to $\mathcal{P}$, and thus orthogonal to the baseline $\boldsymbol{b}$ (since $\mathcal{P}$ contains $\boldsymbol{b}$). Here, observations consist only of the 2-D feature projections, and the baseline is unknown; however, Eq. (6) provides a constraint on $\boldsymbol{b}$. Thus $\boldsymbol{b}$ can be inferred, up to unknown scale, solely from two or more corresponding feature pairs.

Define $\boldsymbol{m}_{ij} \equiv \boldsymbol{x}_i \times \boldsymbol{y}_j$. For the correct pairs of $i$ and $j$—that is, for those $(i, j)$ couplets in which $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ arise from a single 3-D scene point—the constraint in Eq. (6) becomes

$$\boldsymbol{m}_{ij} \cdot \boldsymbol{b} = 0.$$

If the $\boldsymbol{m}_{ij}$ are interpreted as projective epipolar lines, then the baseline $\boldsymbol{b}$ is the projective *focus of expansion*, and its antipode the *focus of contraction*, the apparent intersections of all epipolar lines (Fig. 11).

**4.2.2   Geometric Match Constraints.** Both correspondence and the baseline are initially unknown, so the above construction may at first appear hopelessly underconstrained. There are $MN$ possible individual feature matches and

$$\sum_{F=0}^{F'} \binom{M}{F} \binom{N}{F} F!$$

possible correspondence *sets*, making the search space enormous (super-exponential in the number of features; see Appendix). However, matching constraints can drastically lower the search space dimension, both by reducing the number of features in each node, and by eliminating most candidate correspondences. The constraints presented here rely on two assumptions: first, that each point feature arises from the intersection of two or more 2-D line features; and second, that the true baseline lies within a bounded region inferred from rough initial pose.

The 3-D line directions and 2-D line feature classification obtained from rotational pose recovery both provide strong cues for feature culling and point correspondence rejection. Presumably, objects exhibiting parallel lines possess sufficient structure for determination of translational offsets; thus, the algorithm discards point features not associated with any parallel line set (i.e., those whose constituent lines have high outlier probability). Also, it discards point features inferred from lines subtending less than a threshold angle (i.e., those that could not be reliably localized).

A set of all possible candidate matches is constructed from the remaining sets of point features. Each match is evaluated according to the following criteria:

- **Constituent line directions.** The 3-D line directions forming $\boldsymbol{x}_i$ must be identical to those forming $\boldsymbol{y}_j$ (Fig. 12); otherwise the putative match $\boldsymbol{m}_{ij}$ is discarded.

- **Baseline uncertainty bound.**   The angular bound on the baseline direction induces a conservative equatorial band within
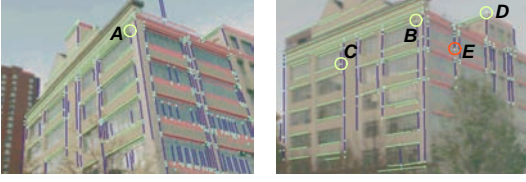
*Figure 12.* Line Constraints. Given two images of the same building, a point feature $A$ in the first image has several plausible matches. Point $B$ is the true match, but $C$ and $D$ are also plausible because they are formed by the intersection of lines whose directions match those of the lines forming $A$ (note that $D$ is formed by the intersection of three rather than two distinct line directions). The directions of the lines forming $E$ do not match those forming $A$, so $E$ is rejected.

which all true epipolar plane normals must lie (Fig. 13); matches outside this band are discarded. Analogously, matches for which $\boldsymbol{y}_j$ is closer than $\boldsymbol{x}_i$ to $\boldsymbol{b}$ are discarded, as they imply backward motion.

- **Depth of 3-D point.** An excessive angle between $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ implies either a 3-D point too close to the node, or an abnormally long baseline. Such matches are therefore discarded.
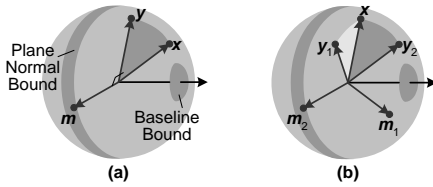


*Figure 13.* Direction Constraints. (a) Uncertainty in the baseline direction induces an equatorial band of uncertainty for epipolar lines. The match between features $\boldsymbol{x}$ and $\boldsymbol{y}$ is plausible because it implies motion in the correct direction. (b) The match between $\boldsymbol{x}$ and $\boldsymbol{y}_1$ is rejected because its epipolar line does not lie in the uncertainty band; the match with $\boldsymbol{y}_2$ is rejected because it implies backward motion.

### 4.3   Single Baseline Inference

This section describes methods for inferring the translation direction between a pair of nodes, first assuming explicit correspondence is known, then relaxing this assumption. As noted above, a given correspondence between features $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ constrains the inter-node baseline $\boldsymbol{b}$ according to Eq. (6), and a set of such correspondences can be used to estimate $\boldsymbol{b}$.

Projective fusion techniques can be used to estimate the probability density of $\boldsymbol{b}$. Recall from Section 1.2 that every point feature represents the intersection of two image lines, each of which is an uncertain equatorially-distributed Bingham variable with known parameters. Bingham uncertainty in the intersection can be determined by fusing the two lines, so that each point feature's Bingham distribution is known. Each correspondence between random variables $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ in turn induces an epipolar line $\boldsymbol{m}_{ij}$, whose equatorial Bingham distribution can be determined by fusion of $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$. The problem that remains is to determine the distribution of $\boldsymbol{b}$ given a set of uncertain epipolar line observations $\boldsymbol{m}_{ij}$.

**4.3.1   Known Correspondence.** If true correspondences between the feature sets $\mathcal{X}$ and $\mathcal{Y}$ are known, the baseline distribution parameters $\boldsymbol{M_b}$ can be computed according to the fusion equation

$$\boldsymbol{M_b} = \boldsymbol{M}_0 + \sum_{(i,j)\in\mathcal{F}} \boldsymbol{M}_{ij}$$

where $\boldsymbol{M}_{ij}$ represents the uncertainty of the epipolar line $\boldsymbol{m}_{ij}$, $\boldsymbol{M}_0$ is the prior distribution on $\boldsymbol{b}$, $\mathcal{F}$ is the set of $F$ pairings $(i,j)$ representing true matches, and the sum is taken only over indices $(i,j) \in \mathcal{F}$.

Equivalently, inference can be performed by associating a binary variable $b_{ij}$ with *every possible* correspondence, where

$$b_{ij} = \begin{cases} 1, & \text{if } \boldsymbol{x}_i \text{ matches } \boldsymbol{y}_j \\ 0, & \text{otherwise.} \end{cases}$$

The Bingham parameters of $\boldsymbol{b}$ are then

$$\boldsymbol{M_b} = \boldsymbol{M}_0 + \sum_{i=1}^{M}\sum_{j=1}^{N} b_{ij}\boldsymbol{M}_{ij}. \qquad (7)$$

**4.3.2   Probabilistic Correspondence.** Because motion directions and point features are uncertain quantities, and because ambiguous epipolar geometry may arise from certain motions, *hard* or *explicit* correspondence cannot always be determined. We account for this by using continuous variables $w_{ij} \in [0,1]$, rather than binary variables $b_{ij} \in \{0,1\}$, can be applied to the observations $\boldsymbol{m}_{ij}$, effectively representing the probability that feature $\boldsymbol{x}_i$ matches feature $\boldsymbol{y}_j$.

Inference of $\boldsymbol{b}$ in this formulation becomes

$$\boldsymbol{M_b} = \sum_{i=1}^{M}\sum_{j=1}^{N} w_{ij}\boldsymbol{M}_{ij} + \boldsymbol{M}_0,$$

with more emphasis given to matches with higher likelihood. (The binary variables $b_{ij}$ are the deterministic limit of the $w_{ij}$.)

**4.3.3  Feature Match Weights.** In reality, each feature observed in one node has at most one true match in the other node. A true match exists only if the feature observation corresponds to a real 3-D point, and if its counterpart in the other node is visible; otherwise, the feature has *no* match—either it is itself spurious, or its match is unobserved (e.g. occluded or otherwise missed by detection).

In the case of binary variables, the above condition can be enforced by requiring that at most one $b_{ij}$ for every $i$, and at most one $b_{ij}$ for every $j$, is equal to one, and that the rest are equal to zero. More formally,

$$\sum_{j=1}^{N} b_{ij} \le 1 \quad \forall i \qquad \sum_{i=1}^{M} b_{ij} \le 1 \quad \forall j. \qquad (8)$$

Inequality constraints are mathematically inconvenient; thus, the "null" features $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$ are appended to $\mathcal{X}$ and $\mathcal{Y}$, respectively, and the inequality constraints of Eq. (8) become equality constraints via the introduction of binary-valued slack variables $b_{i0}$ and $b_{0j}$ [CR00b], which take value one if $\boldsymbol{x}_i$ (or $\boldsymbol{y}_j$, respectively) matches no other feature, and zero otherwise. Thus,

$$\sum_{k=0}^{N} b_{ik} = \sum_{k=0}^{M} b_{kj} = 1 \qquad \begin{matrix} i \in [1, M] \\ j \in [1, N] \end{matrix}. \qquad (9)$$

To ensure valid weights $w_{ij}$ in the probabilistic case, an analogous condition must be satisfied:

$$\sum_{k=0}^{N} w_{ik} = \sum_{k=0}^{M} w_{kj} = 1 \qquad \begin{matrix} i \in [1, M] \\ j \in [1, N] \end{matrix}. \qquad (10)$$

This condition enforces a symmetric (two-way) distribution over all correspondences: each feature in the first node can match a set of possible features in the second node, with the weights normalized so that they sum to one, and vice versa.

The set of weights can also be represented by an $(M+1) \times (N+1)$ matrix $\boldsymbol{W}$ (or $\boldsymbol{B}$, in the binary case), whose rows represent the features $\mathcal{X}$, whose columns represent the features $\mathcal{Y}$, and whose individual entries are the weights themselves (Fig. 14). The condition in Eq. (10) is then equivalent to the requirement that the weight matrix be doubly stochastic, i.e. that both its rows and its columns sum to one.
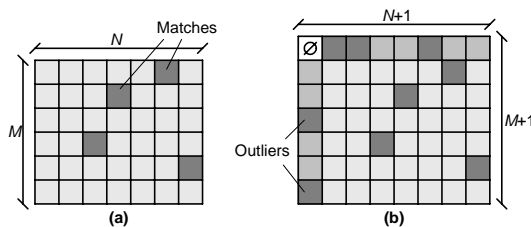


*Figure 14.*  Augmented Match Matrix. The match matrix encodes correspondences between features in two different nodes. (a) An example of a binary match matrix. Rows represent features in the first node, and columns represent features in the second. There can be at most one non-zero entry per row and per column. (b) An augmented matrix, with an extra row and column to account for outliers and missing features (i.e., matches to a null feature "∅"). This matrix has exactly one non-zero entry in each row but the top-most, and each column but the left-most.

**4.3.4  Obtaining a Prior Distribution.** Because motion direction and correspondence are tightly coupled, it is difficult to determine them without prior information. This section shows how using initial pose estimates and geometric constraints from Section 4.2.2 allows an accurate initial estimate of $\boldsymbol{b}$ to be obtained without correspondence.

Let $\mathcal{M}$ represent the set of *all* plausible correspondences (epipolar lines) between $\mathcal{X}$ and $\mathcal{Y}$, and let the subset $\mathcal{M}' \in \mathcal{M}$ contain only the $F$ true matches. If all lines in $\mathcal{M}$ are drawn on $\mathbb{S}^2$, those in $\mathcal{M}'$ (in the absence of noise) will intersect at the motion direction $\boldsymbol{b}$, and the remainder, which represent false matches, will intersect at random points on the sphere.

