Structure from Small Baseline Motion with Central Panoramic Cameras*

Omid Shakernia René Vidal Shankar Sastry
Department of Electrical Engineering & Computer Sciences, UC Berkeley
{omids,rvidal,sastry}@eecs.berkeley.edu

Abstract

In applications of egomotion estimation, such as real-time vision-based navigation, one must deal with the double-edged sword of small relative motions between images. On one hand, tracking feature points is easier, while on the other, two-view structure-from-motion algorithms are poorly conditioned due to the low signal-to-noise ratio. In this paper, we derive a multi-frame structure from motion algorithm for calibrated central panoramic cameras. Our algorithm avoids the conditioning problem by explicitly incorporating the small baseline assumption in the algorithm's design. The proposed algorithm is linear, amenable to real-time implementation, and performs well in the small baseline domain for which it is designed.

1. Introduction

In applications of structure from motion (SFM) such as mobile robot navigation, camera frame rates are typically high, which means that the relative motion between images frames is small relative to the scene depth. This scenario is a double-edged sword for the SFM problem: on one hand, tracking feature points is easier because they are closer in the image plane, while on the other hand, SFM algorithms are poorly conditioned because the small baseline translation between images causes a low signal-to-noise ratio.

One approach to tackling this problem is to approximate the camera motion as a *velocity* and recover the infinitesimal camera motions from the *optical flow* in the image plane. This approach has been taken for the two-view case in [8, 14, 12], and in the multiple view case in[10, 11].

Another approach is to track discrete feature points and perform pairwise SFM for central panoramic cameras as proposed in [3, 13, 6]. However, as noted above, the two-view SFM algorithms are very sensitive to noise in applications such as outdoor navigation because the small baseline motion makes the triangulation poorly conditioned. An approach to overcoming the small baseline problem by using multiple central panoramic images was presented in [4]. However, their approach is not applicable to motion estimation for navigation and discards much information because it chooses *multiple image pairs* for which the essential matrix is well conditioned and then applies two-view SFM.

In this paper, following the approach of [9], we derive the first *multi-frame* structure from motion (MFSFM) algorithm for calibrated central panoramic cameras. Our algorithm avoids the conditioning problem of small baseline motion by using *all* the information available in a central panoramic image sequence, and by *explicitly* incorporating the small baseline assumption in the algorithm's design. The proposed algorithm is linear, amenable to real-time implementation, and through extensive simulation with synthetic image sequences, we show that it performs well in the small baseline domain for which it is designed.

2. Central Panoramic Cameras

A catadioptric realization of an omnidirectional camera combines a curved mirror and a lens. In [1], an entire class of catadioptric systems containing a single effective focal point is derived. A single effective focal point is necessary for the existence of epipolar geometry that is independent of the scene structure [13]. Camera systems that have a unique effective focal point are called *central panoramic cameras*.

2.1. Projection Model

It was shown in [5] that all central panoramic cameras can be modeled by a mapping of a 3D point onto a sphere followed by a projection onto the image plane from a point in the optical axis of the camera. By varying two parameters (ξ, m), one can model all catadioptric cameras that have a single effective viewpoint, *e.g.* parabolic mirror with orthographic lens, or hyperbolic mirror with perspective lens. The particular values of (ξ, m) in terms of the shape parameters of different types of mirrors are listed in [2].

According to the unified projection model [5], the image point $(x, y)^T$ of a 3D point $q = (X, Y, Z)^T$ obtained through a central panoramic camera with parameters (ξ, m) is given by:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{\xi + m}{-Z + \xi\sqrt{X^2 + Y^2 + Z^2}} \begin{bmatrix} s_x X \\ s_y Y \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix}, \quad (1)$$

where $0 \leq \xi \leq 1$, and (s_x, s_y) are scales that depend on the

*We thank the support of ONR grant N00014-00-1-0621.

