To appear in IJCV, merged with "rotation" paper.

Scalable, Absolute Position Recovery for Omni-Directional Image Networks

Matthew Antone Seth Teller
MIT Computer Graphics Group

Abstract

We describe a linear-time algorithm that recovers absolute camera positions for networks of thousands of terrestrial images spanning hundreds of meters, in outdoor urban scenes, under varying lighting conditions. The algorithm requires no human input or interaction. It is robust to up to 80% outliers for synthetic data. For real data, it recovers camera pose which is globally consistent on average to roughly 0.1° and five centimeters, or about four pixels of epipolar alignment, expending a few CPU-hours of computation on a 250MHz processor.

This paper's principal contributions include an extension of Monte Carlo Markov Chain estimation techniques to the case of unknown numbers of feature points, unknown occlusion and deocclusion, and large scale (thousands of images, and hundreds of thousands of point features) and dimensional extent (tens of meters of inter-camera baseline, and hundreds of meters of baseline overall). Also, a principled method is given to manage uncertainty on the sphere of directions; a new use of the Hough Transform is proposed; and a method for aggregating local baseline constraints into a globally consistent constraint set is described.

The algorithm takes intrinsic calibration information, and a connected, rotationally registered image network as input. It then assembles local, purely translational motion estimates into a global constraint set, and determines camera positions with respect to a single scene-wide coordinate system. The algorithm's output is an assignment of metric, accurate 6-DOF camera pose, along with its uncertainty, to every image. We assume that the scene exhibits local point features for probabilistic matching, and that adjacent cameras observe overlapping portions of the scene; no further assumptions are made about scene structure, illumination conditions, or camera motion.

We assess the algorithm's performance on synthetic and real data, and demonstrate several results. First, wide-FOV imagery makes registration fundamentally more robust against failure, and more accurate, than ordinary imagery. Second, we show that by combining thousands of noisy, gradient-based (point) features into a small number of projective motion estimates (baselines), the algorithm achieves accurate registration even in the face of significant lighting variations, low-level feature noise, and errors in initial position estimates.

1 Introduction

Extrinsically 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 visual simulation of realistic scenes. In practice, registering imagery can require substantial manual effort, for example to specify matching tie points across multiple images as constraints to 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.

We have developed two automated camera registration algorithms as part of a system for automated model capture in extended urban environments [52, 53]. In our system, a human operator moves a sensor [11] to many viewing positions in and around the scene of interest. At each position, the sensor acquires a high-resolution, high dynamic range image of some portion of the scene, along with a rough estimate of the acquiring camera's pose, or position and orientation, in absolute (Earth) coordinates. The result is a set of omni-directional images, each with a hemispherical or greater field of view, acquired 15 to 20 meters apart.

Images are grouped by optical center into single, wide-FOV mosaics called "nodes" [18]. Each node is subsequently treated as a rigid, super-hemispherical image with a single pose. The use of wide-FOV imagery provides a significant advantage in practice, by reducing the number of optimization parameters, and by eliminating classical bias and ambiguities in camera motion estimation [26, 18, 22].

The sensor's initial camera pose estimates are not sufficiently accurate for 3D reconstruction, which requires epipolar geometry consistent to a few pixels across any image pair viewing common geometry. Thus one critical component of our system is the refinement of the sensor's initial camera pose estimates to bring all cameras into registration. Since the scale of the dataset rules out interactive techniques, our pose recovery algorithms must be fully automated, and their running times must scale well with the number of images. However, most image pairs observe nothing in common due to occlusion; thus we can not apply algorithmic techniques which assume that common scene structure is observed by all images. Instead, we use the (rough) initial pose estimates to associate cameras which are likely to have observed overlapping scene structure, then use an efficient local-to-global alignment strategy to register all images.

Solving the general pose-recovery problem involves determining six parameters for each camera: three of rotation and three of position. Our approach decouples the 6-DOF problem into a pure rotation (3-DOF) and pure translation (3-DOF) component. This paper addresses only position recovery; a companion paper [3] addresses the prior recovery of scene-relative image orientations.

1.1 Algorithm Overview

The goal of our algorithm is to accurately register every camera (node) to a single, common coordinate system. Our approach uses the fact that nodes are registered rotationally upon input, leaving only node positions to be determined. Also, it exploits the tendency of adjacent (nearby) nodes to have observed overlapping scene structure. The algorithm first detects shared structure across pairs of adjacent nodes, accurately estimating a local displacement direction relating each pair. These local constraints are then propagated throughout the node graph to assign a globally consistent position to each node.

More formally, the algorithm proceeds as follows. Point features are coupled across each node adjacency to give a crude estimate of inter-image baselines. A Monte-Carlo expectation maximization (MCEM) algorithm, based on a projective uncertainty model and initialized with a Hough Transform, simultaneously refines the baseline estimate, deweights outliers, and tests whether low-level match additions or deletions improve the current estimate. All inter-node baseline estimates are assembled into a network-wide constraint set, and a global optimization assigns node positions consistent with all pairwise baselines. Finally, a global rigid transformation is applied to express the node pose in absolute coordinates, maximally consistent with the input position estimates.

1.2 Input Requirements and Assumptions

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

- ***Accurate intrinsic calibration.*** Images have been corrected for radial distortion, and pinhole camera parameters (i.e., focal length, principal point, skew) are given.
- ***Accurate extrinsic orientations.*** Scene-relative orientations and vanishing point directions are supplied for each node [3].
- ***Rough camera locations.*** Absolute (GPS-based) position estimates for each node are supplied by the image acquisition platform.
- ***Camera adjacency.*** For each node, a list of the node's neighbors is given, identifying cameras likely to have viewed overlapping portions of the scene.
- ***Point features.*** For each image, sub-pixel point features, produced by intersecting pairs of gradient-based image edges [12], are supplied.

In practice, the algorithm achieves registration when the following conditions are met:

- ***Node field of view (FOV) is large.*** Our algorithm can be applied to images with any FOV. However, wide-FOV images are fundamentally more powerful observations than conventional images: they provide maximal observations of surrounding structure; disambiguate small rotations from small translations; reduce bias in inference; and in general enables more reliable convergence and higher accuracy.
- ***Nodes view overlapping scene features.*** The dataset has sufficient density that adjacent nodes observe overlapping scene geometry (in this case, 3D points). The inter-node distance in our datasets is typically about fifteen meters.

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 the translation recovery algorithm, and Section 4 reports the result of applying the algorithm to various synthetic and real datasets. Finally, Section 5 reviews previous work on image registration, and Section 6 summarizes our contributions and results.

2 Preliminaries

This section reviews the representations of coordinate transformations and uncertain projective features used by the position recovery algorithm.

2.1 Extrinsic Pose

A rigid transformation, consisting of a 3×1 translation $\boldsymbol{t}$ and orthonormal rotation $\boldsymbol{R}$, expresses points $\boldsymbol{p}^w$ in world space as points $\boldsymbol{p}^c$ in camera space. Its inverse specifies the orientation and position of the camera with respect to the scene coordinate system. Formally,

$$\boldsymbol{p}^c \;=\; \boldsymbol{R}^\top(\boldsymbol{p}^w - \boldsymbol{t}); \qquad \boldsymbol{p}^w \;=\; \boldsymbol{R}\boldsymbol{p}^c + \boldsymbol{t} \tag{1}$$

where $\boldsymbol{t}$ is the position of the focal point, and the columns of $\boldsymbol{R}$ are the principal axes of the camera coordinate system, both expressed in scene coordinates. These two quantities thus summarize the external pose of the camera. The algorithm in this paper assumes that all rotation matrices $\boldsymbol{R}$ (represented as unit quaternions) are known, and addresses the recovery of $\boldsymbol{t}$ for each camera.

2.2 Projective Points

We represent points in the image plane as coordinate pairs (u, v). Although the Euclidean plane is a convenient space for feature detection, it is not ideal for feature representation: it implies non-uniform sampling with respect to the focal point, and can not stably represent rays nearly parallel to the image plane. This leads to instability and poor conditioning in inference tasks, especially when (as in our setting) the field of view is large.

Thus to represent point features we use the *projective plane* $\mathbb{P}^2$, a closed topological manifold containing the set of all 3-D lines through the focal point. Points along any given 3-D line, except the focal point itself, constitute an equivalence class $\sim$:

$$\boldsymbol{p} \sim \boldsymbol{r} \quad \Leftrightarrow \quad \boldsymbol{p} = \alpha \boldsymbol{r}, \tag{2}$$

where α is a real nonzero scalar value. Because of the relationship in (2), the projective plane is a *quotient space* on $\mathbb{R}^3$ (minus the focal point) and also on the surface of the unit sphere $\mathbb{S}^2$, sometimes referred to in the literature as the *Gaussian sphere* [6]. 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.

Points in the Euclidean image plane can be transformed to the sphere by augmentation to homogeneous (projective) coordinates and normalizing:

$$(u, v) \quad \to \quad \boldsymbol{p} = (u, v, 1)^\top \quad \to \quad \frac{\boldsymbol{p}}{\|\boldsymbol{p}\|}. \tag{3}$$

We make use of the following duality between projective points and lines: a given image point $\boldsymbol{p}$ can be viewed as a *pencil* of image lines which contain, and thus intersect at, that point. The parameterizations $\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots$ of such lines must satisfy $\boldsymbol{p} \cdot \boldsymbol{l}_i = 0$. This relationship is depicted in Figure 1; we will return to it in §3.2.1.

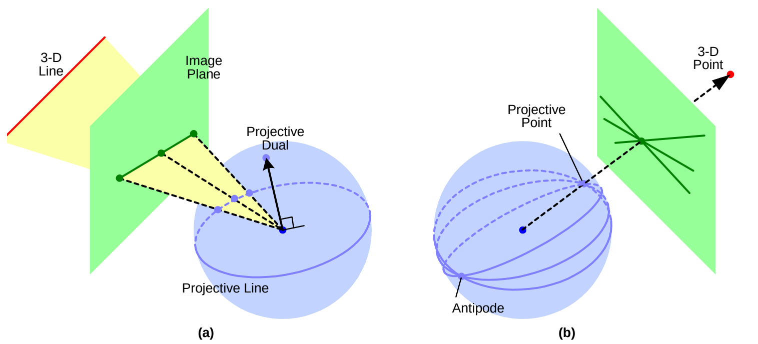

Figure 1: Projective Image Features

(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 representation. (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.

2.3 Bingham's Distribution

Features viewed by a single camera are inherently projective, since no depth information is available. We wish to represent projective features with suitable spherical probability distributions.

Exponential distributions are useful for inference tasks [8], but the most commonly used multi-variate Gaussian density is a Euclidean probability measure and is therefore not suitable for projective variables. Conditioning a zero-mean Gaussian variable $\boldsymbol{x} \in \mathcal{R}^3$) on the event that $\|\boldsymbol{x}\| = 1$ results in *Bingham's distribution*, a flexible exponential density defined on the unit sphere [10, 32, 56].

This distribution can be generalized to arbitrary dimension, and is parameterized by a symmetric $n \times n$ matrix $\boldsymbol{M}$, and diagonalized into the product $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\kappa}\boldsymbol{U}^\top$, 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 κ_i. The density is given by

$$p(\boldsymbol{x}) \;=\; \frac{1}{c(\boldsymbol{\kappa})}\exp(\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{x}) \;=\; \frac{1}{c(\boldsymbol{\kappa})}\exp\left(\sum_{i=1}^{n}\kappa_i(\boldsymbol{u}_i^\top\boldsymbol{x})^2\right) \tag{4}$$

where $c(\boldsymbol{\kappa})$ is a normalizing coefficient that depends only on the concentration parameters. We denote this density by $\mathcal{B}_n(\boldsymbol{x};\boldsymbol{\kappa},\boldsymbol{U})$, or simply $\mathcal{B}_n(\boldsymbol{x};\boldsymbol{M})$, with the subscript n denoting the dimension of the space. The matrix $\boldsymbol{M}$ is analogous to the *information matrix* (inverse of the covariance) of a zero-mean Gaussian distribution [45].