The point of maximum incidence on $\mathbb{S}^2$ is then the most likely baseline direction. This point can be found by discretizing $\mathbb{S}^2$ and accumulating all candidate epipolar lines $\mathcal{M}$ in a Hough transform (Fig. 15). Since the approximate motion direction is known, the transform need only be evaluated over a small portion of the sphere's surface around
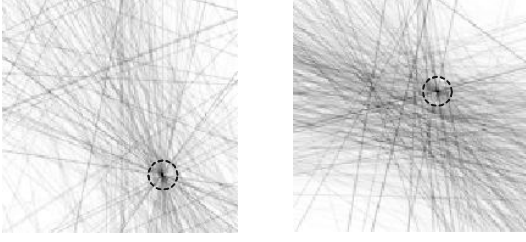
*Figure 15.* Hough Transform for Baseline Estimation. Two examples of Hough transforms for baseline estimation. Epipolar lines for all plausible matches are accumulated; the transform peak represents the baseline direction.

this approximate direction. As in Section 3.2.4, we choose the bin size as a small multiple of the feature noise.

The motion direction $\boldsymbol{b}_0$ can be determined as the peak in the transform with highest magnitude. False correspondences outnumber true correspondences, however, because there are $MN$ possible matches and only $F$ (at most $\min(M,N)$) true matches. The desired peak may therefore be obscured by spurious peaks. For example, a point feature in one node lying very close to the motion direction can match many features in the other node, producing a sharp, false peak when all matches are equally weighted (Fig. 16).
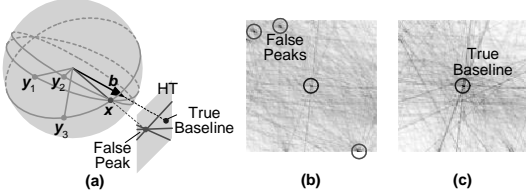


*Figure 16.* False Hough Transform Peaks. (a) False peaks in the Hough transform can be caused by features too close to the motion direction, which have many matches and thus produce high-incidence regions. (b) An example in which false peaks are evident. (c) The same example after normalization.

To solve this problem, a set of weights $w_{ij}$ must be assigned to the epipolar lines in $\mathcal{M}$ that deemphasizes features with many possible matches. These weights must also satisfy Eq. (10). We use an iterative normalization procedure [Sin64] to transform the initial (invalid) match matrix into a (valid) doubly stochastic matrix. First, the matrix $\boldsymbol{W}$ is set to zero; entries for matches satisfying the geometric constraints of Section 4.2.2, as well as all entries in row $M+1$ and column $N+1$, are then assigned an initial value of one. The algo-

rithm alternatively normalizes rows and columns until convergence as follows:

$$w'_{ij} \;=\; w_{ij} \Big/ \sum_{j=0}^{N} w_{ij} \;\; \forall i \in \{1, \dots, M\};$$

$$w''_{ij} \;=\; w'_{ij} \Big/ \sum_{i=1}^{M} w'_{ij} \;\; \forall j \in \{1, \dots, N\}.$$

The algorithm produces a provably unique, doubly-stochastic matrix $\boldsymbol{W}'$. The new matrix does not represent the "correct" distribution, because it is somewhat arbitrarily initialized, but it provides a useful approximation for the purposes of the Hough transform described above.

For a planar accumulation space, each linear constraint of the form in Eq. (6) contributes a single straight line to the transform. After weights have been obtained, the epipolar lines are accumulated, weighted by the appropriate value $w_{ij}$. This normalization to a valid probability distribution over correspondences dramatically improves the coherence of the true motion direction (Fig. 16c).

Although the Hough transform's discrete nature inherently limits its accuracy, the resulting motion direction estimate $\boldsymbol{b}_0$ is used only to initialize a more accurate technique (described in the next section). Further, it can be used as a strong prior distribution (with parameters $\boldsymbol{M}_0$ in the notation of Section 4.3) in subsequent inference. The matrix $\boldsymbol{M}_0$ is computed from a bipolar scatter matrix approximation in the region surrounding the peak.

### 4.4 Monte Carlo EM

In general, true feature correspondence is unknown; feature point measurements and uncertainty are the only information available for baseline inference. This section outlines a method for determining accurate motion estimates without explicit correspondence. Using maximum likelihood notation,

$$\boldsymbol{b}^* = \operatorname{argmax}_{\boldsymbol{b}}[p(\boldsymbol{b}|\mathcal{M})]. \tag{11}$$

The conditional probability $p(\boldsymbol{b}|\mathcal{M})$ can be expanded using Bayes' rule:

$$p(\boldsymbol{b}|\mathcal{M}) = \sum_{\boldsymbol{B}} p(\boldsymbol{b}|\boldsymbol{B},\mathcal{M})p(\boldsymbol{B}|\mathcal{M}) \tag{12}$$

where $\boldsymbol{B}$ is a valid binary-valued correspondence matrix, and $p(\boldsymbol{B}|\mathcal{M})$ is the prior distribution on

the correspondence set. This prior distribution incorporates the geometric match constraints of Section 4.2.2. As in Section 3.4.1, the likelihood is expressed as a summation rather than an integration, because the collection of all possible correspondence sets is discrete.

### 4.4.1 Baseline Estimation Without Correspondence.
The expression in Eq. (11) suggests that the optimal estimate of the motion direction $\boldsymbol{b}$ can be found *without using explicit correspondence*, by maximizing $p(\boldsymbol{b}|\mathcal{M})$ alone [DSTT00]. Correspondence sets can be treated as nuisance parameters in a Bayesian formulation, as in Eq. (12), in which the likelihood is evaluated over all possible matrices $\boldsymbol{B}$. The EM algorithm is well-suited to this classification/optimization problem. Convergence to the optimal solution is virtually guaranteed because of the initial estimate provided by the Hough transform.

The log likelihood to be maximized is

$$L = \sum_{\boldsymbol{B}} p(\boldsymbol{B}|\boldsymbol{b},\mathcal{M}) \log p(\boldsymbol{b}|\boldsymbol{B},\mathcal{M}). \qquad (13)$$

Substituting Eq. (7) into Eq. (13) gives

$$
\begin{aligned}
L \;\propto\; & \boldsymbol{b}^\top \boldsymbol{M}_0 \boldsymbol{b} \\
& + \sum_{\boldsymbol{B}} p(\boldsymbol{B}|\boldsymbol{b},\mathcal{M}) \sum_{i=1}^{M} \sum_{j=1}^{N} b_{ij} \boldsymbol{b}^\top \boldsymbol{M}_{ij} \boldsymbol{b}. \;(14)
\end{aligned}
$$

Define $w_{ij}$ as the marginal posterior probability of match $b_{ij}$, regardless of the other matches; that is,

$$w_{ij} \equiv p(b_{ij} = 1|\boldsymbol{b},\mathcal{M}) = \sum_{\boldsymbol{B}} \delta(i,j) p(\boldsymbol{B}|\boldsymbol{b},\mathcal{M}).$$

Then Eq. (14) becomes

$$L \propto \boldsymbol{b}^\top \boldsymbol{M}_0 \boldsymbol{b} + \sum_{i=1}^{M} \sum_{j=1}^{N} w_{ij} \boldsymbol{b}^\top \boldsymbol{M}_{ij} \boldsymbol{b}. \qquad (15)$$

Given the weights $w_{ij}$, the technique described in Section 4.3.2 maximizes $L$. However, determining the $w_{ij}$ is not so straightforward. Individual matches are not mutually independent, because information about one match provides information about others. For example, given that $b_{ij} = 1$, it must be true that

$$b_{ik} = 0 \qquad \forall k \neq j.$$

Independence therefore does not hold, because

$$p(b_{ik} = 1|b_{ij} = 1, \boldsymbol{b}, \mathcal{M}) = 0 \neq p(b_{ik} = 1|\boldsymbol{b}, \mathcal{M}),$$

and the joint likelihood $p(\boldsymbol{B}|\boldsymbol{b},\mathcal{M})$ cannot be factored. Precise evaluation of Eq. (15) apparently requires evaluation of Eq. (14), a difficult task due to the large number of correspondence sets. However, the following sections show how to evaluate the $w_{ij}$ efficiently by Monte Carlo sampling.

### 4.4.2 Sampling the Posterior Distribution.
Markov chain Monte Carlo (MCMC) algorithms are useful for evaluating sums of the form in Eq. (14). In this context, each valid binary match matrix $\boldsymbol{B}^k$ represents a distinct *state*; random transitions between states occur until *steady state* is reached. If the transition likelihoods are appropriately chosen, then the steady-state probabilities represent the distribution on $\boldsymbol{B}$.

Our approach combines Metropolis sampling [MRR+53], which ensures appropriate transition probabilities, with simulated annealing [KGV83], which avoids relative likelihood maxima by visiting a larger portion of the sample space. The approach can be summarized as follows:

> Start with initial temperature $T = T_0 > 1$
> Loop until $T \leq 1$ (E-step):
>> Set $k = 0$
>> Start with valid state $\boldsymbol{B}^0$
>> Compute initial parameter matrix $\boldsymbol{M}^0$
>> Compute initial likelihood coefficient $c(\boldsymbol{M}^0)$
>> Set $\boldsymbol{A} = \boldsymbol{0}$
>> Loop until $k$ sufficiently high (steady state):
>>> Randomly perturb state to $\tilde{\boldsymbol{B}}^k$
>>> Evaluate the likelihood ratio $\beta$
>>> If $\beta \geq 1$ Then keep new state
>>> Else keep new state with probability $\beta^{1/T}$
>>> If new state kept Then
>>>> Set $\boldsymbol{B}^{k+1} = \tilde{\boldsymbol{B}}^k$
>>>> Compute $\boldsymbol{M}^{k+1}$ and $c(\boldsymbol{M}^{k+1})$
>>> Else set $\boldsymbol{B}^{k+1} = \boldsymbol{B}^k$
>>> Set $\boldsymbol{A} = \boldsymbol{A} + \boldsymbol{B}^{k+1}$
>>> Set $k = k + 1$
>> Set $\boldsymbol{W} = \boldsymbol{A}/k$
>> Estimate new $\boldsymbol{b}$ given $\boldsymbol{W}$ (M-step)
>> Set $T = \alpha T$ (for $0 < \alpha < 1$)

The likelihood function used to compute $\beta$ (i.e., the ratio of the new likelihood to the old) is

$$p(\boldsymbol{B}^k|\boldsymbol{b},\mathcal{M}) = c(\boldsymbol{M}^k) \exp\left[\boldsymbol{b}^\top \boldsymbol{M}^k \boldsymbol{b}\right] \qquad (16)$$

where $\boldsymbol{b}$ is taken as the modal direction of the current baseline distribution estimate. Expansion of Eq. (16) gives

$$p(\boldsymbol{B}^k|\boldsymbol{b}, \mathcal{M}) = c(\boldsymbol{M}^k) \exp\left[\boldsymbol{b}^\top \sum_{i=1}^{M} \sum_{j=1}^{N} b_{ij}^k \boldsymbol{M}_{ij} \boldsymbol{b}\right].$$

Efficient calculation of the ratio $\beta$ is described in Section 4.4.4.

In a particular E-step loop, $\boldsymbol{A}$ is an $(M + 1) \times (N + 1)$ matrix that accumulates the visits to each state. $\boldsymbol{W}$ is a valid matrix of marginal probabilities (weights) $w_{ij}$ obtained by averaging all state visits. The initial temperature $T_0$ is set to a relatively low value; high initial temperatures would explore larger regions of the parameter space, which is unnecessary because the Hough transform provides a reasonably accurate initial estimate $\boldsymbol{b}_0$. The value of $T_0$ is chosen proportionally to the uncertainty of $\boldsymbol{b}_0$, and is typically between 1.5 and 2.0 in practice.