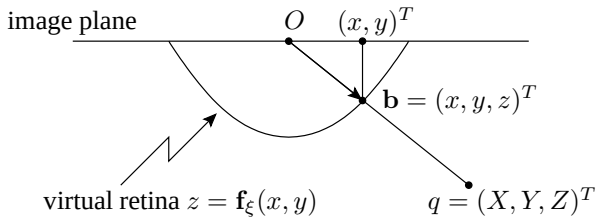

Figure 1: Showing the curved virtual retina in central panoramic projection and back-projection ray $\mathbf{b}$ associated with image point $(x,y)^T$.

geometry of the mirror, the focal length and the aspect ratio of the lens, and $(c_x, c_y)^T$ is the mirror center.

Since central panoramic cameras for $\xi \neq 0$ can be easily calibrated from a single image of three lines [7, 2], in this paper, we assume that the camera has been calibrated, *i.e.* we know the parameters $(s_x, s_y, c_x, c_y, \xi, m)$. Therefore, without loss of generality, we consider the following *calibrated* central panoramic projection model:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} X \\ Y \end{bmatrix}, \quad \lambda \triangleq -Z + \xi\sqrt{X^2+Y^2+Z^2} \tag{2}$$

which is valid for $Z < 0$. It is direct to check that $\xi = 0$ corresponds to the case of perspective projection, and $\xi = 1$ corresponds to paracatadioptric projection (a catadioptric camera with a parabolic mirror and an orthographic lens).

2.2. Back-projection Rays

Since central panoramic cameras have a unique effective focal point, one can efficiently compute the *back-projection ray* (a ray from the optical center in the direction of the 3D point being imaged) associated with each image point.

We showed in [12] that one may consider the central panoramic projection model in equation (2) as a simple projection onto an curved virtual retina whose shape depends on the parameter ξ. We define the *back-projection ray* as the *lifting* of the image point $(x,y)^T$ onto this retina. That is, as shown in Figure 1, given an image $(x,y)^T$ of a 3D point $q = (X,Y,Z)^T$, define the back-projection rays as:

$$\mathbf{b} \triangleq (x,y,z)^T, \tag{3}$$

where $z = \mathbf{f}_\xi(x,y)$ is the height of the virtual retina. We construct $\mathbf{f}_\xi(x,y)$ in order to re-write the central panoramic projection model as a simple scaling:

$$\lambda \mathbf{b} = q, \tag{4}$$

where λ is the scale in (2) which is lost in the projection. Using equations (4) and (2), it is direct to solve for the height of the virtual retina as:

$$z \triangleq \mathbf{f}_\xi(x,y) = \frac{-1+\xi^2(x^2+y^2)}{1+\xi\sqrt{1+(1-\xi^2)(x^2+y^2)}}. \tag{5}$$

Then, we can re-write the central panoramic projection (2) in terms of the back-projection ray as $\mathbf{b} = \pi_\xi(q)$, where:

$$\begin{aligned} \pi_\xi : \mathbb{R}^3 &\rightarrow \mathcal{R}_\xi \\ q &\mapsto \frac{q}{-e_3^T q + \xi\|q\|}, \end{aligned} \tag{6}$$

with $e_3 = (0,0,1)^T \in \mathbb{R}^3$ and

$$\mathcal{R}_\xi = \{(x,y,z)^T \in \mathbb{R}^3 \mid x^2+y^2 \leq 1, z = \mathbf{f}_\xi(x,y)\}. \tag{7}$$

It is clear (and will be useful in Section 3) that for any nonzero $q \in \mathbb{R}^3$ and $\gamma \in \mathbb{R}$ we have $\pi_\xi(\gamma q) = \pi_\xi(q)$. Notice also that when $\xi = 0$, $\mathcal{R}_\xi$ is the (perspective) image plane and π_ξ is simply perspective projection. Similarly, when $\xi = 1$, $\mathcal{R}_\xi$ is the parabolic surface of the mirror and π_ξ is paracatadioptric projection.