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 $\boldsymbol{y}$ also has a Bingham distribution given by $\mathcal{B}_n(\boldsymbol{y};\boldsymbol{\kappa},\boldsymbol{R}\boldsymbol{U})$. Finally, the Bingham representation is expressive: the concentration parameters can describe a wide variety of distributions, including uniform, bipolar, and equatorial.

We represent projective image features on $\mathbb{S}^2$ as Bingham variables $\mathcal{B}_3(\cdot)$. In addition, since unit quaternions are antipodally symmetric and defined on the surface of the unit hypersphere $\mathbb{S}^3$, we represent rotational uncertainty by the Bingham variables $\mathcal{B}_4(\cdot)$.

3 Position Recovery

Recovery of structure and motion from image information encompasses several coupled problems: camera registration, feature correspondence, and determination of scene structure. In our setting, the input cameras are rotationally registered. This simplifies the epipolar geometry and reduces the dimension of the search space, but the coupling between correspondence and translational pose remains. Our approach is to estimate both correspondence and pose as probability densities, deferring commitment to deterministic values until global information is assembled and propagated throughout the constraint network.

3.1 Overview

This section describes the position recovery algorithm; a high-level diagram of the algorithm is shown in Figure 2 below [4]. First, translation directions are estimated for all node adjacencies in the data set. A Hough transform efficiently finds the most likely motion direction in a given pair by looking for consistency among all possible feature matches. This approximate direction serves to initialize an expectation maximization method whose E-step, which requires sampling from an extremely high-dimensional distribution, relies on a Markov chain Monte-Carlo algorithm. This so-called *MCEM* algorithm, presented in §3.4, determines the best motion direction by averaging over all possible correspondence sets.

Once all relevant pair-wise motion directions have been computed, they are assembled into a global optimization that estimates the camera positions most consistent with these directions (§3.5). A final step, described in §3.6, performs rigid 3-D to 3-D registration on the resulting set of cameras to find the best metric scale, position, and orientation given the approximate initial pose estimates.

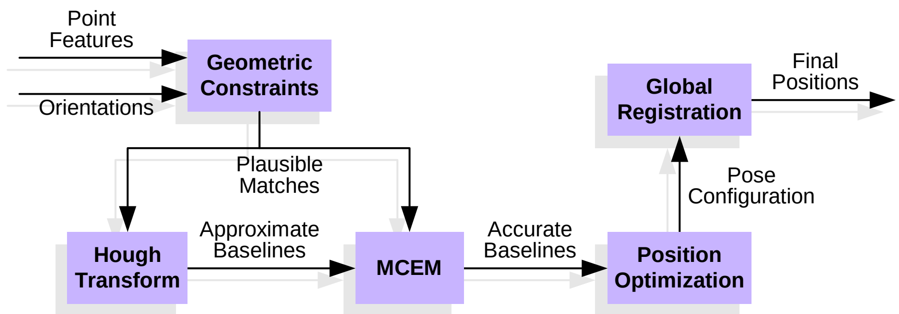

Figure 2: Translational Registration

Determining consistent positions for all cameras without explicit correspondence. First, geometric constraints are imposed to reduce the number of possible point matches. Next, a Hough transform determines approximate baseline directions, which are then refined probabilistically. A global optimization, constrained by the pair-wise directions, produces consistent camera positions. Finally, the cameras are registered with the initial pose to recover metric scale, orientation, and position.

3.2 Two-Camera Translation Geometry

Given two rotationally registered cameras $\mathcal{A}$ and $\mathcal{B}$, and two sets of respective features $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ and $\mathcal{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M\}$, the goal is to determine the direction of motion $\boldsymbol{b}$ from $\mathcal{A}$ to $\mathcal{B}$ most consistent with the available data. This section describes the simplified epipolar geometry resulting from known rotational pose and discusses geometric constraints that may be used to reject physically unlikely point matches. Given a set of explicit matches within this framework, estimation of the direction of translation between a given pair of cameras reduces to a projective inference problem quite similar to that of single vanishing point estimation.

3.2.1 Epipolar Geometry with Known Orientation

An epipolar plane $\mathcal{P}$ contains two camera centers and a 3-D point seen by both cameras. Projections of the 3-D point onto each of the images, $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ respectively, must therefore also lie in $\mathcal{P}$ (see Figure 3).

For rotationally registered cameras, the following relation holds:

$$(\boldsymbol{x} \times \boldsymbol{y}) \cdot \boldsymbol{b} = 0. \tag{5}$$

Intuitively, the cross product of $\boldsymbol{x}$ with $\boldsymbol{y}$ is orthogonal to $\mathcal{P}$, and thus necessarily orthogonal to the baseline $\boldsymbol{b}$ as well, since $\mathcal{P}$ contains $\boldsymbol{b}$. Here, observations consist only of the 2-D feature projections, and the baseline is unknown; however, (5) provides a constraint on $\boldsymbol{b}$ up to unknown scale. This suggests that $\boldsymbol{b}$ can be inferred solely from two or more corresponding pairs of features.

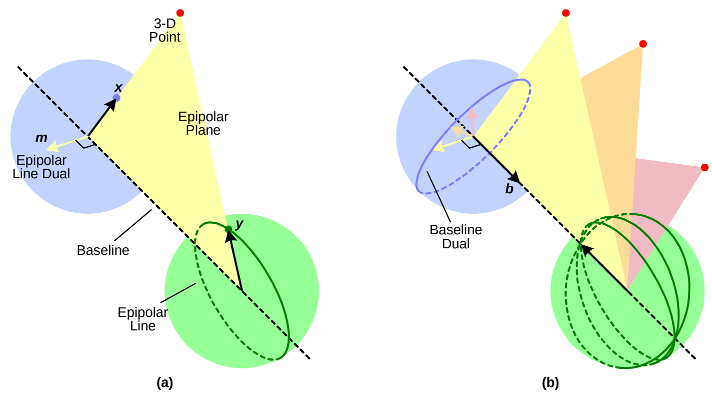

Figure 3: Pair Translation Geometry

The epipolar geometry for two rotationally aligned cameras is similar to the geometry of vanishing points. (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.

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 feature $\boldsymbol{x}_i$ truly matches feature $\boldsymbol{y}_j$—the constraint in (5) becomes

$$\boldsymbol{m}_{ij} \cdot \boldsymbol{b} = 0. \tag{6}$$

If the $\boldsymbol{m}_{ij}$ are viewed as projective epipolar lines, then the baseline $\boldsymbol{b}$ can be viewed as a projective *focus of expansion*, and its antipode the *focus of contraction*, the apparent intersections of all epipolar lines.

3.2.2 Geometric Constraints on Correspondence

Both correspondence and the baseline are initially unknown, so the above construction seems hopelessly underconstrained. There are NM possible individual feature matches, and more importantly, a combinatorial number of possible correspondence *sets* that can be chosen, making the search space enormous.

However, additional information can be used to drastically lower the dimension of the search space [43] both by reducing the number of features N and M in each image, and by eliminating many of the candidate correspondences. The constraints presented here rely on two assumptions: first, that each point feature represents the intersection of two or more 2-D line features; and second, that the baseline is known to lie within some restricted region. Bounds on this region can be obtained using the roughly-known initial pose.

Knowledge of 3-D line directions and classification of 2-D line features obtained from rotational pose recovery both provide strong cues for feature culling and point correspondence rejection. Presumably, objects consisting of parallel lines possess sufficient structure for determination of translational offsets; thus, image features not associated with any parallel line sets can be safely discarded. In particular, lines having high “outlier” probability, along with any point features inferred by these lines, are deemed invalid. Points inferred from lines shorter than a given threshold in Euclidean image space can also be discarded, as such lines are unreliable.

A set of all possible candidate matches is constructed from the surviving sets of point features. Each match is kept or discarded according to the following criteria:

- ***Directions of constituent lines.*** If the 3-D point inferred by a given match truly corresponds to the intersection of two or more 3-D lines, then the 3-D directions of image lines forming a given image point $\boldsymbol{x}_i$ should be identical to those forming the point $\boldsymbol{y}_j$ (Figure 4). Matches $\boldsymbol{m}_{ij}$ for which this condition does not hold are discarded.

- ***Baseline uncertainty bound.*** A given angular bound on the translation direction induces a conservative equatorial band within which all correct epipolar plane normals must lie (Figure 5); any $\boldsymbol{m}_{ij}$ outside this band is discarded, since it implies "sideways" motion. Furthermore, any match for which $\boldsymbol{y}_j$ is closer than $\boldsymbol{x}_i$ to $\boldsymbol{b}$ is also discarded, as such a match implies "backward" motion.

- ***Depth of 3-D point.*** If the angle between $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ exceeds a threshold, the 3-D point inferred by the match (via triangulation) is too close to the camera or implies an abnormally wide baseline. These matches are therefore discarded.

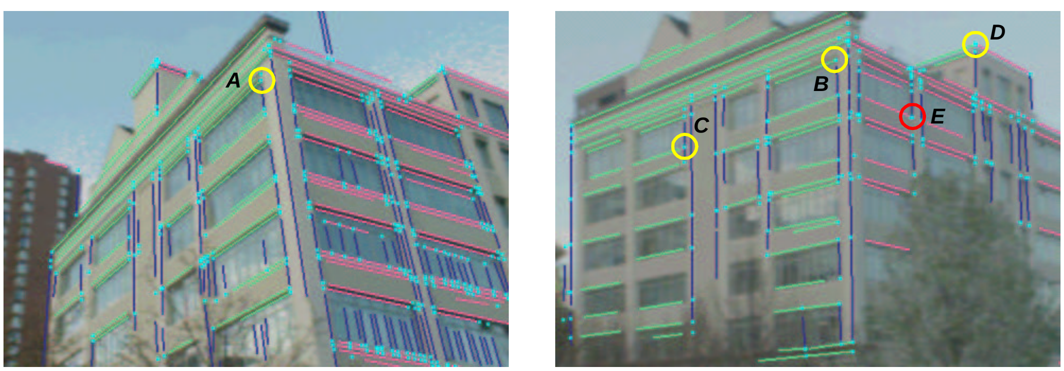

Figure 4: Line Constraints

Two images viewing the same building are shown, and possible matches for a particular point feature A in the first image are considered. 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. The directions of the lines forming E do not match those forming A, so E is rejected. Note that D is formed by the intersection of three rather than two distinct line directions.

3.3 Inference of Translation Direction

This section describes methods for inferring the translation direction between a pair of cameras, 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-camera baseline $\boldsymbol{b}$ according to (5), and a set of such correspondences can be used to estimate $\boldsymbol{b}$.

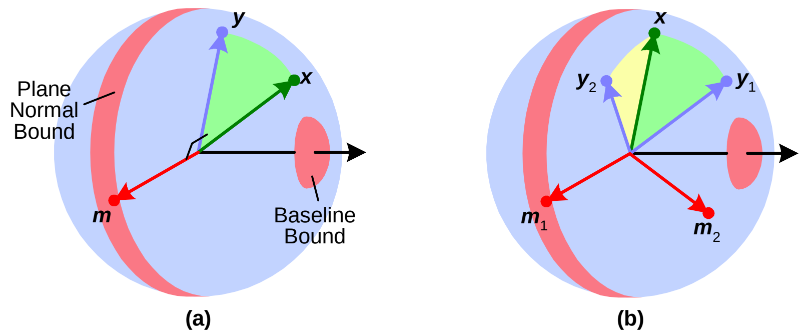

Figure 5: Direction Constraints

(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 it implies backward motion; the match with $\boldsymbol{y}_2$ is also rejected because its epipolar line does not lie in the uncertainty band.