The MCMC algorithm requires a valid starting state, and random state perturbations that satisfy detailed balance (meaning that every valid state is reachable from every other valid state). Thus perturbations must be defined which can visit the entire state space. These perturbations are described next.

### 4.4.3   Match Perturbations.
When $\boldsymbol{B}^k$ is a square permutation matrix (i.e. all features are visible in all images), simple swap perturbations suffice to reach all states [DSTT00], so that $\boldsymbol{B}^{k+1}$ is identical to $\boldsymbol{B}^k$ except for a single row or column swap (Fig. 17). However, when the number of 3-D features is unknown, or when outliers and occlusion are present, detailed balance is no longer satisfied by simple match swapping, since states with more or fewer matches than the current state are never reached.

We generalize this technique, in the two-camera case, to handle an unknown number of visible 3-D features, and also to handle outliers and occlusion. First, we augment the state matrix $\boldsymbol{B}$ and probability matrix $\boldsymbol{W}$ with an extra row and column (Section 4.3.3) to represent an appropriate state space (i.e. to account for unmatched features). Second, we introduce novel, complementary *split* and *merge* perturbations that allow all states to be
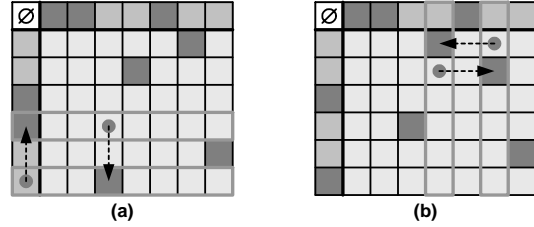


*Figure 17.* Row and Column Swaps. (a) Two rows of the match matrix, including outliers, are interchanged. (b) Two columns are interchanged.

visited (Fig. 18). The split perturbation converts a valid match into two outliers. The merge perturbation joins two outliers into one valid match.
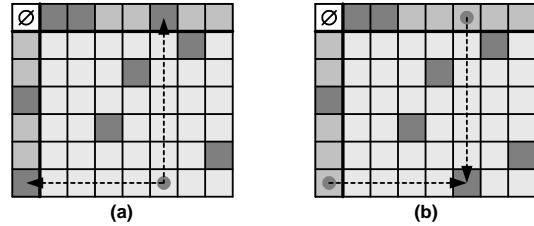


*Figure 18.* Split and Merge Perturbations. (a) A valid correspondence is split into two outliers, reducing the number of valid matches by one. (b) Two outliers are merged into a valid correspondence, increasing the number of valid matches by one.

### 4.4.4   Efficient Sampling.
The sampling algorithm outlined above may seem computationally expensive, especially for the large state matrices typical of real images with many features. However, three optimizations can be applied to significantly improve the algorithm's performance. Most entries of any given state matrix are zero; in fact, out of $(M + 1)(N + 1)$ possible entries, a maximum of $M + N$ are non-zero (this corresponds to the case where all features are outliers). Thus the first optimization is to use sparse matrix representations for $\boldsymbol{B}$ and for state perturbations. Because of the geometric match constraints from Section 4.2.2, many configurations $\boldsymbol{B}$ are invalid. Thus, the second optimization is to consider only those state perturbations involving valid matches.

The final optimization involves computation of the likelihood ratios $\beta$. Each perturbation represents only an incremental change in the state involving at most four entries in $\boldsymbol{B}$. The exponential form of the likelihood function in Eq. (16)

facilitates computation of ratios:

$$\beta = \frac{p(\tilde{\boldsymbol{B}}^k | \boldsymbol{b}, \mathcal{M})}{p(\boldsymbol{B}^k | \boldsymbol{b}, \mathcal{M})} = \frac{c(\tilde{\boldsymbol{M}}^k) \exp\left[\boldsymbol{b}^\top \tilde{\boldsymbol{M}}^k \boldsymbol{b}\right]}{c(\boldsymbol{M}^k) \exp\left[\boldsymbol{b}^\top \boldsymbol{M}^k \boldsymbol{b}\right]}$$

$$= \frac{c(\tilde{\boldsymbol{M}}^k)}{c(\boldsymbol{M}^k)} \exp\left[\boldsymbol{b}^\top (\tilde{\boldsymbol{M}}^k - \boldsymbol{M}^k) \boldsymbol{b}\right]. \qquad (17)$$

When swapping two rows, say row $m$ which contains a one in column $n$ with row $p$ which contains a one in column $q$, most terms in the sum of Eq. (16) remain unchanged; only $b_{mn}$, $b_{mq}$, $b_{pn}$, and $b_{pq}$ differ. The new matrix is

$$\tilde{\boldsymbol{M}}^k = \boldsymbol{M}^k - \boldsymbol{M}_{mn} + \boldsymbol{M}_{mq} + \boldsymbol{M}_{pn} - \boldsymbol{M}_{pq},$$

which involves only four new terms that can be computed from the current parameter matrix.

Split and merge perturbations have equally simple incremental computations, since they also involve only a few entries of $\boldsymbol{B}^k$. If a valid correspondence $b_{mn}$ is split into outliers $b_{m0}$ and $b_{0n}$, the new parameter matrix is

$$\tilde{\boldsymbol{M}}^k = \boldsymbol{M}^k - \boldsymbol{M}_{mn}.$$

If two outliers $b_{m0}$ and $b_{0n}$ are merged into a valid correspondence $b_{mn}$, the new parameter matrix is

$$\tilde{\boldsymbol{M}}^k = \boldsymbol{M}^k + \boldsymbol{M}_{mn}.$$

Thus, in each case the difference $(\tilde{\boldsymbol{M}}^k - \boldsymbol{M}^k)$ in Eq. (17) can be computed incrementally.

### 4.5  Multi-Node Position Optimization

At this point, the algorithm has approximate node positions from the sensor, and has recovered projective pairwise baselines (i.e., motion directions up to an unknown scale factor). This section illustrates how the baseline directions can be used to recover a globally self-consistent pose configuration. We employ an iterative algorithm that updates each node's position $\boldsymbol{p}_i$ using baseline constraints imposed by the node's neighbors. At each iteration, the list of all nodes is traversed in random order. For a given node $i$, a set of constraints is assembled by constructing rays originating at the current positions $\boldsymbol{p}_j$ of its neighbor nodes and emanating in the direction of the baselines $\boldsymbol{b}_{ji}$
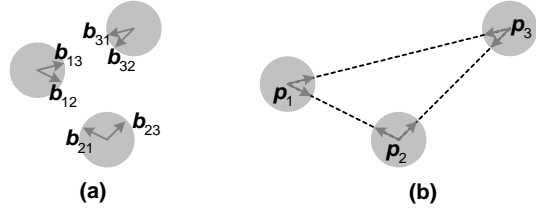


*Figure 19.* Assembling Translation Directions. (a) After motion directions are estimated between all relevant node pairs, node positions remain unknown. (b) A pose configuration consistent with all motion directions can be determined.
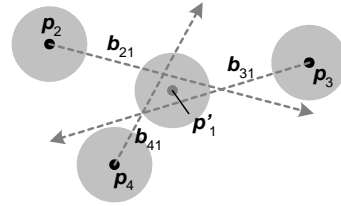


*Figure 20.* Single Node Baseline Constraints. A node's position is constrained by adjacent positions and baselines.

(Fig. 20). For perfect baselines, the new position $\boldsymbol{p}_i'$ for node $i$ is computed to minimize its sum-of-squared distance to the rays as

$$\boldsymbol{p}_i' = \left(\sum_j \boldsymbol{C}_{ji}\right)^{-1} \left(\sum_j \boldsymbol{C}_{ji} \boldsymbol{p}_j\right), \qquad (18)$$

where $\boldsymbol{C}_{ji} = \boldsymbol{I} - \boldsymbol{b}_{ji} \boldsymbol{b}_{ji}^\top$. Uncertainty in baseline directions can be incorporated by replacing $\boldsymbol{b}_{ji} \boldsymbol{b}_{ji}^\top$ in Eq. (18) with the second-moment matrix of the baseline's Bingham density. Uncertainty in $\boldsymbol{p}_i'$, in the form of a $3 \times 3$ Euclidean covariance matrix, is given by the inverse matrix in Eq. (18).

### 4.6  Metric Registration

The pose estimates produced by the method of Section 4.5 are globally self-consistent. However, they reside in an arbitrary coordinate system that does not necessarily correspond to the metric space of the scene. Thus we must deduce the rigid transformation (consisting of a translation, rotation, and scale) that expresses the node pose in this metric space, while preserving the local relationships among nodes. The sensor produces pose estimates in absolute (Earth-relative) coordinates [BdT99]. These estimates provide a ground-truth reference frame to which the node network is to be registered. We assume that the sensor is unbiased, so that the Euclidean transformation that best fits

recovered node positions to initial node positions produces an optimal pose assignment.

### 4.6.1  *Absolute Orientation.*

This section reviews a deterministic 3-D to 3-D registration algorithm [Hor87] that finds the translation, rotation, and scale that best align $N$ recovered node positions (source points) $\boldsymbol{x}_i$ with $N$ corresponding initial positions (target points) $\boldsymbol{y}_i$ (Fig. 21).
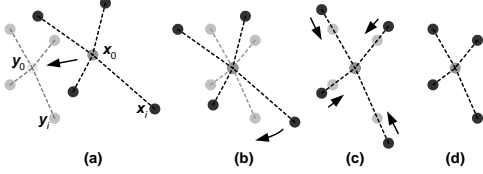


*Figure 21.* Metric Registration Process. A two-dimensional depiction of metric registration. (a) The source configuration is shifted so that the two centroids coincide. (b) Rays from the centroid to each node are rotationally aligned. (c) The optimal scale is computed and applied. (d) The final configuration.

First, each point set is translated so that its centroid is coincident with the origin. The resulting point sets are

$$\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \boldsymbol{x}_0, \qquad \tilde{\boldsymbol{y}}_i = \boldsymbol{y}_i - \boldsymbol{y}_0$$

where

$$\boldsymbol{x}_0 = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i, \qquad \boldsymbol{y}_0 = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{y}_i.$$

This allows rotation and scale to be applied relative to the same origin, namely the centroid $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$ of the two 3-D point sets.

The source points are then rotated by a matrix $\boldsymbol{R}$ to optimally align the rays through the points $\tilde{\boldsymbol{x}}_i$ and $\tilde{\boldsymbol{y}}_i$ originating at $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$. The rotation $\boldsymbol{R}$ is computed using the deterministic two-node rotation method of Section 3.3.1. Next, the optimal scale factor $s$ is computed as

$$s = \sqrt{\frac{\sum_{i=1}^{N} \tilde{\boldsymbol{y}}_i \cdot \tilde{\boldsymbol{y}}_i}{\sum_{i=1}^{N} \tilde{\boldsymbol{x}}_i \cdot \tilde{\boldsymbol{x}}_i}}.$$

Finally, the points are shifted from the origin to the target centroid $\boldsymbol{y}_0$. The overall transformation acting on the source points is thus given by

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{x}_i) &= s\boldsymbol{R}(\boldsymbol{x}_i - \boldsymbol{x}_0) + \boldsymbol{y}_0 \\ &= s\boldsymbol{R}\boldsymbol{x}_i + \boldsymbol{t} \end{aligned}$$

where $\boldsymbol{t} = \boldsymbol{y}_0 - s\boldsymbol{R}\boldsymbol{x}_0$. The next section shows how to transform pose uncertainty accordingly.