3. Multi-frame Structure from Motion

In this section, we present a multiple frame structure from motion (MFSFM) algorithm designed for central panoramic image sequences. Using the back-projection ray defined in Section 2.2, we give a natural generalization of the perspective MFSFM algorithm [9] to central panoramic projection.

We consider an image sequence containing N image points in F frames taken by a moving central panoramic camera with projection model (2). Consider a set of static 3D points $q_p \in \mathbb{R}^3$ for $p = 1, \ldots, N$. If the camera undergoes a rotation $R^i \in SO(3)$ and translation $T^i \in \mathbb{R}^3$ between the *base frame* $(i = 0)$ and the subsequent camera frames $i = 1, \ldots, F-1$, then the coordinates of q_p in camera frame i are given by:

$$q_p^i = R^i(q_p + T^i). \tag{8}$$

By using (5), we compute the back-projection ray $\mathbf{b}_p$ for $p = 1, \ldots, N$ in the base frame such that $\lambda_p \mathbf{b}_p = q_p$. We call λ_p the *structure* since it is the scale term in (2) which is lost in the projection, or equivalently, the scale by which one multiplies the back-projection ray $\mathbf{b}_p$ to recover the 3D point q_p. The corresponding back-projection rays $\mathbf{b}_p^i$ in the image frames $i = 1, \ldots, F-1$ are given by:

$$\mathbf{b}_p^i = \pi_\xi\left(R^i(q_p + T^i)\right) = \pi_\xi\left(R^i(\mathbf{b}_p + T^i/\lambda_p)\right). \tag{9}$$

Our multi-frame structure from motion algorithm is designed to work under the following assumptions.

Assumption 3.1. The baseline translation is small. *Specifically, we assume the ratio between the largest translation and the smallest depth is small, i.e.* $\tau \triangleq \|T\|_{\max}/\lambda_{\min} < 1$.

Assumption 3.2. The translation is general. *Specifically, we assume the matrix of translations* $T = [T^1 \cdots T^{F-1}] \in \mathbb{R}^{3\times(F-1)}$ *is of rank 3.*

Assumption 3.1 is easily satisfied, for example, in outdoor navigation where the images are obtained by a camera which moves slowly compared to the frame rate. Assumption 3.2 is more restrictive. For example, it is violated in applications such as mobile robot navigation in the X-Y plane. In Section 3.3 we give a modified version of the general algorithm for the case of estimating motion in the X-Y plane. It is important to notice that we make *no assumption* on the rotational motion of the camera.

3.1. Back-projection Displacements

When the translation baselines are small, the rotations are easy to recover relatively accurately by standard linear techniques. Specifically, one may compute estimates R^i_{est} of the the rotations R^i by assuming zero translation and solving linearly for R^i from[1]:

$$[\mathbf{b}^i_p]_\times R^i \mathbf{b}_p \approx 0, \quad p = 1, \ldots, N$$

and then projecting onto $SO(3)$. It can be shown [9] that if the baseline τ is small, then the error in rotation estimate will be small as well (of order τ). That is, to first order we have ${R^i_{\text{est}}}^T R^i = \exp([\Omega^i]_\times) \approx I + [\Omega^i]_\times$, where $\|\Omega^i\| = o(\tau)$.

Given an estimate R^i_{est} for the camera rotation in frame i, we compute the so-called *back-projection displacement* between the *un-rotated* back-projection ray at frame i, ${R^i_{\text{est}}}^T \mathbf{b}^i$ (we drop the subscript p for simplicity), and its corresponding back-projection ray in the base frame $\mathbf{b}$ as:

$$\Delta\mathbf{b}^i \triangleq \pi_\xi({R^i_{\text{est}}}^T \mathbf{b}^i) - \mathbf{b}. \tag{10}$$