One method is by minimization of an objective function such as

$$E = \sum_{(i,j)\in\mathcal{F}} \boldsymbol{m}_{ij} \cdot \boldsymbol{b}; \tag{7}$$

here, $\mathcal{F}$ is the set of F pairings (i, j) that represent the true matches. The optimal least-squares baseline direction can be found by constructing an $F \times 3$ matrix $\boldsymbol{A}$ whose rows contain the feature cross products $\boldsymbol{m}_{ij}$, then choosing the eigenvector associated with the smallest eigenvalue of $\boldsymbol{A}^\top \boldsymbol{A}$.

Projective fusion techniques can be used to estimate the probability density of $\boldsymbol{b}$. Recall from §1.2 that each point feature in the image 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 the parameters of each image point's distribution are 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 now remains is to determine the distribution of $\boldsymbol{b}$ given a set of epipolar line observations $\boldsymbol{m}_{ij}$ (for $(i, j) \in \mathcal{F}$) with known uncertainty.

3.3.1 Motion Direction from Known Correspondence

If true correspondences between the feature sets $\mathcal{X}$ and $\mathcal{Y}$ are known, the parameters $\boldsymbol{M}_{\boldsymbol{b}}$ of the baseline distribution can be inferred according to the fusion equation

$$\boldsymbol{M}_{\boldsymbol{b}} = \sum_{(i,j)\in\mathcal{F}} \boldsymbol{M}_{ij} + \boldsymbol{M}_0 \tag{8}$$

where $\boldsymbol{M}_{ij}$ represents the uncertainty of the epipolar line $\boldsymbol{m}_{ij}$, $\boldsymbol{M}_0$ is the prior distribution on $\boldsymbol{b}$, and the sum is taken only over indices associated with the true matches. Equivalently, inference can be performed by associating a binary-valued variable b_{ij} with *every possible* correspondence, where

$$b_{ij} = \begin{cases} 1, & \text{if } \boldsymbol{x}_i \text{ matches } \boldsymbol{y}_j \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

The Bingham parameters of $\boldsymbol{b}$ can then be determined by

$$\boldsymbol{M}_{\boldsymbol{b}} = \sum_{i=1}^{M} \sum_{j=1}^{N} b_{ij} \boldsymbol{M}_{ij} + \boldsymbol{M}_0, \tag{10}$$

where the new sum is evaluated over every possible (i, j) pairing.

3.3.2 Motion Direction from Probabilistic Correspondence

Because motion directions and point features are uncertain quantities, and because of ambiguities in epipolar geometry that may arise from particular viewpoints, *hard* or *explicit* correspondence cannot always be determined. Thus, in the more general case, continuously-valued variables $w_{ij} \in [0, 1]$, rather than binary-valued 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 weighted formulation becomes

$$\boldsymbol{M}_{\boldsymbol{b}} = \sum_{i=1}^{M} \sum_{j=1}^{N} w_{ij} \boldsymbol{M}_{ij} + \boldsymbol{M}_0, \tag{11}$$

with more emphasis given to matches with higher likelihood. Note that the binary variables b_{ij} represent the deterministic limit of the w_{ij} in this probabilitic formulation.

3.3.3 Feature Match Weights

In reality, each feature observed in one image has at most one true match in the other image. A true match exists only if the feature observation corresponds to a real 3-D point, and if its counterpart in the other image is visible; otherwise, the feature has *no* matches—either it is itself spurious, or its match is unavailable (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,

$$\sum_{j=1}^{N} b_{ij} \leq 1 \quad \forall i \qquad \sum_{i=1}^{M} b_{ij} \leq 1 \quad \forall j. \tag{12}$$

Inequality constraints are mathematically inconvenient, however; 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 (12) become equality constraints via the introduction of binary-valued *slack variables* b_{i0} and b_{0j} [16], which take value one if $\boldsymbol{x}_i$ (or $\boldsymbol{y}_j$, respectively) matches no other feature, and zero otherwise. Thus,

$$\sum_{j=0}^{N} b_{ij} = 1 \quad 1 \le i \le M \qquad \sum_{i=0}^{M} b_{ij} = 1 \quad 1 \le j \le N. \tag{13}$$

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

$$\sum_{j=0}^{N} w_{ij} = 1 \quad 1 \le i \le M \qquad \sum_{i=0}^{M} w_{ij} = 1 \quad 1 \le j \le N. \tag{14}$$

This condition enforces a symmetric (two-way) distribution over all correspondences: each feature in the first image can match a set of possible features in the second image, 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 (Figure 6). The condition in (14) 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.

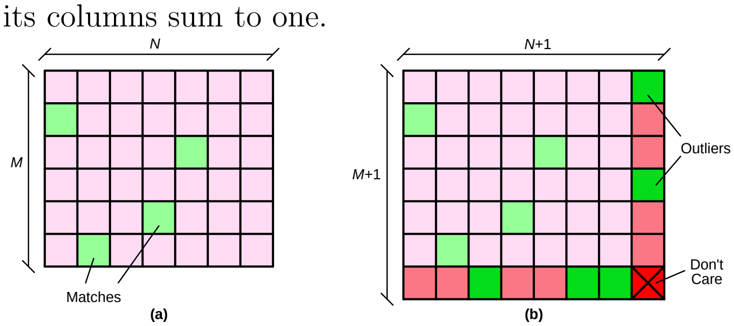

Figure 6: Augmented Match Matrix

The match matrix encodes correspondences between features in two different images. (a) An example of a binary match matrix. Rows represent features in the first camera, and columns represent features in the second. There can be at most one non-zero entry per row and per column. (b) The augmented matrix, with an extra row and column to account for outliers and missing features. In the augmented matrix, there must be exactly one non-zero entry in each of the first M rows and N columns.

3.3.4 Initialization: Obtaining a Prior Distribution

Because motion direction and correspondence are tightly coupled, it is difficult to determine these quantities without prior information. However, as this section will demonstrate, utilization of initial pose estimates and the geometric constraints introduced in §3.2.2 allows reasonably accurate estimates of $\boldsymbol{b}$ to be obtained *without* knowledge of correspondence.

Let $\mathcal{M}$ represent the set of *all* plausible correspondences (epipolar lines) between $\mathcal{X}$ and $\mathcal{Y}$, and let the special 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 perfectly at the motion direction $\boldsymbol{b}$, and the remainder (which represent false matches) will intersect at random points on the sphere.

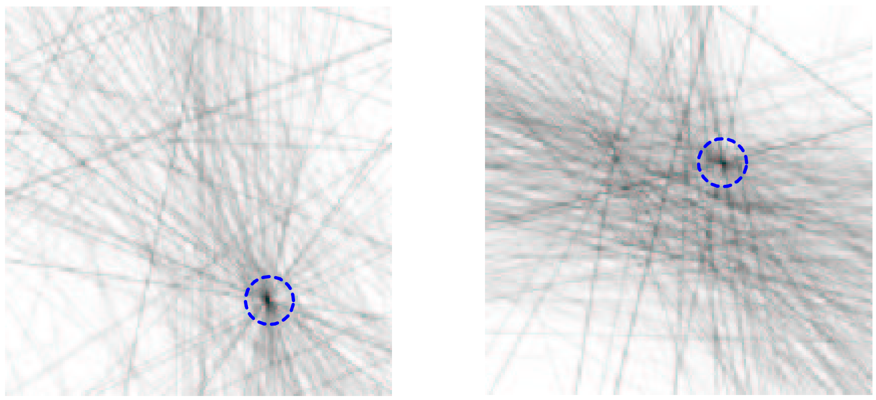

Figure 7: Hough Transform for Baseline Estimation

Two examples of Hough transforms for baseline estimation. Epipolar lines for all plausible matches are accumulated; the transform peak represents the baseline direction.

In essence, the point of maximum incidence on $\mathbb{S}^2$ is the most likely direction of motion. This point can be found by discretizing $\mathbb{S}^2$ and accumulating all candidate epipolar lines $\mathcal{M}$ in a Hough transform. Since the motion direction is approximately known from initial pose estimates, the transform need only be formulated over a small portion of the sphere's surface around this initial direction. Examples are shown in Figure 7.

The motion direction $\boldsymbol{b}_0$ can be determined as the peak in the transform with highest magnitude. False correspondences greatly 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 arising from certain geometric anomalies. For example, a point feature in one image lying very close to the initial direction of motion can match many features in the other image, thus producing a perfectly sharp false peak if all matches are equally weighted (Figure 8).

To solve this problem, a mutually consistent set of weights w_{ij} must be assigned to the epipolar lines in $\mathcal{M}$ such that features having many possible matches are de-emphasized. In order to ensure that the condition in (14) is satisfied, an iterative normalization procedure proposed by Sinkhorn [47, 16] is utilized to transform an 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 §3.2.2, as well as all entries in row $M + 1$ and column $N + 1$, are then assigned an initial value of one. Sinkhorn's algorithm alternatively normalizes the rows and columns until

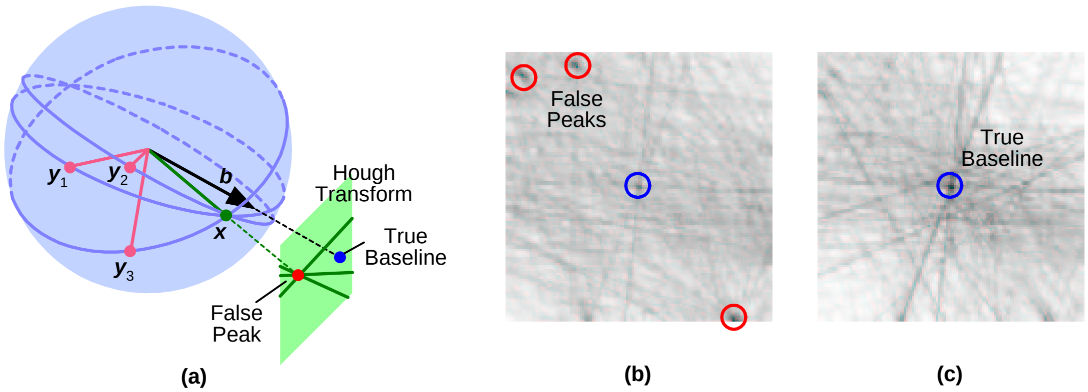

Figure 8: False Hough Transform Peaks

(a) False peaks in the Hough transform can be caused by features too close to the direction of motion, 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.

convergence as follows:

$$w'_{ij} = w_{ij} \Bigg/ \sum_{j=0}^{N} w_{ij} \ \ \forall i \in \{1, \ldots, M\}; \qquad w''_{ij} \ = \ w'_{ij} \Bigg/ \sum_{i=1}^{M} w'_{ij} \ \ \forall j \in \{1, \ldots, N\}.$$

Each entry in the matrix is normalized by the sum of entries in its row; each entry in the resulting matrix is then normalized by the sum of entries in its column, and so on. The algorithm produces a provably *unique* factorization $\boldsymbol{W}' = \boldsymbol{D}_1 \boldsymbol{W} \boldsymbol{D}_2$ [47], such that $\boldsymbol{W}'$ is doubly stochastic. 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 technique described above.

For a planar accumulation space, each linear constraint of the form in (5) contributes a single straight line to the transform. Thus, once a set of weights has been obtained by the above method, the epipolar lines are accumulated, and when drawn, their accumulation values are 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 (Figure 8c).

Although accuracy is inherently limited by the discrete nature of the Hough transform, the resulting motion direction estimate $\boldsymbol{b}_0$ can be used to initialize more accurate techniques. Further, it can be used as a strong prior distribution (with parameters $\boldsymbol{M}_0$ in the notation of §3.3) in subsequent inference tasks. The matrix $\boldsymbol{M}_0$ is obtained by using a bipolar scatter matrix approximation on the region surrounding the peak.

3.4 Monte Carlo Expectation Maximization

In general, true feature correspondence is completely unknown; feature point measurements and uncertainty serve as the only available information for inference of motion. This section outlines a method for determining accurate motion estimates from this information alone, *without* requiring explicit correspondence, by employing an expectation algorithm in which the posterior distribution is discretely sampled.