### 4.6.2  *Transforming Uncertainty.*

Let $\boldsymbol{x}$ be the 3-D position of a given camera before metric registration, with uncertainty described by a Gaussian random variable with mean $\boldsymbol{x}$ and $3 \times 3$ covariance matrix $\boldsymbol{\Lambda}_{\boldsymbol{x}}$, and let $\boldsymbol{y} = s\boldsymbol{R}\boldsymbol{x} + \boldsymbol{t}$ be the camera's position after registration. Since $\boldsymbol{y}$ is a linear transformation of $\boldsymbol{x}$, the new covariance is

$$\boldsymbol{\Lambda}_{\boldsymbol{y}} = s^2 \boldsymbol{R} \boldsymbol{\Lambda}_{\boldsymbol{x}} \boldsymbol{R}^{\top}.$$

Camera orientation is not affected by pure translation or scale; thus, orientational uncertainty is altered only by the rotation $\boldsymbol{R}$. We represent each camera's orientation by a unit quaternion $\boldsymbol{q}$, which is a Bingham random variable $\mathcal{B}_4(\boldsymbol{q}; \boldsymbol{\kappa}, \boldsymbol{U})$. Intuitively, the concentration parameters $\boldsymbol{\kappa}$ should remain unchanged by the rotation; however, the orthogonal columns of $\boldsymbol{U}$, each of which is itself a quaternion, must be transformed by $\boldsymbol{R}$. A quaternion acts on another quaternion as a matrix multiplication; thus, the new orientation quaternion $\tilde{\boldsymbol{q}}$ is given by $\tilde{\boldsymbol{q}} = \boldsymbol{Q}\boldsymbol{q}$, where $\boldsymbol{Q}$ is a $4 \times 4$ matrix representing $\boldsymbol{R}$. The same matrix transforms the columns of $\boldsymbol{U}$, resulting in a new random variable $\mathcal{B}_4(\tilde{\boldsymbol{q}}; \boldsymbol{\kappa}, \boldsymbol{Q}\boldsymbol{U})$.

## 5  Experiments

We implemented the registration algorithm in roughly 12,000 lines of C++ code. This section assesses the algorithm's end-to-end performance on both synthetic and real data using several objective metrics. Ground truth for the real data was not available due to its scope. Also, there are a number of other low-level error sources in the data, for example small internal mis-calibrations and noisy feature detection. To assess the registration algorithm, therefore, we used a variety of generic and application-specific self-consistency measures.

### 5.1  Synthetic Datasets

We generated sets of points and parallel lines in 3-D, then projected them onto unit spheres (i.e., synthetic nodes) with controllable noise levels and outlier percentages. We then evaluated the pose refinement methods of Sections 3 and 4 using these synthetic datasets.

### 5.1.1  *VP Detection and Accuracy.*

We studied the robustness and accuracy of vanishing point detection (Section 3.2) by applying the
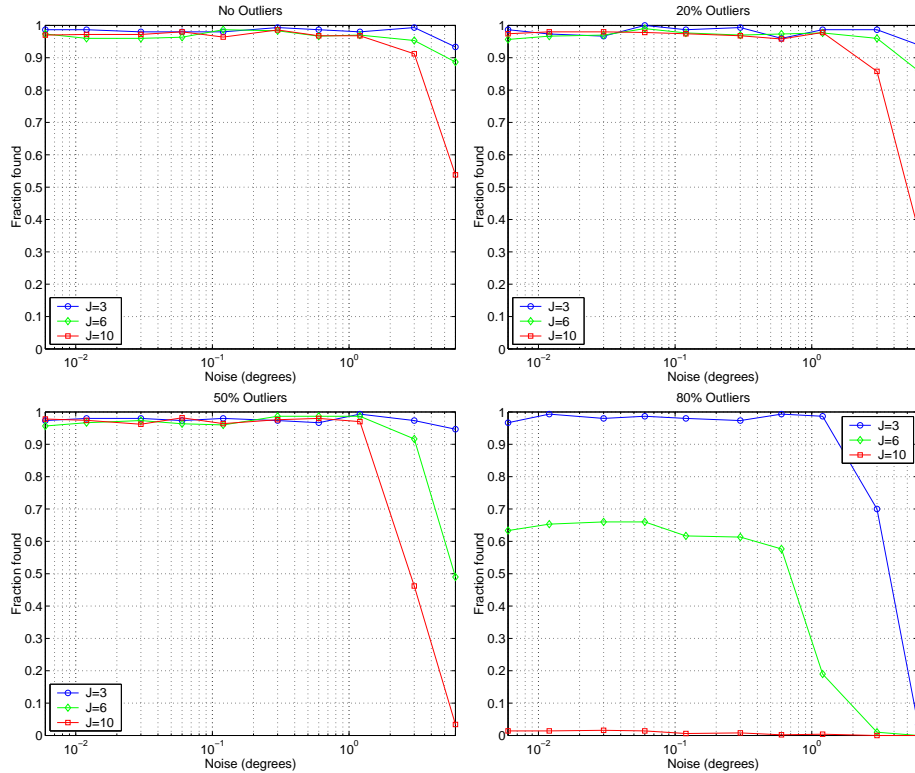
*Figure 22.* VP Detection. The percentage of true VPs detected as a function of point projection error ($x$ axis), number of true line directions ($J = 3, 6, 10$), and percentage of outlier features (zero, 20, 50, and 80%).

Hough transform EM initialization step (with roughly $1°$ cells and a $5°$ peak detection window) to 50 datasets, each with a mixture of 500 points and outliers. We determined the percentage of true peaks detected, as well as the angular deviation of the peaks from the true 3-D line directions, as a function of measurement noise, outlier percentage, and the number of true line directions (Fig. 22). Successfully detected VPs were consistently within about $1°$ of the true directions. A small number of false peaks were identified (about 2%), but only when feature noise exceeded several degrees.

We studied the subsequent EM algorithm's performance for the same parameter variations (Fig. 23). VP error grew linearly with observation noise, as expected, and remained nearly constant as the outlier percentage grew to 60%.

Overall the VP estimation method is robust, but its performance can degrade as the number of contributing vanishing points increases, because features tend to crowd the closed projective space,

causing vanishing point clusters to interfere with each other. However, real nodes typically observe fewer than six prominent line directions, so this interference effect is rare in practice.

***5.1.2    Two-Node Orientation Recovery.*** We compared the two-node stochastic registration method (Section 3.3) to classical deterministic registration using a set of four noisy 3-D line directions and outliers. Our method more reliably and accurately aligned the nodes (Fig. 24).

***5.1.3    Multi-Node  Orientation  Recovery.*** We studied the end-to-end orientation recovery method (Section 3.4) by projecting noisy VP geometry onto randomly situated nodes with noisy initial orientations (Fig. 25). The algorithm recovered accurate orientations for arbitrary initial orientation error (up to $180°$). Accuracy increased slightly as either the number of observed vanishing points, or the number of observing nodes, increased—as expected, since estimates generally improve with more data.
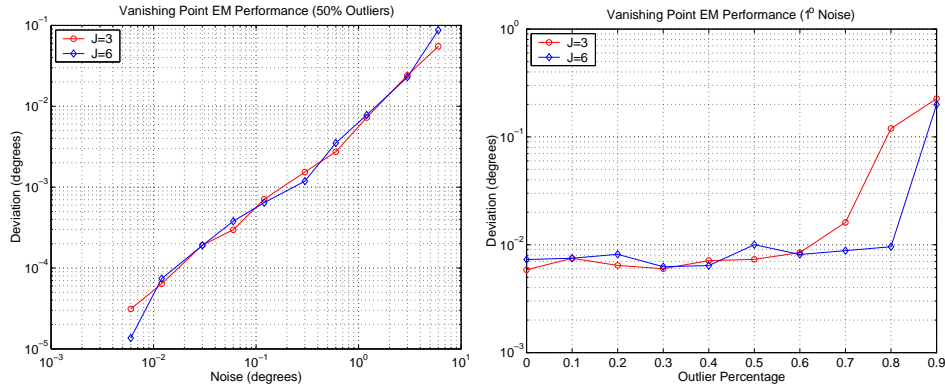
*Figure 23.* EM Vanishing Point Error. Average error in VPs estimated by the EM algorithm, as a function of line feature noise with 50% outliers (left) and outlier percentage with 1° feature noise (right).
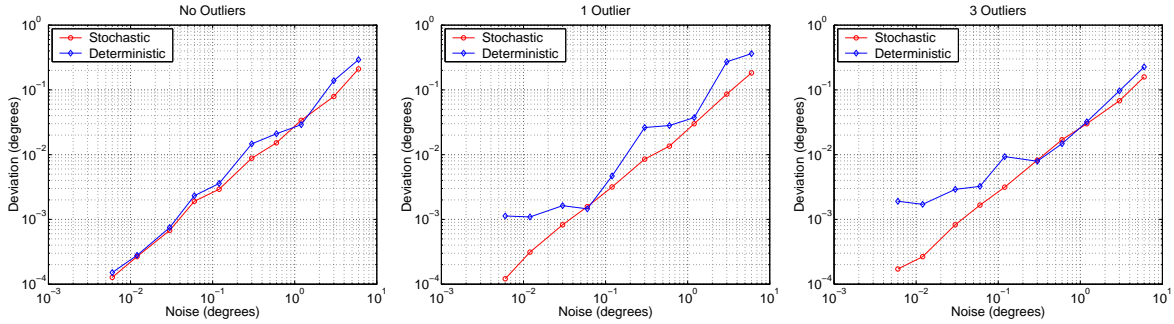


*Figure 24.* Comparison of Two-Node Rotation Methods. The stochastic two-node rotational registration technique is compared with the classical deterministic technique with four vanishing points. The plots show relative pose error as a function of vanishing point noise with 0, 1, and 3 outlier directions.
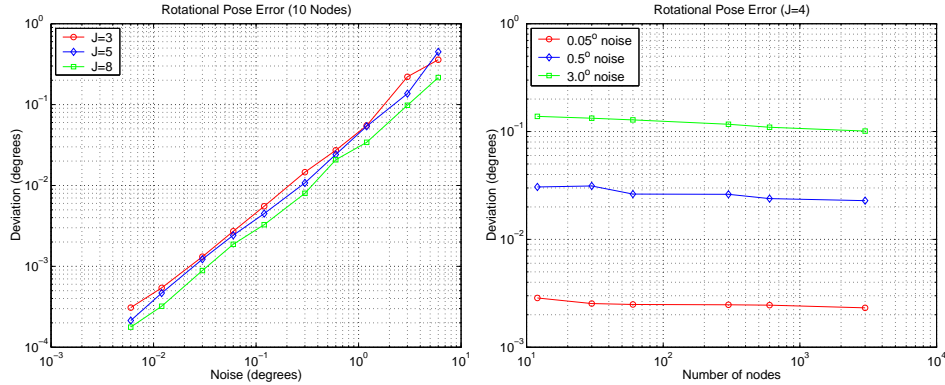


*Figure 25.* Multi-Node Orientation Performance. Average orientation error (left) as a function of vanishing point noise for 10 nodes viewing varying numbers of 3-D line directions. Orientation error (right) as a function of the number of nodes in the configuration with varying degrees of noise in 4 vanishing points.