Since both translation and the error in rotation estimate are small, we make the following first order approximation:

$$\begin{aligned}
\pi_\xi({R^i_{\text{est}}}^T \mathbf{b}^i) &= \pi_\xi\left({R^i_{\text{est}}}^T R^i(\mathbf{b} + T^i/\lambda)\right) \\
&= \pi_\xi\left((I + [\Omega^i]_\times)(\mathbf{b} + T^i/\lambda)\right) + o(\tau^2) \\
&= \pi_\xi\left(\mathbf{b} + [\Omega^i]_\times \mathbf{b} + T^i/\lambda\right) + o(\tau^2) \\
&= \pi_\xi(\mathbf{b}) + \frac{\partial \pi_\xi}{\partial q}\left([\Omega^i]_\times \mathbf{b} + T^i/\lambda\right) + o(\tau^2),
\end{aligned}$$

where and the Jacobian $\frac{\partial \pi_\xi}{\partial q}$ in the Taylor series is evaluated at $\mathbf{b}$. Notice that by construction $\pi_\xi(\mathbf{b}) = \mathbf{b}$. It is direct to show that the Jacobian of π_ξ evaluated at $\mathbf{b}$ is given by:

$$\frac{\partial \pi_\xi}{\partial q}(\mathbf{b}) = \left(I + \mathbf{b}e_3^T - \frac{\xi^2 \mathbf{b}\mathbf{b}^T}{1 + e_3^T \mathbf{b}}\right), \tag{11}$$

where $e_3 = (0, 0, 1)^T$. Therefore, under the small baseline assumption, we have derived the following first order approximation of the back-projection displacement:

$$\Delta\mathbf{b}^i \approx -(I + \mathbf{b}e_3^T)[\mathbf{b}]_\times \Omega^i + \frac{1}{\lambda}\left(I + \mathbf{b}e_3^T - \frac{\xi^2 \mathbf{b}\mathbf{b}^T}{1 + e_3^T \mathbf{b}}\right) T^i. \tag{12}$$

Then, if $\mathbf{b} = (x, y, z)^T$ and $\Delta\mathbf{b}^i = (\Delta x^i, \Delta y^i, \Delta z^i)^T$, the first two rows in (12) give the following approximation of the *image displacement*:

$$\begin{aligned}
\begin{bmatrix} \Delta x^i \\ \Delta y^i \end{bmatrix} \approx & \begin{bmatrix} xy & z - x^2 & -y \\ -(z - y^2) & -xy & x \end{bmatrix} \Omega^i + \\
& \frac{1}{\lambda} \begin{bmatrix} 1 - \rho x^2 & -\rho xy & (1 - \rho z)x \\ -\rho xy & 1 - \rho y^2 & (1 - \rho z)y \end{bmatrix} T^i,
\end{aligned} \tag{13}$$

where $z = \mathbf{f}_\xi(x, y)$ is given in (5) and

$$\rho \triangleq \frac{\xi^2}{1 + z} = \frac{\xi\left(1 + \xi\sqrt{1 + (1 - \xi^2)(x^2 + y^2)}\right)}{\xi(x^2 + y^2) + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}. \tag{14}$$

The right hand side of (14) is numerically stable as $\xi \to 0$ in which case $z \to -1$ and $\rho \to 0$. Now, given the back-projection displacements $\Delta\mathbf{b}^i_p$ for points $p = 1, \ldots N$ and frames $i = 1, \ldots, F - 1$, we construct the *image displacement matrix* $D \in \mathbb{R}^{2N \times (F-1)}$:

$$D \triangleq \left[\begin{array}{ccc|ccc} \Delta x^1_1 & \cdots & \Delta x^1_N & \Delta y^1_1 & \cdots & \Delta y^1_N \\ \vdots & & \vdots & \vdots & & \vdots \\ \Delta x^{F-1}_1 & \cdots & \Delta x^{F-1}_N & \Delta y^{F-1}_1 & \cdots & \Delta y^{F-1}_N \end{array}\right]^T \tag{15}$$

which we will use as the input to our MFSFM algorithm. It is important to remember that the image displacements $(\Delta x^i, \Delta y^i)^T$ are not simply computed from the given image points, but from the *"un-rotated" back-projection rays* as in equation (10). Notice that when $\xi = 0$, equation (13) gives the perspective image displacements and our algorithm reduces to the MFSFM algorithm of [9].