Using maximum likelihood notation,

$$\boldsymbol{b}^* = \mathrm{argmax}_{\boldsymbol{b}}[p(\boldsymbol{b}|\mathcal{M})] \tag{15}$$

The conditional probability above can be expanded using Bayes' rule:

$$p(\boldsymbol{b}|\mathcal{M}) \;=\; \sum_{\boldsymbol{B}} p(\boldsymbol{b}, \boldsymbol{B}|\mathcal{M}) \;=\; \sum_{\boldsymbol{B}} p(\boldsymbol{b}|\boldsymbol{B}, \mathcal{M}) p(\boldsymbol{B}|\mathcal{M}) \tag{16}$$

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 is assumed to be uniform, but can equally well incorporate the geometric match constraints of §3.2.2. Note also that the likelihood is expressed as a summation rather than an integration, because the collection of all possible correspondence sets is discrete.

3.4.1 Structure from Motion without Correspondence

The expression in (15) 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 [21]. Correspondence sets can be treated as nuisance parameters in a Bayesian formulation, as illustrated by (16), in which the likelihood is evaluated over all possible matrices $\boldsymbol{B}$.

The expectation maximization algorithm lends itself well to this type of optimization problem. The algorithm alternates between the M-step, in which a log likelihood function is maximized given a posterior likelihood, and the E-step, in which the likelihood function is evaluated given the current parameter estimate $\boldsymbol{b}$. Convergence to the optimal solution is guaranteed because of the initial estimate provided by the Hough transform approach above.

The log likelihood to be maximized is

$$L = \sum_{\boldsymbol{B}} p(\boldsymbol{B}|\boldsymbol{b}, \mathcal{M}) \log p(\boldsymbol{b}|\boldsymbol{B}, \mathcal{M}). \tag{17}$$

Substitution of (10) into (17) gives

$$L \propto \sum_{\boldsymbol{B}} p(\boldsymbol{B}|\boldsymbol{b}, \mathcal{M}) \sum_{i=1}^{M} \sum_{j=1}^{N} b_{ij} \boldsymbol{b}^\top \boldsymbol{M}_{ij} \boldsymbol{b} + \boldsymbol{b}^\top \boldsymbol{M}_0 \boldsymbol{b} \tag{18}$$

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

$$w_{ij} \equiv p(b_{ij} = 1|\boldsymbol{b}, \mathcal{M}) = \sum_{\boldsymbol{B}} \delta(i, j) p(\boldsymbol{B}|\boldsymbol{b}, \mathcal{M}); \tag{19}$$

then (18) becomes

$$\sum_{i=1}^{M}\sum_{j=1}^{N} w_{ij}\boldsymbol{b}^\top \boldsymbol{M}_{ij}\boldsymbol{b} + \boldsymbol{b}^\top \boldsymbol{M}_0\boldsymbol{b}. \tag{20}$$

Maximization of (20), given the set of weights w_{ij}, can be easily performed using the technique described in §3.3.2; however, determination of the w_{ij} is not so straightforward. Individual matches are not mutually independent, because knowledge about one match provides knowledge about others. For example, given that $b_{ij} = 1$, it must be true that

$$b_{ik} = 0 \qquad \forall k \neq j, \tag{21}$$

which follows from the implicit constraints in (13). The condition for independence is thus violated, because

$$p(b_{ik} = 1 | b_{ij} = 1, \boldsymbol{b}, \mathcal{M}) = 0 \neq p(b_{ik} = 1 | \boldsymbol{b}, \mathcal{M}), \tag{22}$$

and the joint likelihood $p(\boldsymbol{B}|\boldsymbol{b}, \mathcal{M})$ cannot be factored. Precise evaluation of (20) apparently requires evaluation of (18), a difficult task due to the combinatorial number of terms. However, the following sections demonstrate that the w_{ij} can be evaluated efficiently by Monte Carlo sampling.

3.4.2 Sampling the Posterior Distribution

Markov chain Monte Carlo (MCMC) algorithms are useful for evaluating sums of the form in (18). In this context, each possible binary match matrix $\boldsymbol{B}^k$ represents a distinct *state*; random transitions between states occur until transitions equalize and *steady state* is reached. If the transition likelihoods are appropriately chosen, then the steady-state probabilities represent the distribution on correspondence matrices $\boldsymbol{B}$.

Our approach combines Metropolis sampling [38], which ensures appropriate transition probabilities, with simulated annealing [34], which allows relative likelihood maxima to be avoided by visiting a larger portion of the sample space. The approach can be summarized as follows:

Start with initial temperature $T = T_0$
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 $\boldsymbol{A} = \mathbf{0}$
    Loop until k sufficiently high (steady state):
        Randomly perturb state to $\tilde{\boldsymbol{B}}^k$
        Evaluate the likelihood ratio β

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 $\boldsymbol{W} = \boldsymbol{A}/k$
Solve for new $\boldsymbol{b}$ given $\boldsymbol{W}$ (M-step)
Set $T = \alpha T$ (for $0 < \alpha < 1$)
Set $n = n + 1$

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

$$p(\boldsymbol{B}^k|\boldsymbol{b}, \mathcal{M}) \;=\; c(\boldsymbol{M}^k)\exp\left[\boldsymbol{b}^\top \boldsymbol{M}^k \boldsymbol{b}\right] \;=\; c(\boldsymbol{M}^k)\exp\left[\boldsymbol{b}^\top \sum_{i=1}^{M}\sum_{j=1}^{N} b_{ij}^k \boldsymbol{M}_{ij}\boldsymbol{b}\right] \qquad (23)$$

where $\boldsymbol{b}$ is taken as the polar direction of the current baseline distribution estimate. Efficient calculation of β is described in §3.4.4.

In a particular E-step loop, $\boldsymbol{A}$ is an $(M+1)\times(N+1)$ accumulation matrix that counts the number of 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 serve to 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 according to uncertainty bounds on $\boldsymbol{b}_0$, 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 effectively 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 in the following sections.

3.4.3 Match Perturbations

For the case where $\boldsymbol{B}^k$ is a square permutation matrix (i.e. all features are visible in all images), Dellaert proposes simple swap perturbations, so that $\boldsymbol{B}^{k+1}$ is identical to $\boldsymbol{B}^k$ except for a single row (or, equivalently, column) swap. It can be proven that all states are reachable using these perturbations. When the number of visible 3-D features is unknown, however, and when outliers and occlusion are present, detailed balance is no longer satisfied by simple match swapping, because such swapping preserves the number of valid matches; therefore, states with greater or fewer matches than the current state are never reached.

We generalize Dellaert's technique, in the two-camera case, to handle an unknown number of visible 3-D features, and also to handle outliers and occlusion. The state matrix $\boldsymbol{B}$ and the probability matrix $\boldsymbol{W}$ are each augmented with an extra row and column (§3.3.3) to represent an appropriate state space (i.e. to account for features having no matches). Novel perturbations in addition to row and columns swaps are also introduced which allow all states to be visited.

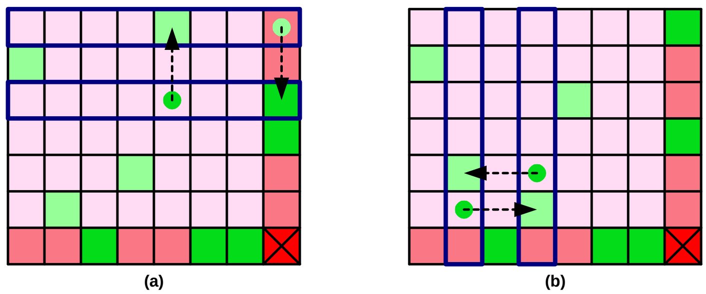

Figure 9: Row and Column Swaps

(a) Two rows of the match matrix, including outliers, are interchanged. (b) Two columns are interchanged.

In particular, to allow the number of valid matches to change, two complementary operations are proposed. The *split* perturbation converts a valid match into two outlier features, and the *merge* perturbation joins two outlier features into one valid match. Figure 10 depicts these operations in terms of the correspondence matrix $\boldsymbol{B}$.

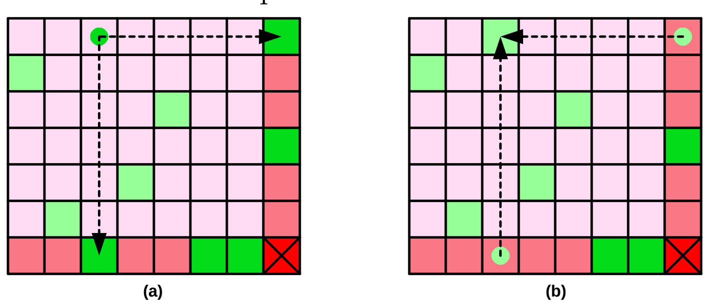

Figure 10: Split and Merge Perturbations

(a) A valid correspondence is split into two outliers, thus reducing the number of valid matches by one. (b) Any two outliers can be merged into a valid correspondence to increase the number of valid matches by one.

3.4.4 Efficient Sampling

The sampling algorithm outlined above seems at first to be computationally expensive, especially for the large state matrices typical of real images containing many features. However, three optimizations can be applied to significantly improve the algorithm's performance. The majority of any given state matrix is zero; in fact, out of MN possible entries, a maximum of $N + M - 1$ 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 §3.2.2, 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 β. Each perturbation represents only an incremental change in the state that involves at most four entries in $\boldsymbol{B}$. The exponential form of the likelihood function in (23) facilitates computation of ratios:

$$\beta \;=\; \frac{p(\tilde{\boldsymbol{B}}^k|\boldsymbol{b},\mathcal{M})}{p(\boldsymbol{B}^k|\boldsymbol{b},\mathcal{M})} \;=\; \frac{c(\tilde{\boldsymbol{M}}^k)\exp\left[\boldsymbol{b}^\top\tilde{\boldsymbol{M}}^k\boldsymbol{b}\right]}{c(\boldsymbol{M}^k)\exp\left[\boldsymbol{b}^\top\boldsymbol{M}^k\boldsymbol{b}\right]} \;=\; \frac{c(\tilde{\boldsymbol{M}}^k)}{c(\boldsymbol{M}^k)}\exp\left[\boldsymbol{b}^\top(\tilde{\boldsymbol{M}}^k-\boldsymbol{M}^k)\boldsymbol{b}\right]. \quad (24)$$

In the case of swapping two rows, say row m which contains a one in column n and row p which contains a one in column q, most terms in the sum of (23) remain unchanged; only the entries b_{mn}, b_{pq}, b_{mq}, and b_{pn} differ. The new parameter matrix is given by

$$\tilde{\boldsymbol{M}}^k = \boldsymbol{M}^k + \boldsymbol{M}_{mq} + \boldsymbol{M}_{pn} - \boldsymbol{M}_{mn} - \boldsymbol{M}_{pq}, \quad (25)$$

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, then the new parameter matrix becomes

$$\tilde{\boldsymbol{M}}^k = \boldsymbol{M}^k - \boldsymbol{M}_{mn}; \quad (26)$$

if two outliers are merged, the new parameter matrix is

$$\tilde{\boldsymbol{M}}^k = \boldsymbol{M}^k + \boldsymbol{M}_{mn}. \quad (27)$$

Incremental computation of the difference $(\tilde{\boldsymbol{M}}^k - \boldsymbol{M}^k)$ in (24) is thus straightforward.

3.5 Multi-Camera Method

Translations recovered between camera pairs are merely directions, and thus can only be determined up to unknown scale. This section illustrates how baseline directions can be assembled into a set of constraints on camera positions and used to recover a globally consistent pose configuration.

3.5.1 Baseline Constraints

Only the *directions* (not distances) between adjacent nodes and rough initial camera positions are known. We employ an iterative algorithm that updates each node's position $\boldsymbol{p}_i$ using constraints imposed by its associated baselines.