**5.1.4    Two-Node Baseline Recovery.** We assessed the pairwise baseline estimation method (Sections 4.3, 4.4) while varying feature noise, outlier percentage (Fig. 26), input orientation error, and the number of features (Fig. 27). These experiments used initial baselines perturbed by random
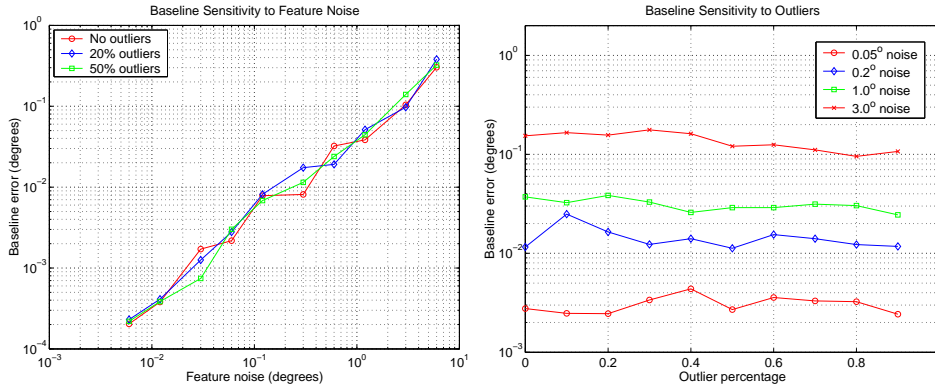
*Figure 26.* Baseline Estimation Error I. Baseline error varies roughly linearly with feature noise (left), and is roughly insensitive to the number of outliers (right).
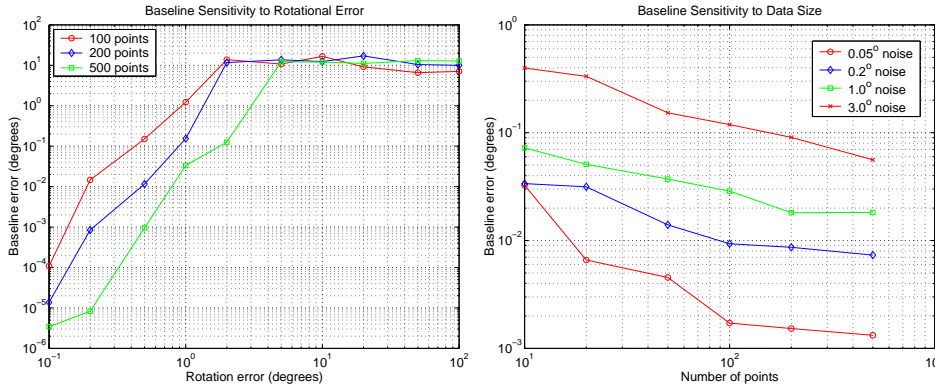


*Figure 27.* Baseline Estimation Error II. Baseline error increases rapidly with the error in supplied node orientations (left), but eventually plateaus at the baseline bound (Section 4.2.2). Error decreases with increasing number of sample points (right).

angles with variance $\sigma^2$, and an uncertainty bound of $3\sigma$ (Section 4.2.2). The algorithm reliably determines the baseline direction, even for a nine-to-one outlier to data ratio, due to the inherent robustness of the HT initialization step. The technique fared less well for noisy orientations. These violate the assumption of rotationally registered nodes (Section 4.2), thus preventing strong HT peaks.

We assessed MCEM baseline estimation (Section 4.4.2) by visualizing the evolution of the match probability matrix (Fig. 28). The method does not perfectly capture feature correspondence in the presence of noise. However, perfect correspondence is not needed; the operative performance measure is baseline accuracy, not 3-D structure.

Finally, we compared the baseline estimates obtained from MCEM to those produced by a deterministic Iterated Closest Point (ICP) method (Fig. 29). The ICP algorithm is identical to the MCEM algorithm, except that instead of estimating probabilistic match weights at each E-step, ICP determines the set of "best" one-to-one (i.e., binary) matches given the current baseline direction. As feature noise and outliers increase, MCEM is consistently more accurate than ICP.

**5.1.5    *Global Registration.*** We assessed the accuracy of the global registration stage (Section 4.5), using a set of noisy initial baselines. We recovered an end-to-end pose assignment, then compared the recovered and "ground truth" node positions (Fig. 30). As expected, position recov-
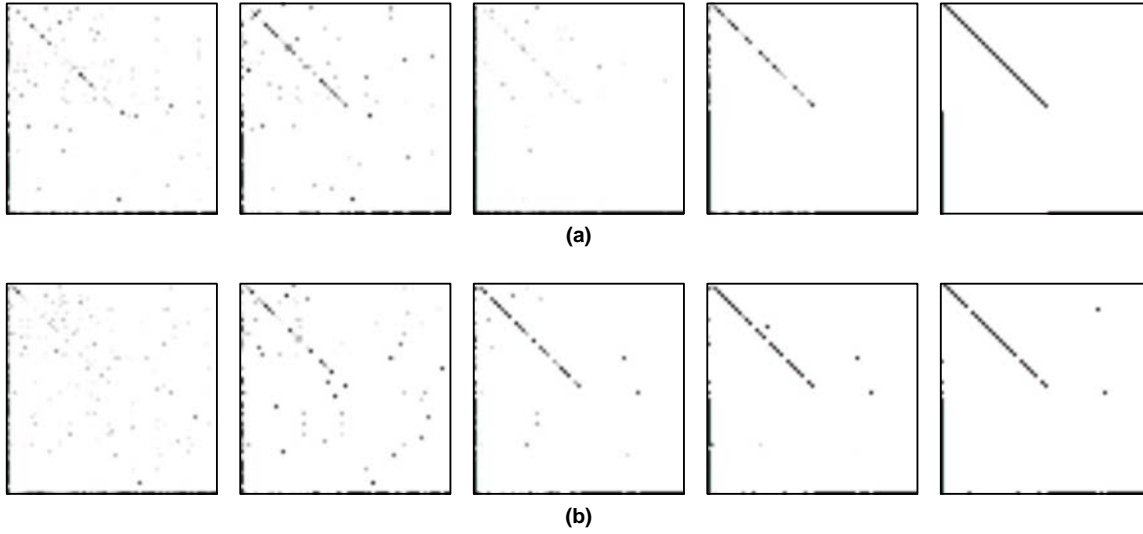
**(a)**



**(b)**

*Figure 28.* Evolution of MCEM Match Probability Matrix. Evolution of the match matrix as the MCEM algorithm proceeds. (a) Successive iterations for point feature noise of $0.05°$; correspondence is perfectly recovered. (b) Iterations for point feature noise of $0.5°$; a few features are misclassified.

ery error grew linearly with baseline perturbations smaller than $10°$ (or roughly five-meter position errors for thirty-meter baselines). The accuracy of the recovered positions did not increase significantly as the number of nodes increased, since the method uses only a constant number of constraints (one per adjacency) to update each node position.

### 5.2   Real Datasets

We assessed the end-to-end performance of the registration method for several real datasets. In lieu of ground truth, which would be difficult or impossible to obtain at this scale, we formulated and evaluated a variety of consistency metrics. We report the following quantities for each dataset:

- **Data Size and Extent.** We report the dimensions of the acquisition area in meters, the average inter-node baseline (i.e., the average distance between a node and its neighbors), and the number of narrow-FOV (raw) images, line and point features, omni-directional nodes, and adjacent node pairs.

- **Detected VPs.** We report the number of VPs detected in each node and narrow-FOV image, and the total number of distinct global line directions detected. (We define a VP as

detected in a narrow-FOV image if at least 10% of the VP's constituent 2-D line features were contributed by that image.)

- **Orthogonality Measure.** When two VPs arise from scene lines thought to be orthogonal, we report the discrepancy between the angle they form and $90°$ ("VP Ortho Error").

- **Computation Time.** We report average and total running times for each stage of the algorithm, excluding file I/O, on a 250 MHz SGI Octane with 1.5 gigabytes of memory.

- **Angular and Positional offsets.** We report the average and maximum difference between each node's initial pose (from the input) and its output pose (assigned by our algorithm) as "Rot Offset" and "Trans Offset." These quantities characterize both the quality of the system's initial pose estimates, and the robustness of the registration methods to initial pose error.

- **Consistency Measures.** We report average and maximum probability density parameters of vanishing points, node orientations, and node positions ("VP Bound," "Rot Bound," and "Trans Bound" respectively) by evaluating the size of the volume enclosing 95% of
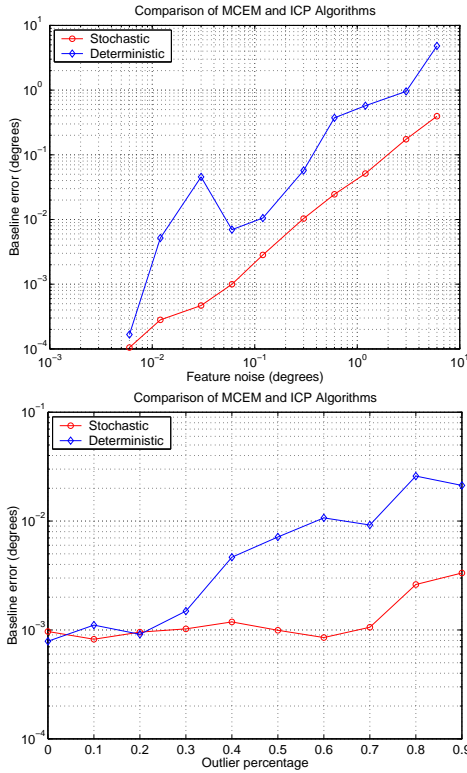
*Figure 29.* Baseline Recovery with MCEM and ICP Methods. Baseline recovery error for the (stochastic) MCEM and (deterministic) ICP methods, as a function of feature noise (top) and outlier percentage (bottom). MCEM outperforms ICP in both cases.

the underlying probability distribution.

- **Feature consistency.** We assessed end-to-end feature consistency by converting each MCEM match probability matrix to a binary match matrix. Each match probability exceeding an 80% threshold was interpreted as an unambiguous match, and its constituent point features were examined using two error measures. We tabulate the average and maximum 3-D distance (in centimeters) between rays extruded from each node through the point feature ("3-D Ray Error"), and the average and maximum 2-D distance (in pixels) between each point feature and its epipolar line in the other node ("2-D Epi Error").

**5.2.1    Tech Square.** This dataset consists of 81 nodes spanning an area of roughly 285 by 375 meters. The average inter-node baseline was 30.9 meters. The rotation stage registered 75 (or roughly
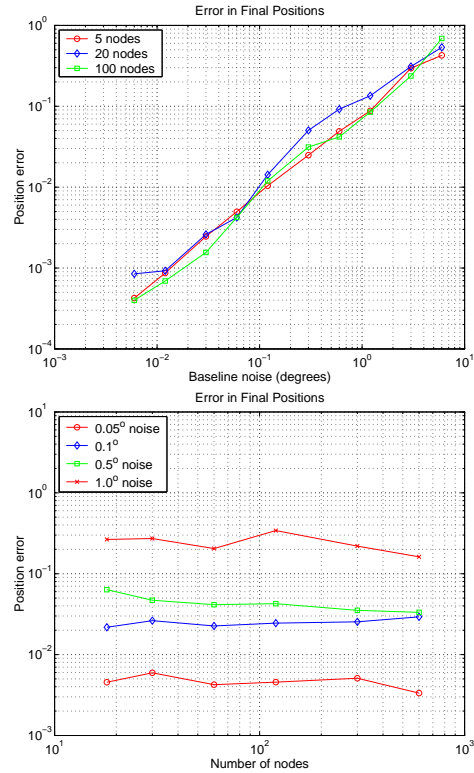


*Figure 30.* Global Position Recovery. Error in global position recovery as a function of baseline error (top) and number of nodes (bottom).

92%) of the nodes, discarding 6 with fewer than two VPs. The translation stage registered all remaining nodes successfully.
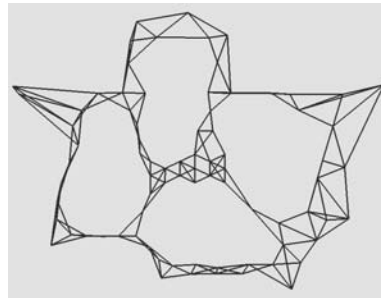


*Figure 31.* Tech Square Node Configuration. The average baseline was 30.9 meters.