3.2. Algorithm Description

From equation (13), we have:

$$D \approx \Psi\Omega + \Phi(\{\lambda^{-1}\})T \tag{16}$$

where $\Psi \in \mathbb{R}^{2N \times 3}$, $\Phi(\{\lambda^{-1}\}) \in \mathbb{R}^{2N \times 3}$, and:

$$\Psi = \begin{bmatrix} \{xy\} & \{z - x^2\} & -\{y\} \\ -\{z - y^2\} & -\{xy\} & \{x\} \end{bmatrix}, \tag{17}$$

$$\Phi = \begin{bmatrix} \{\frac{1-\rho x^2}{\lambda}\} & \{\frac{-\rho xy}{\lambda}\} & \{\frac{(1-\rho z)x}{\lambda}\} \\ \{\frac{-\rho xy}{\lambda}\} & \{\frac{1-\rho y^2}{\lambda}\} & \{\frac{(1-\rho z)y}{\lambda}\} \end{bmatrix}, \tag{18}$$

$$T = [T^1 \cdots T^{F-1}] \in \mathbb{R}^{3 \times (F-1)}$$

$$\Omega = [\Omega^1 \cdots \Omega^{F-1}] \in \mathbb{R}^{3 \times (F-1)}$$

where, *e.g.* $\{xy\} = (x_1 y_1, \cdots, x_N y_N)^T \in \mathbb{R}^N$. Equation (16) depends on the residual error in rotation Ω which

[1] Here, $[u]_\times \in so(3)$ represents the skew symmetric matrix generating the cross product, *i.e.* for all $u, v \in \mathbb{R}^3$ we have $u \times v = [u]_\times v$.

is of order τ. Therefore, if we solved for T from (16) by neglecting the residual Ω, we would obtain translation estimates of order τ, *i.e.* the order of the translations themselves. In order to obtain translation estimates with an error of order τ^2, we compute the matrix $H \in \mathbb{R}^{(2N-3)\times 2N}$ which annihilates the rotational flows, *i.e.* $H\Psi = 0$ (this can be done conveniently with SVD). Therefore we have:

$$HD \approx H\Phi(\{\lambda^{-1}\})T.$$

We conclude that $\mathrm{rank}(HD) \approx \mathrm{rank}(T)$, which, by Assumption 3.2, is $\mathrm{rank}(T) = 3$. In order to factorize $HD = SM^T$ into its structure $S = H\Phi(\{\lambda^{-1}\}) \in \mathbb{R}^{(2N-3)\times 3}$ and motion $M = T^T \in \mathbb{R}^{(F-1)\times 3}$ components, let us consider its singular value decomposition (SVD) $HD = \mathcal{U}\Sigma\mathcal{V}^T$. Letting $\tilde{S} = \mathcal{U}$, we have $S = \tilde{S}A$, $\tilde{M}^T = \Sigma\mathcal{V}^T$ and $M = \tilde{M}A^{-T}$ for some nonsingular $A \in \mathbb{R}^{3\times 3}$. One can solve linearly for A and the inverse structure $\{\lambda^{-1}\}$ from:

$$\begin{bmatrix} G_x & -\tilde{S} & 0 & 0 \\ G_y & 0 & -\tilde{S} & 0 \\ G_z & 0 & 0 & -\tilde{S} \end{bmatrix} \begin{bmatrix} \{\lambda^{-1}\} \\ A_1 \\ A_2 \\ A_3 \end{bmatrix} = 0 \qquad (19)$$