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 estimated baselines $\boldsymbol{b}_{ji}$ (Figure

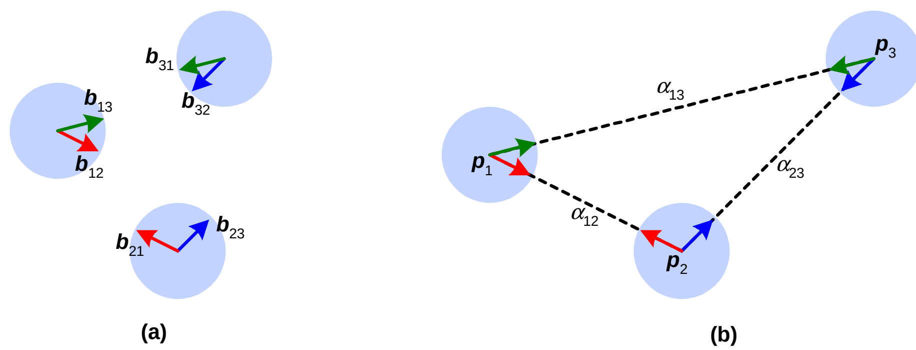

Figure 11: Assembling Translation Directions

(a) After motion directions are estimated between all relevant camera pairs, camera positions are still unknown. (b) A pose configuration consistent with all motion directions can be determined.

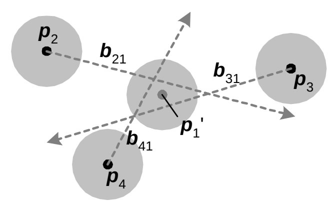

Figure 12: Single Node Baseline Constraints

A node's position is constrained by adjacent positions and baselines.

12). The new position $\boldsymbol{p}_i'$ for node i is chosen to minimize the mean-square distance to each baseline ray. In the absence of baseline uncertainty, $\boldsymbol{p}_i'$ can be determined according to

$$\boldsymbol{p}_i' = \left(\sum\nolimits_j (\boldsymbol{I} - \boldsymbol{b}_{ji}\boldsymbol{b}_{ji}^\top)\right)^{-1} \left(\sum\nolimits_j (\boldsymbol{I} - \boldsymbol{b}_{ji}\boldsymbol{b}_{ji}^\top)\boldsymbol{p}_j\right). \tag{28}$$

Uncertainty in baseline directions can be incorporated by replacing $\boldsymbol{b}_{ji}\boldsymbol{b}_{ji}^\top$ in (28) with the second-moment matrix of the baseline's Bingham density. Uncertainty in $\boldsymbol{p}_i'$, in the form of a 3×3 Euclidean covariance matrix, is approximated by the inverse matrix in (28).

3.6 Metric Registration

Pose estimates recovered using the methods of §3.5 are globally consistent relative to each other. However, they reside in a locally defined and somewhat arbitrary coordinate system that does not necessarily correspond to the metric space of the scene. A rigid transformation consisting of translation, rotation, and scale can express camera pose with respect to any desired coordinate system while preserving the local relationships among cameras.

The sensor produces pose estimates in an absolute (Earth-relative) coordinates [11]. These estimates provide a ground-truth reference frame to which the camera configuration is finally registered. We assume that the sensor estimates are unbiased, so that the Euclidean transformation that best fits recovered camera positions to initial camera positions produces an optimal pose assignment.

3.6.1 Absolute Orientation

This section outlines the approach to absolute orientation, or 3-D to 3-D registration, as proposed by Horn [30]. The goal is to find the translation, rotation, and scale that best align the N recovered camera positions, or source points $\boldsymbol{x}_i$, with the N initial positions, or target points $\boldsymbol{y}_i$.

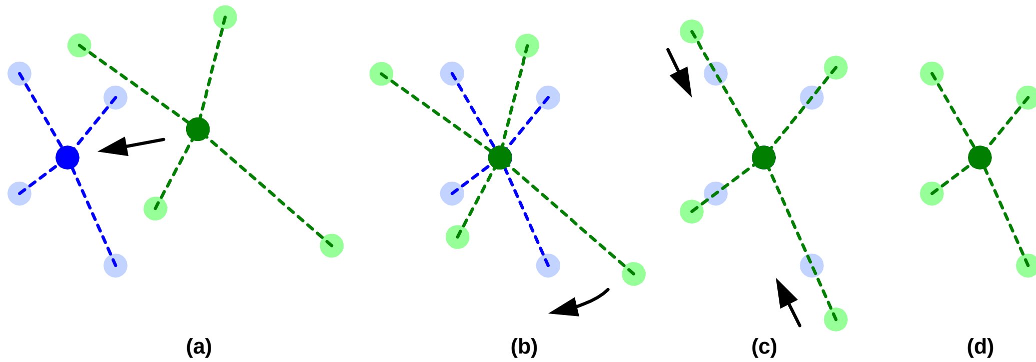

Figure 13: Metric Registration Process

A two-dimensional depiction of metric registration. (a) The original configuration is shifted so that the two centroids coincide. (b) Rays from the centroid to each camera 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. A new set of points is thus defined so that

$$\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \boldsymbol{x}_0, \qquad \tilde{\boldsymbol{y}}_i = \boldsymbol{y}_i - \boldsymbol{y}_0 \tag{29}$$

where

$$\boldsymbol{x}_0 = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i, \qquad \boldsymbol{y}_0 = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{y}_i. \tag{30}$$

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 $\boldsymbol{y}_0$, respectively, as shown in Figure 13. The rotation $\boldsymbol{R}$ is estimated using the deterministic two-camera rotation method described in [3]. Next, the optimal scale factor s is computed as

$$s = \sqrt{\frac{\sum_{i=1}^{N}\tilde{\boldsymbol{y}}_i \cdot \tilde{\boldsymbol{y}}_i}{\sum_{i=1}^{N}\tilde{\boldsymbol{x}}_i \cdot \tilde{\boldsymbol{x}}_i}}. \tag{31}$$

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

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{x}_i) &= s\boldsymbol{R}(\boldsymbol{x}_i - \boldsymbol{x}_0) + \boldsymbol{y}_0 \\ &= s\boldsymbol{R}\boldsymbol{x}_i + \boldsymbol{t} \end{aligned} \tag{32}$$

where $\boldsymbol{t} = \boldsymbol{y}_0 - s\boldsymbol{R}\boldsymbol{x}_0$. This is consistent with the derivation in [30].

Probabilistic transformations are not necessary here because the target positions are not ground-truth quantities. However, the previously estimated pose uncertainty must undergo a similar set of transformations, described in the next section.

3.6.2 Transforming Uncertainty

Modification of the camera pose necessitates appropriate modification of its uncertainty. Let $\boldsymbol{x}$ be the position of a given camera before metric registration, with uncertainty described by a Gaussian random variable with mean at $\boldsymbol{x}$ and with covariance matrix $\boldsymbol{\Lambda}_{\boldsymbol{x}}$, and let $\boldsymbol{y}$ represent the camera's position after registration. From (32),

$$\boldsymbol{y} = s\boldsymbol{R}\boldsymbol{x} + \boldsymbol{t}. \tag{33}$$

The new covariance is then given by

$$\begin{aligned}
\boldsymbol{\Lambda}_{\boldsymbol{y}} &= \langle \boldsymbol{y}\boldsymbol{y}^\top \rangle - \langle \boldsymbol{y} \rangle \langle \boldsymbol{y}^\top \rangle \\
&= \langle (s\boldsymbol{R}\boldsymbol{x} + \boldsymbol{t})(s\boldsymbol{x}^\top \boldsymbol{R}^\top + \boldsymbol{t}^\top) \rangle - (s\boldsymbol{R}\langle \boldsymbol{x} \rangle + \boldsymbol{t})(s\langle \boldsymbol{x}^\top \rangle \boldsymbol{R}^\top + \boldsymbol{t}^\top) \\
&= s^2 \boldsymbol{R} \langle \boldsymbol{x}\boldsymbol{x}^\top \rangle \boldsymbol{R}^\top - s^2 \boldsymbol{R} \langle \boldsymbol{x} \rangle \langle \boldsymbol{x}^\top \rangle \boldsymbol{R}^\top \\
&= s^2 \boldsymbol{R} \boldsymbol{\Lambda}_{\boldsymbol{x}} \boldsymbol{R}^\top
\end{aligned} \tag{34}$$

and is therefore independent of the translation $\boldsymbol{t}$.

Camera orientation is not affected by pure translation or scale; thus, rotational pose uncertainty is altered only by the rotation $\boldsymbol{R}$. A given camera's orientation is represented 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 a distinct quaternion, should be transformed by $\boldsymbol{R}$. A quaternion acts on another quaternion as a matrix multiplication; thus, the new orientation quaternion $\tilde{\boldsymbol{q}}$ is then given by $\tilde{\boldsymbol{q}} = \boldsymbol{Q}\boldsymbol{q}$, where $\boldsymbol{Q}$ is a 4×4 matrix representing $\boldsymbol{R}$. The same matrix can be used to transform the columns of $\boldsymbol{U}$, resulting in a new random variable distributed as $\mathcal{B}_4(\tilde{\boldsymbol{q}}; \boldsymbol{\kappa}, \boldsymbol{Q}\boldsymbol{U})$.

3.7 Asymptotic Running Time

The running time of the translational registration algorithm is $O(n)$, or linear in the number of input nodes n. Note that the number of adjacencies in the node graph is linear in n, since every node has at most a constant number neighbors in the graph. The algorithm's pairwise baseline estimation stage therefore requires $O(n)$ time per iteration. In practice a few dozen iterations suffice for convergence.

3.8 Limitations

The algorithm has several limitations. It requires useable point features from a feature detector. It relies on pairwise baseline estimates, so can be unstable for degenerate input configurations, or incorrect node adjacencies. The algorithm's assumption that nearby nodes are likely to have observed overlapping scene structure may be faulty, for example when two nodes lie on opposite sides of a thin building.

3.9 Summary of Position Recovery

This section presented a sequence of steps for the recovery of metrically aligned camera positions given known orientations and scene-relative 3-D line directions. First, camera adjacencies are determined from the approximately known initial pose. For each adjacent camera pair, a direction of motion is determined. Geometric constraints are imposed which drastically reduce the number of putative point feature matches, after which a Hough transform determines the most likely motion direction by considering all matches simultaneously. A MCEM algorithm then alternately refines the motion direction and computes probabilistic correspondence by sampling over all correspondence sets.

All two-camera motion directions are assembled into a set of linear constraints on camera positions, which is iteratively solved to produce a globally consistent pose configuration. Finally, this configuration is rigidly transformed for metric alignment with the original camera pose using a classical 3-D to 3-D alignment technique. The end result is a set of consistent camera positions and orientations, as well as estimates of their uncertainty.

4 Experiments

We implemented the position registration algorithm in roughly 5,000 lines of C++ code, and instrumented its performance on a 250MHz SGI O2 with 1.5 Gigabytes of memory. This section assesses the algorithm's end-to-end performance using several objective metrics, on both synthetic and real data. We give comparisons to ground truth for synthetic data, and use a variety of application-specific consistency measures for real data (where no ground truth is available).

4.1 Synthetic Data

This section describes a series of experiments on synthetic data.

4.1.1 Two-Camera Baseline Recovery

We assessed the algorithm's recovery of pairwise baselines by randomly generating 3-D point features and projecting them into the cameras. We introduced controllable projection noise using bipolar Bingham distributions, and a number of random outlier observations.

Figure 14 plots the accuracy of baseline recovery as feature noise, outlier percentage, rotation error, and the number of features are varied. We perturbed the true baseline by a random angle with variance σ^2 and used an uncertainty bound of 3σ.