Our algorithm corrected initial pose errors of over 17° and nearly 7 meters. It recovered global pose consistent on average to within 0.072°, 5.6 cm, and 1.22 pixels. The maximum pose error for any node was 0.098° of orientation, 11.0 cm of position, and 5.71 pixels. Total CPU time was just under three hours.

*Table 1.* Tech Square Dataset and Run Times.

| Data Type | Per Image | Per Node | Total |
|---|---|---|---|
| Images | — | 48 | 3899 |
| Line Features | 218 | 10,516 | 851,819 |
| Point Features | 227 | 10,958 | 887,598 |
| Nodes | — | — | 81 |
| Node Adjacencies | — | — | 189 |
| VPs | 0.69 | 3.6 | 9 |

| Registration Stage | Time Per Node | Total Run Time | |
|---|---|---|---|
| VP Hough | 0.2 sec | | 15s |
| VP EM | 6.7 sec | 7m | 54s |
| Rotation EM | 0.6 sec | | 46s |
| Baseline Hough | 18.9 sec | 25m | 31s |
| Baseline MCEM | 95.9 sec | 2h | 23m |
| Global Opt | 0.7 sec | | 53s |
| Total | 2.2 min | 2h 58m | 19s |

We compared the orientations computed by the algorithm to those produced by a manual, 6-DOF bundle-adjustment [CMT98]. Interactive inspection of VPs in the manually registered dataset reveal that it does not represent ideal ground truth. Because the number of nodes and adjacencies was so large, the human operator naturally specified as few constraints as possible for convergence of the underlying optimization. This produced unstable constraint sets, and rather poor global pose. Figure 32 compares epipolar geometry for manual and automated pose recovery.

Figure 33 compares epipolar geometry for a window corner from a repeating series of windows obscured by foliage. The manual solution has poor epipolar geometry in this region, since the human

*Table 2.* Tech Square Consistency Measures.

| Measure | Avg. | Max. |
|---|---|---|
| Rot Offset | 1.53° | 17.18° |
| VP Bound | 0.18° | 0.80° |
| Rot Bound | 0.072° | 0.098° |
| VP Ortho Error | 0.056° | 0.09° |
| Trans Offset | 0.70 m | 6.70 m |
| Trans Bound | 5.6 cm | 11.0 cm |

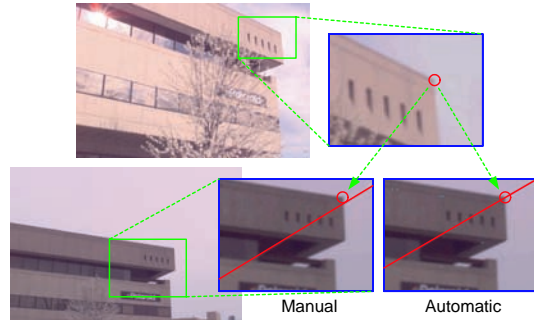| | Avg. | Max. | Std. Dev. |
|---|---|---|---|
| 3-D Ray Distance | 9.6 cm | 12.4 cm | 3.3 cm |
| 2-D Epi Distance | 1.22 pix | 5.71 pix | 2.33 pix |



*Figure 32.* Tech Square Epipolar Geometry I. A point feature in one image and its corresponding epipolar line in another image, as computed using node pose recovered from manual correspondence (bottom middle) vs. our algorithm (bottom right). Note the error in the manual solution, in this case due to insufficient manually-specified match constraints.

user did not enter match constraints here. It is plainly impossible to match these window corners *given only this pair of images*, due to the image's limited FOV; even given omni-directional image pairs, human operators find it difficult or impossible to match such features due to the severe visual clutter. In contrast, our algorithm recovers accurate epipolar geometry.
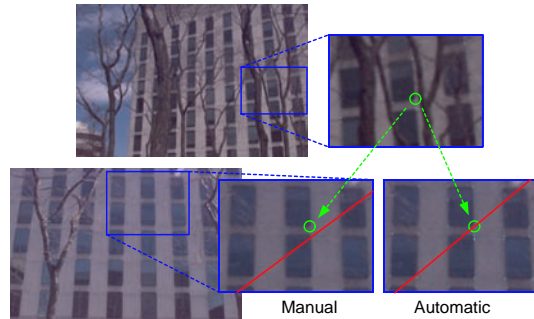


*Figure 33.* Tech Square Epipolar Geometry II. A feature whose match is difficult for a human operator to identify. Epipolar geometry is shown for manual (bottom middle) and automated (bottom right) pose solutions. Note the error in the manual solution.

**5.2.2   Green Building.** We tested the end-to-end registration method using thirty nodes with particularly noisy initial pose, spanning an area of roughly 80 by 115 meters (Fig. 34). The algorithm proved robust, registering all nodes successfully and correcting initial pose errors of nearly seven degrees, six meters, and hundreds of pixels (Fig. 35). The resulting pose estimates were consistent on average to 0.067° of orientation, 4.5 cm
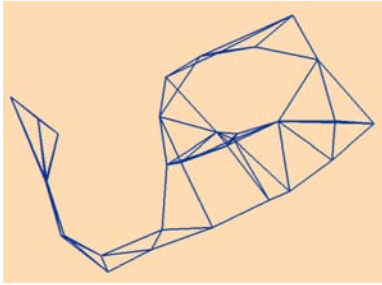
*Figure 34.* Green Building Node Configuration. The average baseline was 15.6 meters.

of position, and 2.21 pixels. The maximum pose error for any node was 0.12° of orientation, 8.1 cm of position, and 4.17 pixels. Total CPU time was just over one hour.

*Table 3.* Green Building Dataset and Run Times.

| Data Type | Per Image | Per Node | Total |
|---|---|---|---|
| Images | — | 23 | 695 |
| Line Features | 237 | 5,498 | 164,945 |
| Point Features | 257 | 5,967 | 179,030 |
| Nodes | — | — | 30 |
| Node Adjacencies | — | — | 80 |
| VPs | 0.35 | 3.3 | 5 |

| Registration Stage | Time Per Node | Total Run Time | |
|---|---|---|---|
| VP Hough | 0.1 sec | | 3s |
| VP EM | 2.9 sec | 1m | 28s |
| Rotation EM | 0.6 sec | | 18s |
| Baseline Hough | 16.5 sec | 8m | 16s |
| Baseline MCEM | 112.7 sec | 56m | 20s |
| Global Opt | 0.7 sec | | 21s |
| Total | 2.2 min | 1h 6m | 46s |

*Table 4.* Green Building Consistency Measures.

| Measure | Avg. | Max. |
|---|---|---|
| Rot Offset | 2.95° | 6.83° |
| VP Bound | 0.092° | 0.52° |
| Rot Bound | 0.067° | 0.12° |
| VP Ortho Error | 0.047° | 0.11° |
| Trans Offset | 2.86 m | 5.97 m |
| Trans Bound | 4.5 cm | 8.1 cm |

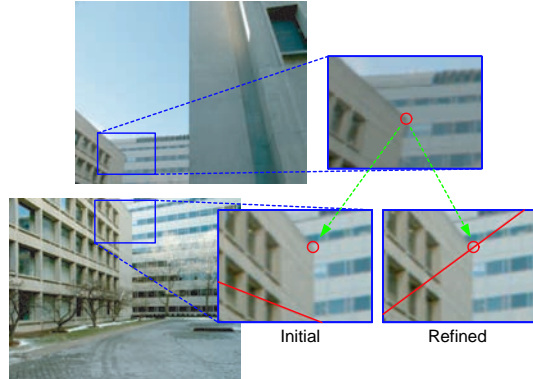| | Avg. | Max. | Std. Dev. |
|---|---|---|---|
| 3-D Ray Distance | 10.2 cm | 18.5 cm | 5.3 cm |
| 2-D Epi Distance | 2.21 pix | 4.17 pix | 1.43 pix |



*Figure 35.* Green Building Epipolar Geometry I. Initial and refined epipolar geometry; the algorithm corrects significant initial pose error.

Many nodes had particularly noisy initial height estimates (Fig. 36). We studied the algorithm's ability to recover consistent node heights.
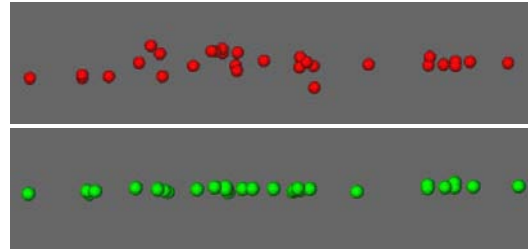


*Figure 36.* Green Building Pose Refinement. (a) A horizontal view of the node topology before pose refinement. All nodes were acquired at roughly the same height above level ground, but noisy GPS caused poor initial height estimates for the nodes. (b) Refinement corrects most of the height variation.

Finally, we studied the accuracy of epipolar geometry for distant points (Fig. 37).
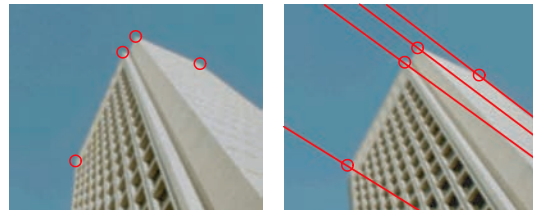


*Figure 37.* Green Building Epipolar Geometry II. The refined epipolar geometry (shown here) is consistent to within a few pixels, even for distant 3-D points. The initial pose was so poor that the epipolar lines of the features at left lie entirely outside the image at right.

**5.2.3 Ames Court.** This dataset consists of 100 nodes spanning roughly 315 by 380 meters

(Fig. 38). Of these, the rotation stage registered 95 successfully. The translation stage registered all remaining nodes. Initial pose was corrected by 5.59° and 6.18 meters, achieving average consistency of 0.095°, 5.7 cm, and 3.88 pixels. The maximum pose inconsistency was 0.21°, 8.8 cm, and 5.02 pixels. Total CPU time was just over four hours.
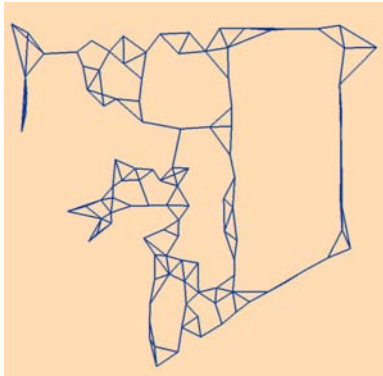


*Figure 38.* Ames Court Node Configuration. The average baseline was 23.5 meters.

### 5.2.4  *Benefit of Wide Field of View.*

Researchers have pointed out the desirability of wide-FOV imagery in a variety of contexts [Kan93, FA98, GN98]. Qualitatively, our system exploits wide FOV to collect more vanishing point observations for the rotation stage, and more 3-D point

*Table 5.* Ames Court Dataset and Run Times.

| | Per Image | Per Node | Total |
|---|---|---|---|
| Images | — | 20 | 2,000 |
| Line Features | 228 | 4,562 | 456,246 |
| Point Features | 257 | 4, 132 | 413,254 |
| Nodes | — | — | 100 |
| Node Adjacencies | — | — | 232 |
| VPs | 0.43 | 3.2 | 8 |

| Registration Stage | Time Per Node | Total Run Time | |
|---|---|---|---|
| VP Hough | 0.1 sec | | 10 s |
| VP EM | 2.5 sec | 4 m | 22 s |
| Rotation EM | 0.3 sec | | 33 s |
| Baseline Hough | 18.1 sec | 30 m | 10 s |
| Baseline MCEM | 122.4 sec | 3 h | 24 m |
| Global Opt | 0.6 sec | 1 m | 4 s |
| Total | 2.4 min | 4 h  0 m | 19 s |

*Table 6.* Ames Court Consistency Measures.

| Measure | Avg. | Max. |
|---|---|---|
| Rot Offset | 2.83° | 5.59° |
| VP Bound | 0.23° | 0.74° |
| Rot Bound | 0.095° | 0.21° |
| VP Ortho Error | 0.043° | 0.09° |
| Trans Offset | 3.53 m | 6.18 m |
| Trans Bound | 5.7 cm | 8.8 cm |

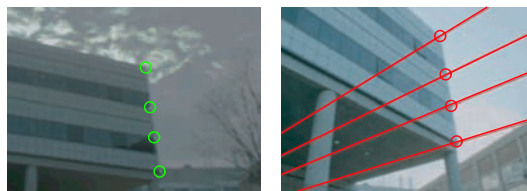| | Avg. | Max. | Std. Dev. |
|---|---|---|---|
| 3-D Ray Distance | 14.9 cm | 20.2 cm | 5.6 cm |
| 2-D Epi Distance | 3.88 pix | 5.02 pix | 2.10 pix |



*Figure 39.* AmesCourt Epipolar Geometry. Point features and corresponding epipolar lines for a typical node pair in the AmesCourt set.

features for the translation stage, increasing the likelihood that common structure will be identified across nearby images.