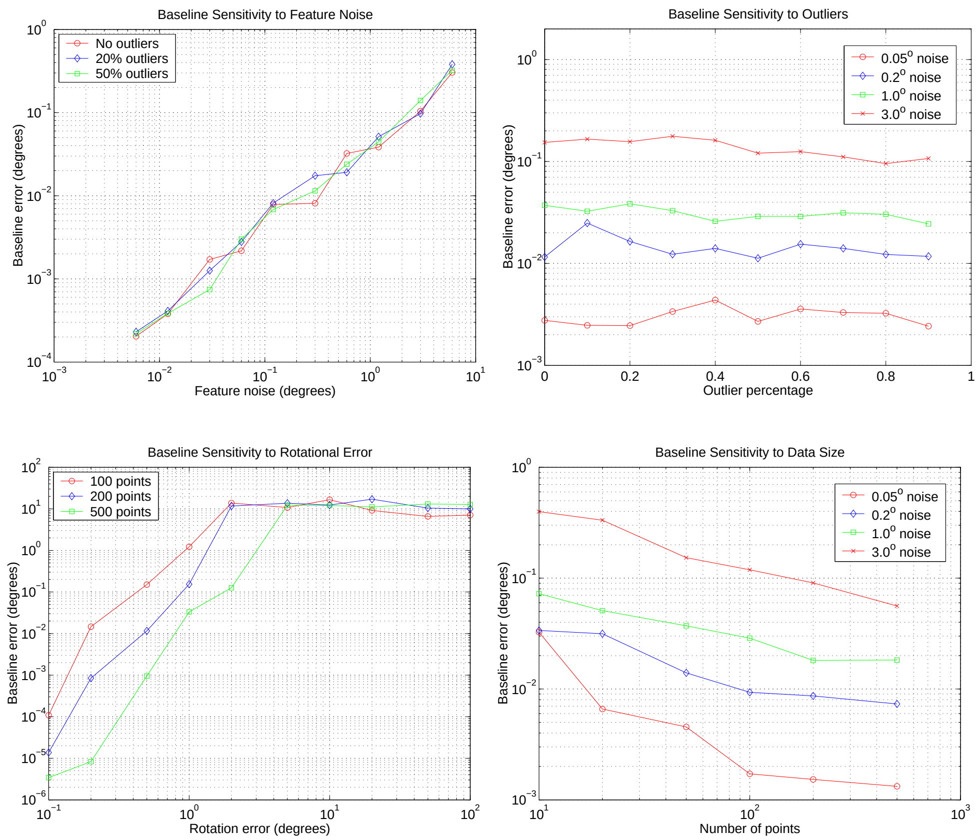

Figure 14: Accuracy of Baseline Estimation

Accuracy of baseline recovery with varying inputs. Error varies roughly linearly with feature noise (upper left). Error is roughly insensitive to the number of outliers (upper right). Error increases rapidly with the error in supplied node rotations (lower left), but eventually plateaus at the explicit bound imposed on the baseline direction (§3.2.2). Error decreases with increasing number of sample points (lower right).

The recovered baseline estimates are robust even against extremely high outlier percentages due to the Hough transform initialization. The technique does not fare so well with error in supplied input rotations, because the epipolar formulation fundamentally depends on accurate camera orientations. Intuitively, poor node rotations "scatters" the accumulation of epipolar lines.

We assessed the MCEM component of the algorithm by visualizing the match probability matrix (§3.4.2) at different stages of its evolution (Figure Figure 15). The match matrices produced by MCEM do not perfectly capture feature correspondence when significant noise is present. However, this correspondence is never explicitly required, since the end-to-end performance measure is baseline accuracy (Figure 17).

Finally, we compared the baseline direction estimates obtained by the MCEM algorithm to those produced by a deterministic Iterated Closest Point (ICP) method (Figure 16). 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" explicit (i.e., binary) matches given the current baseline direction. MCEM consistently outperforms ICP,

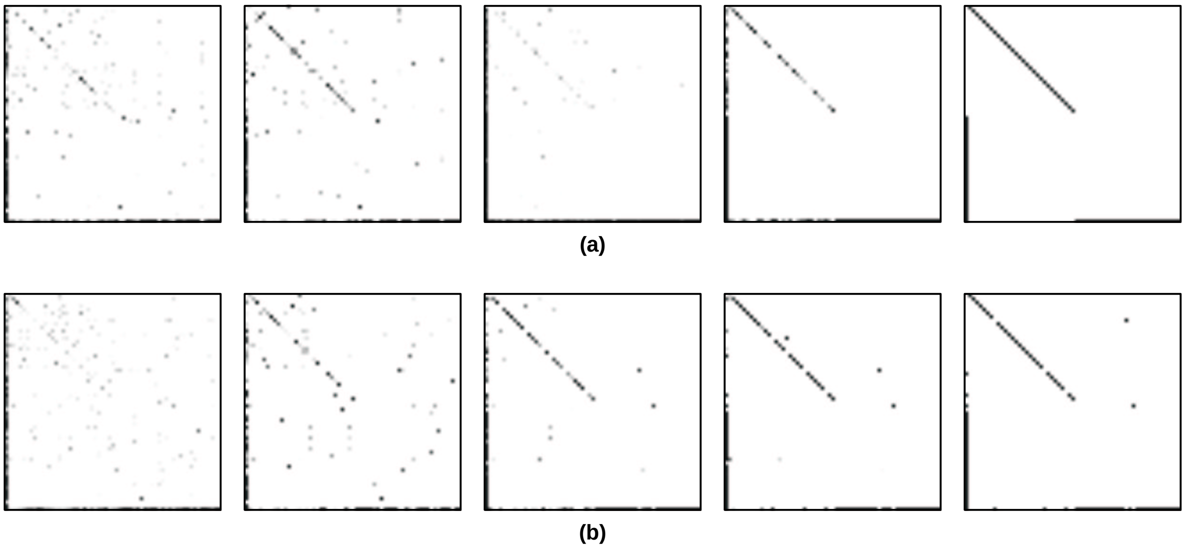

Figure 15: Evolution of MCEM Match Probability Matrix
Evolution of the match matrix as the MCEM algorithm proceeds. (a) Successive iterations for point feature noise of $0.05°$; correspondence is perfectly recovered. (b) Iterations for point feature noise of $0.5°$; a few features are misclassified.

exhibiting less error as both feature noise and the number of outliers increases.

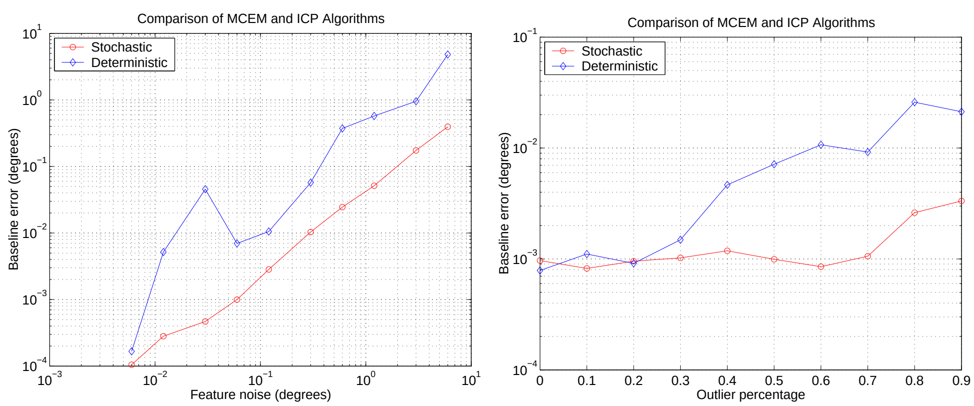

Figure 16: Baseline Recovery with MCEM and ICP Methods
Baseline recovery error is plotted for the (stochastic) MCEM and (deterministic) ICP methods as a function of increasing feature noise (left) and outlier percentage (right). MCEM outperforms ICP in both cases.

4.1.2 Global Registration

We assessed the accuracy of the global registration stage, which determines a consistent set of node positions given all inter-camera baseline directions. We generated a collection of camera positions (and thus known baselines) randomly, then perturbed the baselines by a Bingham noise process with controllable parameters. We then recovered an end-to-end pose assignment and compared the recovered and initial "true" node positions (Figure 17). As expected, position recovery error grows with the amount of baseline perturbation. Recovery error does not decrease significantly with the total number of cameras, since only a constant number of constraints (one for each adjacency) are used to determine each node position.

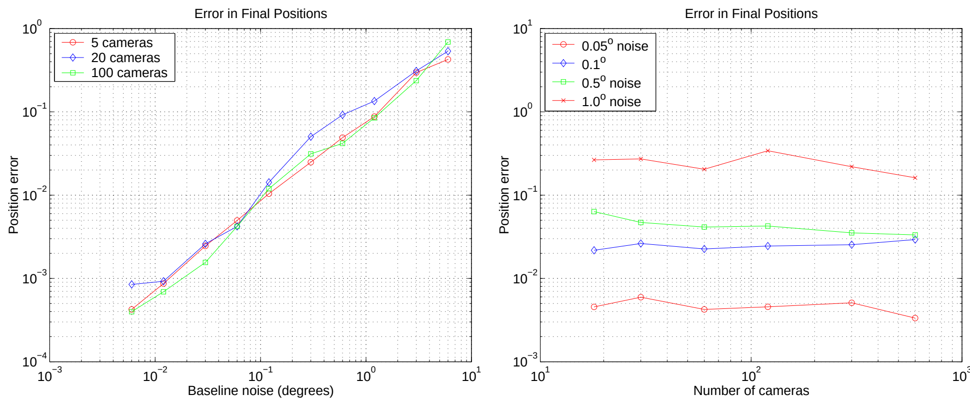

Figure 17: Error in Global Position Recovery

Error in global position recovery, as a function of baseline error (left) and number of nodes (right).

4.2 Real Data

We assessed the end-to-end performance of the registration method for several real datasets acquired as part of the overall model capture project. In lieu of ground truth, which is not available in general and may be difficult or impossible to obtain, we formulated and evaluated a variety of consistency metrics. We report the following quantities for each dataset:

- ***Data size.*** We tabulate the number of rectangular images ("Images"), the number of omni-directional nodes ("Nodes"), and the number of images per node. We report the average and total number of point features detected ("Points"). Finally, we report the number of adjacent camera pairs ("Node Adjacencies") and the average distance between adjacent cameras.

- ***Computation time.*** We report average and total running times for each stage of position recovery, excluding file I/O.

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

- ***End-to-End position error.*** We report uncertainty estimates for the recovered node positions ("Trans Bound") by evaluating the average and maximum sizes at which 95% confidence bounds are reached for the recovered Gaussian densities.

- ***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 a threshold (corresponding roughly to 80% probability) 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").

4.2.1 Technology Square Data Set: Consistency of Pose Recovery

The Technology Square data set consists of 81 nodes spanning an area of roughly 285 by 375 meters (Figure 18). Our orientation alignment algorithm [3] registered 75 (or roughly 92%) of the 81 nodes; 6 nodes were discarded due to insufficient vanishing point information. Of these 75 nodes, the baseline recovery algorithm registered all 75 successfully.

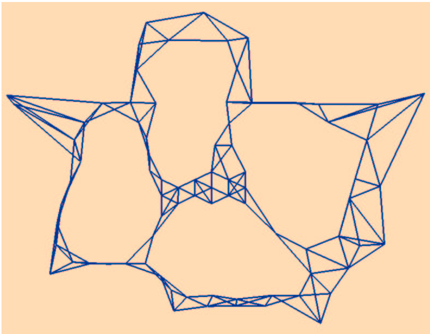

Figure 18: TechSquare Node Configuration

Node positions and adjacencies for the Tech Square data set. The average baseline (for 5 nearest neighbors) was 30.88 meters.

For this data set, our algorithm corrected initial translation errors of nearly seven meters, producing node pose consistent on average to 0.072° of orientation, 5.6 cm of position, 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.

Data Type	Per Image	Per Node	Total
Images	—	48	3899
Point Features	227	10,958	887,598
Nodes	—	—	81
Node Adjacencies	—	—	189

	Per Pair	Total
Baseline Hough	8.1 s	25 m 31 s
Baseline MCEM	45.3 s	2 h 23 m
Global Opt	—	0 m 53 s
Total	53.4 s	2 h 49 m

Table 1: TechSquare Data Size, and Computation Times by Stage

	Average	Maximum
Rot Offset	1.53°	17.18°
Rot Bound	0.072°	0.098°
Trans Offset	0.70 m	6.70 m
Trans Bound	5.6 cm	11.0 cm

	Average	Maximum	Std. Dev.
3-D Ray Distance	9.6 cm	12.4 cm	3.3 cm
2-D Epi Distance	1.22 pixel	5.71 pixel	2.33 pixel

Table 2: Tech Square: 3-D and 2-D (Epipolar) Consistency

4.2.2 Technology Square Data: Comparison to Manual Pose Recovery

A manually generated pose solution was available for this dataset [18], enabling us to compare manual and automatic pose recovery techniques. Entering five or more point matches by hand for each of roughly 200 adjaciences, expending only one minute per match, would require about 16 hours of human effort; thus the student operator omitted many point matches, producing a merely convergent (but not stable) constraint set. Figure 19 compares epipolar geometry for manual and automated pose recovery.

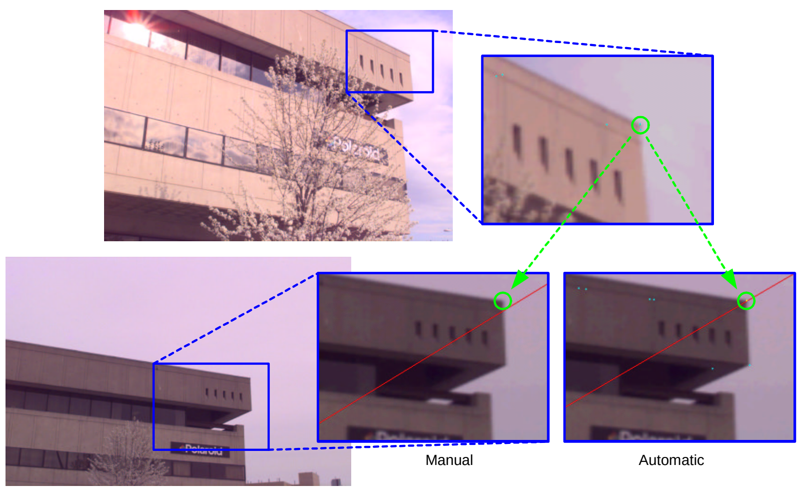

Figure 19: TechSquare Epipolar Geometry Comparison I

A point feature in one image and its corresponding epipolar line in another image, as computed using cameras generated by manual correspondence (bottom middle) vs. our automatic method (bottom right). Note the error in the manual solution, in this case due to insufficient manually-entered match constraints.

Figure 20 compares epipolar geometry for a window corner from a repeating series of windows obscured by foliage. Again, the manual solution has poor epipolar geometry, since the human user did not enter this particular match constraint. We observe that it is plainly impossible to match these window corners *given only this pair of images*, due to the limited camera FOV; even for the omni-directional image pair, human operators find it difficult or impossible to match window corners due to the severe clutter from foliage obscuring most individual views. Our algorithm succeeds where the human fails by combining many omnidirectional observations of many point features, and iteratively reweighting match probabilities until a self-consistent set emerges.

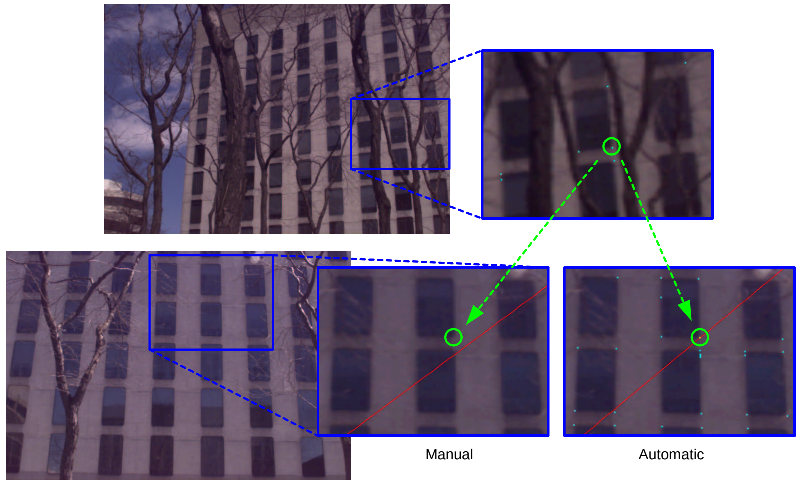

Figure 20: TechSquare Epipolar Geometry Comparison II
A feature whose match is difficult for a human operator to identify. Epipolar geometry is shown for manual (bottom middle) and automated (bottom right) pose solutions. Note the error in the manual solution.

4.2.3 GreenBuilding Data Set (30 nodes)

We identified a small node set with particularly noisy initial pose, in order to test the robustness of the automatic techniques with respect to initial pose error. These 30 nodes spanned an area of roughly 80 by 115 meters (Figure 21); all were successfully registered rotationally and translationally (i.e., end-to-end).

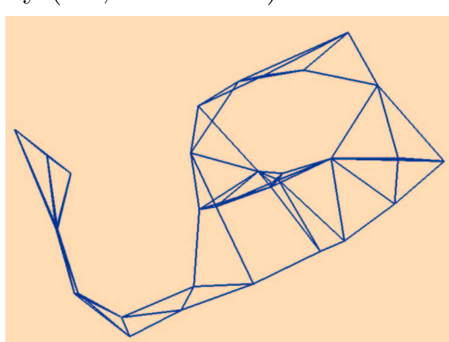

Figure 21: GreenBuilding Node Configuration
Node locations and adjacencies for the GreenBuilding data set. The average baseline was 15.61 meters.

The rotational registration stage corrected initial orientation errors of 6.83°. For this data set, our algorithm corrected initial translation errors of nearly six meters, producing node pose consistent on average to 0.067° of orientation, 4.5 cm 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.

The Green Building node set had particularly noisy initial height estimates (Figure 22), so we studied the algorithm's ability to recover consistent node height (or z).

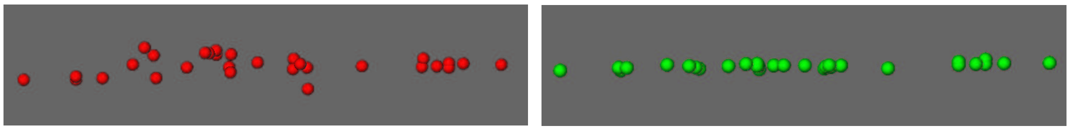

Figure 22: GreenBuilding Height Corrections

(a) A horizontal view of the node topology before pose refinement. All nodes were acquired at roughly the same height above the ground; noisy GPS caused poor initial z estimates for the nodes. (b) After refinement, most of the height variation has been corrected.

Even for nearby 3-D points, the initial epipolar error in this dataset is substantial, roughly hundreds of pixels (Figure Figure 23). The registration algorithm finds accurate epipolar geometry for these points.

Finally, we examined point features known to be very far from all nodes, to assess the algorithm's ability to recover consistent pose away from the immediate vicinity of the acquiring cameras (Figure 24). With poor initial pose, and distant 3-D feature points, our algorithm recovers both node and feature positions to within a few centimeters.

Data Type	Per Image	Per Node	Total
Images	—	23	695
Nodes	—	—	30
Point Features	257	5,967	179,030
Node Adjacencies	—	—	80

	Per Pair	Total
Baseline Hough	6.2 s	8 16 m
Baseline MCEM	42.5 s	56 20 m
Global Opt	—	0 21 m
Total	48.7 s	1 05 h

Table 3: Green Building: Data Size and Computation Times by Stage

	Average	Maximum
Rot Offset	2.95°	6.83°
Rot Bound	0.067°	0.12°
Trans Offset	2.86 m	5.97 m
Trans Bound	4.5 cm	8.1 cm

	Average	Maximum	Std. Dev.
3-D Ray Distance	10.2 cm	18.5 cm	5.3 cm
2-D Epi Distance	2.21 pixel	4.17 pixel	1.43 pixel

Table 4: Green Building: 3-D and 2-D (Epipolar) Consistency

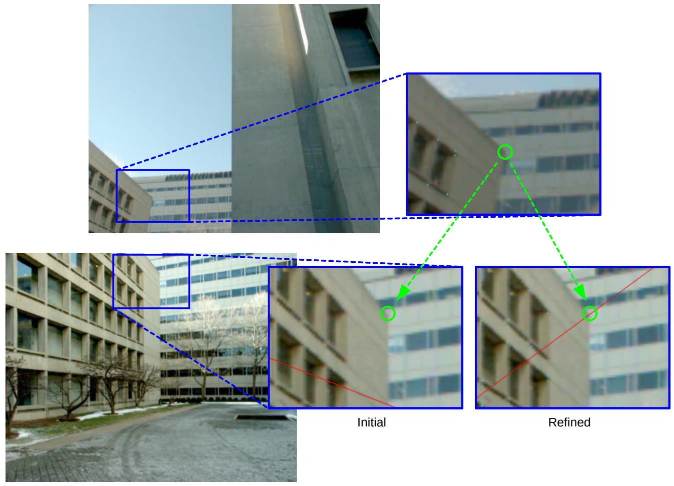

Figure 23: GreenBuilding Epipolar Geometry Comparison
Initial and refined epipolar geometry; the algorithm corrects significant initial pose error.

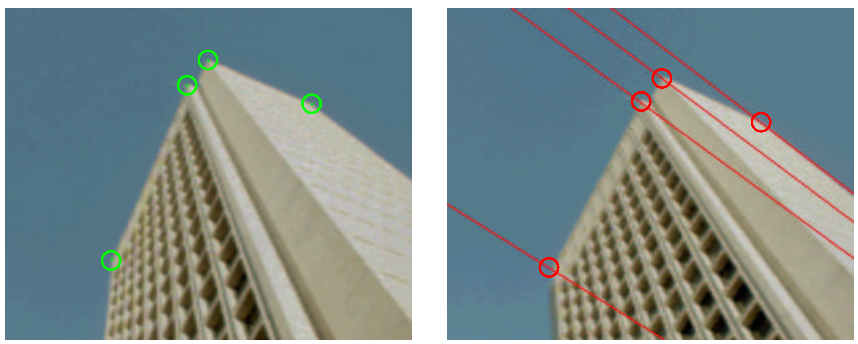

Figure 24: GreenBuilding Epipolar Geometry for Distant Points
Epipolar lines after registration are consistent to within a few pixels, even for distant 3-D points. Error in initial pose is substantial; the same lines inferred from the this pose fail to intersect the image.

4.2.4 AmesCourt Data Set (100 nodes)

The Ames Court data set spans an area of 315 by 380 meters, representing a larger geographical region and a larger number of camera sites (Figure 25). Of the 100 nodes in this set, the rotational stage registered 95 successfully. The translation stage registered all 95 nodes.

Initial pose was corrected by 5.59° and 6.18 m, 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 under four hours.

4.3 Benefit of Omni-Directional Imagery

There is substantial experimental evidence that wide-FOV (i.e., omni-directional) images are fundamentally more powerful than narrow-FOV (i.e., planar) images in practice. Our companion paper [3] showed evidence that vanishing point estimation becomes more robust

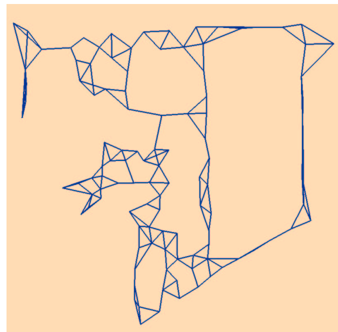

Figure 25: AmesCourt Node Configuration
Node locations and adjacencies for the AmesCourt data set. The average baseline was 23.53 meters.