We gathered quantitative evidence that increasing the FOV enables more reliable pose estimation. We isolated the effect of increasing FOV on two components of pose recovery. First, we applied the vanishing point Hough transform to a single node, varying the node's field of view (Fig. 40) to assess the sharpness of the HT peak arising from a particular VP. Second, we applied the baseline Hough transform to a single node pair, varying both nodes' FOV (Fig. 41) to assess the sharpness of the HT peak arising from epipolar accumulation. In both cases, as the FOV widened, the respective Hough transform peak became more evident (i.e., more reliably detectable). Narrow-FOV images (e.g., the tiles in Fig. 1b) did not provide sufficient feature overlap for convergence in any of our datasets.

### 5.2.5  *Low-Level Error Sources.*

The algorithm described in this paper forms one of a series of automated processing stages in a large-scale, outdoor model capture system. Other stages introduce and propagate error. We estimate the low-level system noise as follows. Our raw image
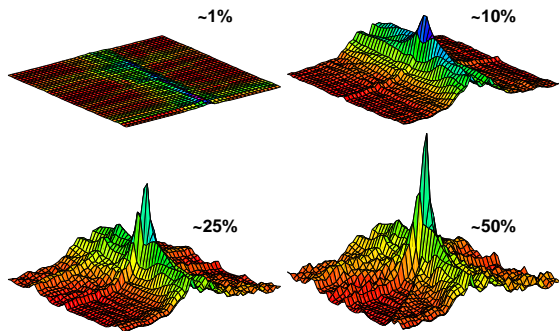
*Figure 40.* VP HT Peak Coherence. Dependence of Hough transform accumulation values on field of view, for a single vanishing point in a single node. Transform values are plotted ($z$ axis) over one face of the discretization cube ($x$ and $y$ axes) as node FOV varies from roughly 1% to 50% of the sphere.
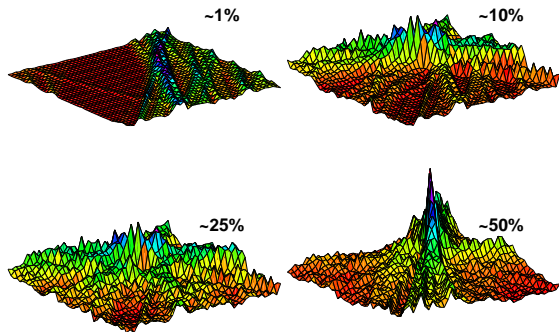


*Figure 41.* Baseline HT Peak Coherence. Dependence of Hough transform accumulation values on field of view, for a single baseline shared by two nodes. Transform values are plotted over the initial angular uncertainty region, as node FOV varies from roughly 1% to 50% of the sphere.

sensor has a resolution of roughly 1 milliradian ($0.05°$) per pixel. The pinhole calibration, radial distortion correction, mosaic generation, and feature localization stages [TAB+01] are each accurate to roughly half a pixel. Thus we estimate that the cumulative image-space localization error is between two and three pixels – roughly the end-to-end accuracy of our algorithm. No data fusion algorithm can overcome this "noise floor" without more comprehensive error modeling.

# 6    Related Work

There is a great deal of literature on extrinsic calibration [Hor86, HZ00, FLP01]. This section reviews methods that decouple geometry, address large scale or unknown correspondence, or model geometric uncertainty.

## 6.1    Decoupled Geometry

A number of authors have proposed geometric factoring to reduce the number of DOFs that must be estimated, or serialize their estimation.

Many interactive systems allow the user to identify vanishing points for camera calibration [TK92, BB95, DTM96, SHS98, CRB99]. Other systems detect and track VPs automatically for egomotion estimation [STI90, LM96].

Projective reconstruction techniques avoid intrinsic camera calibration [MZ92, LF97], recovering structure and pose only up to an arbitrary projective transformation. Other SFM methods have been proposed, for example using SVD [PK94] or random search [ARS00].

## 6.2    Scale and Extent

This section reviews methods that address large input sizes (many images or features) or large dimensional extent (long image baselines or camera excursions).

A variety of structure from motion techniques recover scene geometry and camera egomotion from multiple images [SK94, AP95, LMD95, PZ98]. Methods that use local constraints alone (e.g., [FZ98]) are prone to bias and error accumulation. None of these methods have been demonstrated for noisy observations, significant occlusion and clutter, varying illumination, and wide baselines.

Many researchers have used the Hough transform (e.g., [Bar83, Col93, LMLK94, Shu99]) for efficient processing of large feature sets. However, HT accuracy is inherently limited by discretization, and no principled characterization of uncertainty has been demonstrated in this setting. The HT has been used to initialize continuous-space VP estimation [Col93]. This method uses deterministic clustering and outlier rejection, biasing the resulting VP estimates.

## 6.3    Unknown Correspondence

Determining correspondence is a central problem in computer vision. A variety of interactive tools rely on a human operator to match features across images [BB95, DTM96, SHS98]. These tools do not scale effectively, and are vulnerable to operator error and numerical instability.

Low-level feature trackers [SKS95] and texture trackers [HS81, ZMI00] recover correspondence

under short-baseline motion (i.e., constant brightness and little or no occlusion).

Several robust statistical techniques randomly generate match sets or aligning transformations, selecting the most consistent [FB81, CC91, FZ98, Ste98, TZ00, ARS00]. The speed and accuracy of these methods degrade significantly with increasing occlusion or clutter (as the likelihood of choosing the correct match set, or distinguishing the correct transformation, decreases).

Several EM techniques use probabilistic correspondence to solve coupled structure and motion problems in the presence of matching ambiguity. Some assume that all features are visible in all images [DSTT00], while others treat only image pairs [RCD99, CR00a]. Another method avoids correspondence by maximizing texture correlation [FL94].

### 6.4   Geometric Uncertainty

Sophisticated noise-handling methods exist for least-squares [GL80] and computer vision problems [Ste85, AF84, MM00]. These formulations assume additive Gaussian noise, which is not always appropriate. For example, when recovering the fundamental matrix [CZZF97, Zha98], the physical meaning of its covariance (i.e., its units) is unclear.

Some investigators have used projective probability distributions for inference from noisy features [CW90, Kan94]. Others have used a hybrid of projective and Euclidean distributions to model uncertain 6-DOF pose [NS94]. Other methods propose errors-in-variables models for spherical regression [Cha89, Pre89], but consider only extreme cases in which data points have tightly concentrated symmetric bipolar distributions.

## 7   Summary and Discussion

This section analyzes the running time (Section 7.1) and limitations (Section 7.2) of the end-to-end registration algorithm, and summarizes the paper's contributions (Section 7.3).

### 7.1   Asymptotic Time and Space

The number of adjacencies in the node graph is linear in the number of nodes $n$, since every node has at most a constant number of neighbors. VP estimation expends $\mathcal{O}(\ell)$ time per node, where $\ell$ is the number of line features in the node (assuming

$J$ is constant). The global rotation stage aligns each node independently to common scene structure, expending $\mathcal{O}(n)$ time overall.

The pairwise baseline estimation stage requires $\mathcal{O}(f^2)$ time per adjacency, where $f$ is the maximum number of point features in the two nodes involved; in practice we bound $f$ by a constant. Thus this stage requires $\mathcal{O}(n)$ time as well. Each of the $\mathcal{O}(1)$ iterations of global translation computes $n$ new node positions, expending $\mathcal{O}(1)$ time per node, or $\mathcal{O}(n)$ time overall. The final metric alignment stage runs in $\mathcal{O}(n)$ time.

Thus, the end-to-end (6-DOF) registration algorithm requires $\mathcal{O}(n)$ time.

### 7.2   Limitations

Our algorithm has several limitations. It requires line and point features. It assumes that adjacent nodes are likely to have observed overlapping scene structure; this assumption may fail, for example when two nodes lie on opposite sides of a thin building or wall, or when poor initial position estimates produce "adjacent" nodes that are in fact far apart (and thus view little in common). The Earth's curvature limits rotational registration to node networks spanning less than about ten kilometers; larger areas have a local vertical that varies by more than $0.1°$, our current implementation's error floor. The $\mathcal{O}(J^6)$ running time of brute-force VP correspondence (Section 3.3.3) can be prohibitive for large numbers of VPs. Finally, positional registration relies on pairwise baseline estimates, so can be unstable for degenerate or incorrect adjacency networks.

### 7.3   Contributions

Previous methods for exterior image calibration scale poorly, either because they rely on a human operator, or because they expend computational resources that grow too rapidly with input size. Our contribution stems from the observation that a suitably crafted algorithm can use scale to advantage, exploiting redundant input to reliably overcome noise, ambiguity and clutter.

This motivated our development of a linear-time algorithm combining a number of existing techniques from computer vision and estimation theory: decoupling pose estimation into two 3-DOF problems (to reduce the dimensionality of the parameter space); probabilistic inference on

the sphere (for accurate estimation from noisy projective data); the Hough transform (for efficient, robust initialization); EM algorithms (for probabilistic classification); and MCMC methods (for sampling from high-dimensional probability spaces).

In addition, we introduce a number of novel techniques. First, we exploit *a priori* absolute position estimates, and an image adjacency graph, to limit inter-camera registration to those images likely to have observed common scene structure. This removes the need for a human operator, or expensive pairwise-search process, to identify overlapping images or features.

Second, we use wide-FOV (omni-directional) images to maximize the information obtained from each vantage point. We demonstrate quantitative evidence that the robustness and accuracy of pose estimation improve as the camera's FOV increases.

Third, we extend classical, deterministic camera alignment, VP detection, and point correspondence methods to a stochastic projective framework, handling unknown numbers of features, occlusion, and outliers. This improves accuracy by fusing many noisy observations into a few ensemble quantities (VPs and baselines).

Fourth, we combine the robustness of the Hough transform with the accuracy of EM into a hybrid technique that overcomes the limitations of each method alone.

Finally, we assess the algorithm's asymptotic resource usage and end-to-end performance using a variety of objective measures, enabling meaningful comparison to other methods.

## 8    Conclusion

This paper presented a scalable, end-to-end algorithm that registers thousands of images using hundreds of thousands of noisy features. Most previous registration methods have assumed small inputs, short baselines, constant illumination, and unoccluded, uncluttered scenes. Our method relaxes all of these assumptions simultaneously, yet expends only linear time in the size of the input.

We demonstrated the algorithm on image datasets acquired outdoors, over wide baselines and hundreds of meters, under uncontrolled and varying lighting conditions, and in the presence of significant occlusion and visual clutter. It pro-

duces absolute pose estimates consistent to within a tenth of a degree, five centimeters, and four pixels, representing a new end-to-end capability for automated, absolute registration of terrestrial, omni-directional image networks in urban scenes.