	Per Image	Per Node	Total
Images	—	20	2,000
Nodes	—	—	100
Point Features	257	4, 132	413,254
Node Adjacencies	—	—	232

	Per Pair	Total
Baseline Hough	7.8 s	30 m 10 s
Baseline MCEM	52.6 s	3 h 24 m
Global Opt	—	1 m 04 s
Total	60.4 s	3 h 55 m

Table 5: AmesCourt Data Size and Computation Times by Stage

	Average	Maximum
Rot Offset	2.83°	5.59°
Rot Bound	0.095°	0.21°
Trans Offset	3.53 m	6.18 m
Trans Bound	5.7 cm	8.8 cm

	Average	Maximum	Std. Dev.
3-D Ray Distance	14.9 cm	20.2 cm	5.6 cm
2-D Epi Distance	3.88 pixel	5.02 pixel	2.10 pixel

Table 6: Ames Court: 3-D and 2-D (Epipolar) Consistency

and more accurate with increasing field of view. Here, we show analogous evidence for position (baseline) recovery. We examined the Hough transform, and resulting baseline direction estimate, for a node pair as a function of the number of planar images used. Transform values are plotted in Figure 27. The sharpness of the peak, and the consistency of the resulting baseline estimate, increases directly with field of view. Moreover, we observe that narrow-FOV images do not provide sufficient feature overlap for convergence in any of our datasets.

Figure 26: AmesCourt Epipolar Geometry
Point features and corresponding epipolar lines for a typical node pair in the AmesCourt set.

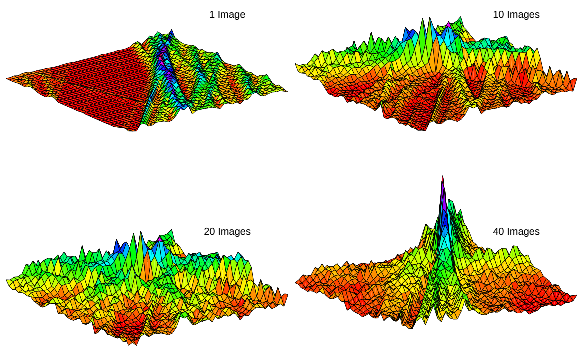

Figure 27: Hough Transform Peak Coherence
The dependence of Hough transform peak coherence on field of view for nodes containing 47 images. Peaks are shown for a baseline direction, for increasing numbers of images (1, 10, 20 and 40) in the node tiling.

4.4 Other Error Sources

The algorithm presented in this paper relies upon the outputs of a number of other algorithms, including camera calibration and noisy feature detection. Without careful surveying of ground-truth 3-D measurements, it is difficult to quantitatively judge the system's end-to-end performance on real data. However, the consistency measures above suggest that node pose is recovered accurately.

5 Related Work

This section reviews prior work in 3-DOF baseline estimation and 6-DOF registration.

5.1 Interactive Pose Estimation Methods

Interactive tools can also be used to impose constraints on camera pose [7, 20, 46]. These tools would require a prohibitive amount of manual effort to register a large image network. They are also vulnerable to operator error, and to numerical instability: since human operators have a finite capacity for work, they will tend to specify as few constraints as possible to achieve convergence.

5.2 Controlled Calibration

Many vision applications assume static cameras [49, 40], or a fixed spatial configuration of two or more cameras [29]. Some systems recover relative pose through the use of known targets [55, 14]. These techniques require 3-D to 2-D correspondence in each calibration image, usually supplied by a human operator, or determined automatically as long as the target remains in clear view. These methods have two principal disadvantages: they require static camera configurations, and they require that known objects be present in the scene.

5.3 Structure from Motion

A class of *structure from motion* (*SFM*) techniques recovers scene geometry and camera pose for a moving camera [35, 42, 51]. These methods are sensitive to image noise, illumination variations, and strong perspective or occlusion due to extended baselines.

Some methods recover pose only between consecutive image pairs or triples in a sequence [24]; these local techniques are prone to bias and error accumulation. Azarbayejani [5] addresses this issue by using an extended Kalman filter to update structure and motion using all available data, incrementally improving the estimates as new data is introduced.

Projective reconstruction techniques avoid intrinsic camera calibration [39, 36, 28], recovering structure and pose only up to an arbitrary projective transformation. Other linearized versions of SFM have been formulated, based on SVD [41] or affine approximations [33].

5.4 Correspondence Methods

Nearly all registration algorithms rely on explicit knowledge of correspondence between features. Low-level trackers [50] and dense texture trackers [31, 58] attempt to compute correspondence under short or infinitesimal baselines (i.e., for situations in which scene brightness and viewpoint change little across images, and there is little or no occlusion).

Robust statistical techniques have been developed to diminish the effects of outliers. Examples include RANSAC (random sampling consensus) [23, 24], MLESAC [54], ROR (rejection of outliers by rotations) [1], and LMS (least median of squares) [13, 49]. These algorithms attempt to find consistent pose assignments by randomly choosing feature subsets and examining remaining features for consistency. However, they can require exponential time in the number of candidate features; they do not account for match ambiguities, or feature noise; and they do not sample the space of all feature sets in a principled way.

Other authors formulate correspondence probabilistically rather than explicitly [44, 15, 21]. None of these techniques have been demonstrated for large numbers of features or extended camera motions. Correspondence-free pose estimation techniques have also emerged (e.g., [25]), but have not been demonstrated for scenes with significant occlusion or lighting variation.

5.5 Measurement Uncertainty

Most authors who have treated measurement uncertainty have used additive Gaussian noise [48, 2, 37, 27]. This noise model has no meaningful interpretation for projective variables. For example, the units of the covariance of the fundamental matrix [19, 59] are not well-defined. Bingham distributions have been shown to be more appropriate for projective variables [17].

5.6 Expectation Maximization Methods

Some authors have proposed EM or EM-like algorithms to solve coupled structure and camera motion problems [57, 9, 15], but none have provided a principled treatment of measurement noise and matching ambiguity. Recently, a probababilistic EM formulation has been proposed [21], which handles multiple images and match ambiguity, but only when the number of 3D features is known, and all features are visible in all images.

6 Contributions and Conclusions

The algorithm described in this paper makes use of a number of fundamental techniques from computer vision and estimation theory, including: the use of gradient-based (point) features for robustness against lighting variations and strong perspective; decoupling 6-DOF pose estimation into two pure 3-DOF problems; probabilistic inference on the sphere; the Hough transform (for efficiently establishing priors), Markov chain Monte Carlo methods (for efficiently sampling from high-dimensional probability spaces), and expectation maximization methods (for iterative solution of coupled classification and estimation problems).

This paper makes several contributions to the recovery of absolute positions for large collections of cameras, and attainment of large-scale 6-DOF extrinsic calibration. First, we propose the use of *a priori* absolute position estimates, and an image adjacency graph, to limit inter-camera registration to those images which are likely to have observed common scene structure. This enables $O(n)$ rather than $O(n^2)$ asymptotic performance, and removes the need for a human operator to supply matching constraints (for example to initialize a bundle-adjustment optimization), or supply photogrammetric tie points to express the resulting pose in an absolute (Earth) coordinate system.

Second, we show quantitative evidence that wide-FOV (omni-directional) images are fundamentally more powerful observations than are narrow-FOV (planar) images for the recovery of inter-image baselines and global positions. Omni-directional images are free of the aperture problem and its attendant ambiguities; nearby clusters of wide-FOV images generally observe more common scene structure than do clusters of narrow-FOV cameras.

Third, we extend existing probabilistic feature correspondence methods to handle unknown numbers of features, unknown occlusion, deocclusion, and outlier features, and to correctly incorporate projective uncertainty.

Fourth, we combine Hough transform and MCMC techniques to address the limitations of both methods. The HT is used in a discrete fashion simply to establish a prior probability on the set of possible point matches. MCMC is used for stochastic optimization of the baseline estimate under balanced match insertions, deletions, and swaps.

Fifth, we describe a method to incorporate a set of pairwise baseline direction constraints, each with an attendant uncertainty, into a global, linear least-squares optimization which produces accurate estimates of final node positions and an aggregate uncertainty for each.

Sixth, we assessed end-to-end error of the 6-DOF pose recovery system which incorporates the position estimation algorithm proposed in this paper. Even in the presence of significant initial position and orientation error (several meters and several degrees), our algorithms recover absolute pose accurately while requiring a few CPU-hours of computation. To our knowledge, the resulting datasets are the largest registered terrestrial image datasets in existence, regardless of whether manual or automated calibration algorithms are used. We estimate that producing equivalent datasets using manual photogrammetric bundle-adjustment would require between tens and hundreds of hours of human effort.

Finally, the algorithm described in this paper expends time and space resources which grow linearly in the number of input images, rather than quadratically or worse as in many previous methods. This removes a fundamental barrier to the development of automated registration techniques for very large numbers of images. In practice, we demonstrated the algorithm's performance on datasets containing roughly one, two, and four thousand images, complexities which can not be attained with any other automated method, and which would be difficult or impossible in an interactive system.

One perhaps unexpected advantage of working at this scaling regime is that of overconstraints and data fusion to reduce uncertainty; our algorithms register images to within four pixels of epipolar error, on average, outperforming manual bundle-adjustment due to the human operator's use of insufficient constraints. We emphasize that the image datasets for which we report performance were acquired outdoors, over wide baselines, under uncontrolled and varying lighting conditions, and in the presence of significant occlusion and visual clutter. Considered together, the algorithms presented here and in [3] represent a new end-to-end capability for automated, absolute registration of terrestrial images.

7 Acknowledgements

Support for this research was provided in part by the Office of Naval Research under MURI Award SA 1524-2582386, and in part by the NTT Corporation under Award MIT9904-20.

References

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

[2] 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.

[3] Matthew Antone and Seth Teller. Automatic recovery of relative camera rotations for urban scenes. In *CVPR*, 2000 (to appear).

[4] Matthew E. Antone and Seth Teller. Automatic recovery of camera positions in urban scenes. Technical Report MIT-LCS-814, Massachusetts Institute of Technology Laboratory for Computer Science, December 2000.

[5] 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.

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

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

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

[9] P. J. Besl and H. D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

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

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

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

[13] 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.

[14] X. Chen, J. Davis, and P. Slusallek. Wide area camera calibration using virtual calibration objects. In *Proc. CVPR*, pages 520–527, June 2000.

[15] 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.

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

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

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

[19] 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.

[20] 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.

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

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

[23] 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.

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

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

[26] Joshua Gluckman and Shree Nayar. Ego-motion and omnidirectional cameras. In *ICCV*, pages 35–42, 1998.

[27] 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.

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

[29] Berthold K. P. Horn. *Robot Vision*. MIT Press, Cambridge, MA, 1986.

[30] 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.

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

[32] 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.

[33] Fredrik Kahl and Anders Heyden. Robust self-calibration and Euclidean reconstruction via affine approximation. In *Proc. ICPR*, 1998.

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

[35] 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.

[36] 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.

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

[38] 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.

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

[40] P. J. Narayanan, Peter W. Rander, and Takeo Kanade. Constructing virtual worlds using dense stereo. In *Proc. ICCV*, pages 3–10, January 1998.

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

[42] 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.

[43] Long Quan and Roger Mohr. Matching perspective images using geometric constraints and perceptual grouping. In *Proc. ICCV*, pages 679–684, 1988.

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

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

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

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

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

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

[50] 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.

[51] 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.

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

[53] Seth Teller. Automated urban model acquisition: Project rationale and status. In *Proc. of the Image Understanding Workshop*, pages 455–462, November 1998.

[54] 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.

[55] Roger Y. Tsai. A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE J. of Robotics and Automation*, 3(4):323–344, 1987.

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

[57] William Wells. Statistical approaches to feature-based object recognition. *IJCV*, 21(1/2):63–98, January 1997.

[58] 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.

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