## Appendix: Correspondence Sets

This section illustrates the enormity of the sample space in Section 4.4.1 by counting the number of possible correspondence sets. Assume for the moment that the number of valid matches $F$ is known. A particular correspondence set is represented by an $M \times N$ binary matrix $\boldsymbol{B}$ containing at most one non-zero entry per row and per column. A non-zero entry in position $(i, j)$ represents a correspondence between features $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$.

Initially, $\boldsymbol{B}$ contains all zeros. Imagine placing $F$ correspondences in the matrix one at a time while preserving the condition in Eq. (9). The first correspondence can be placed in $MN$ possible ways (i.e. anywhere in the matrix). The second, however, has only $(M-1)(N-1)$ possibilities, because it cannot be placed in the same row or column as the first; the third has $(M-2)(N-2)$ possibilities, and so on. The number of possible match sets $S_F$ is therefore

$$
\begin{aligned}
S_F F! &= (M)(N)(M-1)(N-1)\cdots \\
&\quad \cdot (M-F+1)(N-F+1) \\
&= (M)(M-1)\cdots(M-F+1) \\
&\quad \cdot (N)(N-1)\cdots(N-F+1) \\
&= \binom{M}{F}\binom{N}{F}(F!)^2.
\end{aligned}
$$

Here $S_F$ is multiplied by $F!$ to account for permutations of the chosen matches; the placement order is unimportant.

Since $F$ can itself vary, the total number of possible match sets $S$ arises from a sum over $F$:

$$
S = \sum_{F=0}^{F'} S_F = \sum_{F=0}^{F'} \binom{M}{F}\binom{N}{F} F!,
$$

where $F' = \min(M, N)$. For a node pair observing only 20 features (so that $M = N = 20$), $S$ is approximately $10^{21}$.

## Acknowledgements

# References

[AF84]   Yasuo Amemiya and Wayne A. Fuller. Estimation for the multivariate errors-in-variables model with estimated error covariance matrix. *Annals of Statistics*, 12(2):497–509, June 1984.

[AP95]   A. Azarbayejani and A. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):562–575, June 1995.

[ARS00]  A. Adam, E. Rivlin, and I. Shimshoni. ROR: Rejection of outliers by rotations in stereo matching. In *Proc. CVPR*, pages 2–9, June 2000.

[AT00]   Matthew Antone and Seth Teller. Automatic recovery of relative camera rotations for urban scenes. In *Proc. CVPR*, pages 282–289, June 2000.

[Bar83]  Stephen T. Barnard. Methods for interpreting perspective images. *Artificial Intelligence*, 21:435–462, 1983.

[BB95]   Shawn Becker and V. Michael Bove. Semi-automatic 3-D model extraction from uncalibrated 2-D camera views. In *Proc. SPIE Image Synthesis*, volume 2410, pages 447–461, February 1995.

[BdT99]  Michael Bosse, Douglas de Couto, and Seth Teller. Eyes of Argus: Georeferenced imagery in urban environments. *GPS World*, pages 20–30, April 1999.

[Ber79]  Rudolph Beran. Exponential models for directional data. *Annals of Statistics*, 7(6):1162–1178, November 1979.

[Bin74]  Christopher Bingham. An antipodally symmetric distribution on the sphere. *Annals of Statistics*, 2(6):1201–1225, November 1974.

[Can86]  John F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[CC91]   Subhasis Chaudhuri and Shankar Chatterjee. Robust estimation of 3-D motion parameters in presence of correspondence mismatches. In *Proc. Asilomar Conference on Signals, Systems and Computers*, pages 1195–1199, November 1991.

[Cha89]  Ted Chang. Spherical regression with errors in variables. *Annals of Statistics*, 17(1):293–306, March 1989.

[CMT98]  Satyan Coorg, Neel Master, and Seth Teller. Acquisition of a large pose-mosaic dataset. In *Proc. CVPR*, pages 872–878, June 1998.

[Col93]  Robert T. Collins. *Model Acquisition using Stochastic Projective Geometry*. PhD thesis, University of Massachusetts, September 1993.

[CR00a]  Haili Chui and Anand Rangarajan. A feature registration framework using mixture models. In *Proc. IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 190–197, 2000.

[CR00b]  Haili Chui and Anand Rangarajan. A new algorithm for non-rigid point matching. In *Proc. CVPR*, pages 44–51, June 2000.

[CRB99]  Roberto Cipolla, Duncan Robertson, and Edmond Boyer. Photobuilder - 3d models of architectural scenes from uncalibrated images. In *ICMCS*, volume 1, pages 25–31, 1999.

[CW90]   Robert T. Collins and R. Weiss. Vanishing point calculation as statistical inference on the unit sphere. In *Proc. ICCV*, pages 400–403, December 1990.

[CZZF97] G. Csurka, C. Zeller, Z. Zhang, and O. Faugeras. Characterizing the uncertainty of the fundamental matrix. *Computer Vision and Image Understanding*, 68(1):18–36, October 1997.

[DLR77]  A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[DSTT00] Frank Dellaert, Steven M. Seitz, Charles E. Thorpe, and Sebastian Thrun. Structure from motion without correspondence. In *Proc. CVPR*, pages 557–564, June 2000.

[DTM96]  Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proc. SIGGRAPH*, pages 11–20, 1996.

[FA98]   Cornelia Fermüller and Yiannis Aloimonos. Ambiguity in structure from motion: Sphere versus plane. *IJCV*, 28(2):137–154, 1998.

[FB81]   Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[FL94]   P. Fua and Y. G. Leclerc. Registration without correspondence. In *Proc. CVPR*, pages 121–128, June 1994.

[FLP01]   Olivier Faugeras, Q.-T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images.* MIT Press, Cambridge, MA, 2001.

[FZ98]   A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Proc. ECCV*, pages 311–326, June 1998.

[GL80]   Gene H. Golub and Charles F. Van Loan. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis*, 17(6):883–893, December 1980.

[GN98]   Joshua Gluckman and Shree Nayar. Egomotion and omnidirectional cameras. In *ICCV*, pages 35–42, 1998.

[Hor86]   Berthold Klaus Paul Horn. *Robot Vision.* MIT Press, Cambridge, MA, 1986.

[Hor87]   Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. of the Optical Society of America A*, 4(4):629–642, April 1987.

[HS81]   Berthold. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 16(1–3):185–203, August 1981.

[HZ00]   Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, Cambridge, 2000.

[JM79]   P. E. Jupp and K. V. Mardia. Maximum likelihood estimators for the matrix von Mises-Fisher and Bingham distributions. *Annals of Statistics*, 7(3):599–606, May 1979.

[Kan93]   K. Kanatani. 3-d interpretation of optical flow by renormalization. *IJCV*, 11:267–282, 1993.

[Kan94]   Kenichi Kanatani. Statistical analysis of geometric computation. *Computer Vision Graphics and Image Processing*, 59(3):286–306, May 1994.

[KGV83]   S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[LF97]   Q. T. Luong and O. Faugeras. Camera calibration, scene motion, and structure recovery from point correspondences and fundamental matrices. *IJCV*, 22(3):261–289, 1997.

[LM96]   John C. H. Leung and Gerard F. McLean. Vanishing point matching. In *Proc. ICIP*, volume 2, pages 305–308, 1996.

[LMD95]   Mi-Suen Lee, Gerard Medioni, and Rachid Deriche. Structure and motion from a sparse set of views. In *Proc. the International Symposium on Computer Vision*, pages 73–78, November 1995.

[LMLK94]   Evelyne Lutton, Henri Maître, and Jaime Lopez-Krahe. Contribution to the determination of vanishing points using Hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(4):430–438, April 1994.

[MM00]   Bogdan Matei and Peter Meer. A general method for errors-in-variables problems in computer vision. In *Proc. CVPR*, pages 18–25, June 2000.

[MRR+53]   N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. of Chemical Physics*, 21(6):1087–1092, 1953.

[MZ92]   Joseph L. Mundy and Andrew Zisserman, editors. *Geometric Invariance in Computer Vision.* MIT Press, Cambridge, MA, 1992.

[NS94]   Keith E. Nicewarner and A. C. Sanderson. A general representation for orientation uncertainty using random unit quaternions. In *Proc. IEEE International Conference on Robotics and Automation*, volume 2, pages 1161–1168, May 1994.

[PK94]   C. J. Poelman and Takeo Kanade. A paraperspective factorization method for shape and recovery. In *Proc. ECCV*, pages 97–108, May 1994.

[Pre89]   Michael J. Prentice. Spherical regression on matched pairs of orientation statistics. *J. of the Royal Statistical Society, Series B*, 51(2):241–248, 1989.

[PZ98]   Philip Pritchett and Andrew Zisserman. Matching and reconstruction from widely separated views. In *Proc. Workshop on 3-D Structure from Multiple Images of Large-Scale Environments*, pages 78–92, June 1998.

[RCD99]   Anand Rangarajan, Haili Chui, and James S. Duncan. Rigid point feature registration using mutual information. *Medical Image Analysis*, 4:1–17, 1999.

[Riv84]   Louis-Paul Rivest. On the information matrix for symmetric distributions on the unit hypersphere. *Annals of Statistics*, 12(3):1085–1089, September 1984.

[SHS98]   Harry S. Shum, Mei Han, and Richard Szeliski. Interactive construction of 3D models from panoramic image mosaics. In *Proc. CVPR*, pages 427–433, 1998.

[Shu99]   Jefferey A. Shufelt. Performance evaluation and analysis of vanishing point detection techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):282–288, March 1999.

[Sin64]   Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Annals of Mathematical Statistics*, 35(2):876–879, June 1964.

[SK94]   Richard Szeliski and S. B. Kang. Recovering 3D shape and motion from image streams using nonlinear least squares. *J. of Visual Communication and Image Representation*, 5(1):10–28, March 1994.

[SKS95]   R. Szeliski, Sing Bing Kang, and Heung-Yeung Shum. A parallel feature tracker for extended image sequences. In *Proc. International Symposium on Computer Vision*, pages 241–246, 1995.

[Ste85]   Leonard A. Stefanski. The effects of measurement error on parameter estimation. *Biometrika*, 72(3):583–592, December 1985.

[Ste98]   Gideon P. Stein. Tracking from multiple view points: Self-calibration of space and time. In *Proc. CVPR*, pages 521–527, November 1998.

[STI90]   Li Shigang, Saburo Tsuji, and Masakazu Imai. Determining of camera rotation from vanishing points of lines on horizontal planes. In *Proc. ICCV*, pages 499–502, 1990.

[TAB+01]   Seth Teller, Matthew Antone, Michael Bosse, Satyan Coorg, Manish Jethwa, and Neel Master. Calibrated, registered images of an extended urban area. In *Proc. CVPR*, pages I–813–I–820, Dec. 2001.

[Tel97]   Seth Teller. Automatic acquisition of hierarchical, textured 3D geometric models of urban environments: Project plan. In *Proc. of the Image Understanding Workshop*, 1997.

[Tel01]   Seth Teller. Scalable, controlled image capture in urban environments. Technical Report 825, MIT LCS, September 2001.

[TK92]   Camillo J. Taylor and David J. Kriegman. Structure and motion from line segments in multiple images. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1615–1620, May 1992.

[TPG97]   Tinne Tuytelaars, Marc Proesmans, and Luc Van Gool. The cascaded Hough transform. In *Proc. ICIP*, volume 2, pages 736–739, 1997.

[TZ00]   P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.

[Wat83]   G. S. Watson. *Statistics on Spheres*. John Wiley and Sons, New York, NY, 1983.

[Zha98]   Zhengyou Zhang. Determining the epipolar geometry and its uncertainty: A review. *IJCV*, 27(2):161–195, 1998.

[ZMI00]   Lihi Zelnik-Manor and Michal Irani. Multi-frame estimation of planar motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1105–1116, October 2000.