where $A_i \in \mathbb{R}^3$ is the i^{th} column of A and

$$\begin{bmatrix} G_x \\ G_y \\ G_z \end{bmatrix} = \begin{bmatrix} H_x\mathrm{diag}(\{1-\rho x^2\}) + H_y\mathrm{diag}(\{-\rho xy\}) \\ H_x\mathrm{diag}(\{-\rho xy\}) + H_y\mathrm{diag}(\{1-\rho y^2\}) \\ H_x\mathrm{diag}(\{(1-\rho z)x\}) + H_y\mathrm{diag}(\{(1-\rho z)y\}) \end{bmatrix}$$

where $H_x \in \mathbb{R}^{(2N-3)\times N}$ is the first N columns of H and $H_y \in \mathbb{R}^{(2N-3)\times N}$ is the last N columns of H such that $H = [H_x\ H_y]$. Then, given A, one can obtain the translation vectors from $T = A^{-1}\tilde{M}^T = A^{-1}\Sigma\mathcal{V}^T$.

Given the new estimates for the translations and structure, one can improve the estimates for rotation by solving linearly for R^i from:

$$[\mathbf{b}_p^i]_\times R^i(\mathbf{b}_p + T^i/\lambda_p) = 0, \quad p = 1, \ldots, N. \qquad (20)$$

Since the current translation estimate has an error of order τ^2, so does the improved rotation estimate. The algorithm then iterates by updating the translation, structure and rotation estimates until they converge. Under the small baseline assumption $\tau < 1$, one can show that the incremental change in the unknowns between the k^{th} and $(k+1)^{th}$ iteration is approximately proportional to τ^k, and hence the algorithm has good convergence properties (see [9] for details). In our experiments, convergence took only 2 to 4 iterations.

The overall MFSFM algorithm can be summarized as follows.

Algorithm 3.3 (Central Panoramic MFSFM Algorithm). *Given N corresponding image points across F image frames of a central panoramic camera, estimate the motion (R^i, T^i) and inverse structure $\{\lambda^{-1}\}$ as follows:*

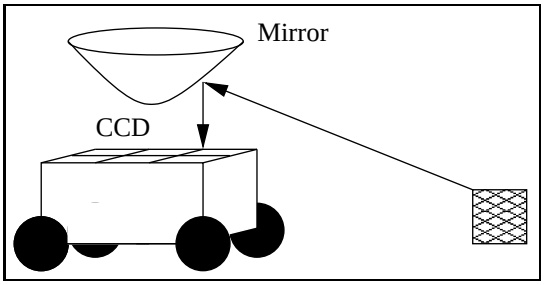

Figure 2: A catadioptric camera attached to a mobile robot.

1. *Initialize $T = 0$, $\{\lambda^{-1}\} = 1$.*
2. *Compute back-projection rays $\{\mathbf{b}_p^i\}$ using (7). Then compute Ψ from (17) and H such that $H\Psi = 0$.*
3. *Solve for R^i_{est} linearly from (20), given T and $\{\lambda^{-1}\}$.*
4. *Given R^i_{est} compute D as in (15). Then using SVD factorize $HD = \mathcal{U}\Sigma\mathcal{V}^T = \mathcal{U}AA^{-1}\Sigma\mathcal{V}^T$.*
   (a) *Solve for $\{\lambda^{-1}\}$ and A linearly from $H\Phi(\{\lambda^{-1}\}) = \mathcal{U}A$.*
   (b) *Solve for $T = A^{-1}\Sigma\mathcal{V}^T$.*
5. *Goto 3. until $(R, T, \{\lambda^{-1}\})$ converge.*

3.3. MFSFM for Motion in the X-Y Plane

Consider the special case where the omnidirectional camera is attached to a mobile robot (see Figure 2) whose motion is restricted to the X-Y plane. Since in this case $\mathrm{rank}(T) = 2$, the factorization method of the general algorithm will break down. However, if the camera is mounted such that the optical axis is aligned normal to the ground plane, then the general algorithm can be easily modified by simply reducing the dimensionality of the factorization. Since with this setup, $T_z^i = 0$, simply consider the translations as two-dimensional $T \in \mathbb{R}^{2\times(F-1)}$ and drop the third column of Ψ in (17) such that $\Psi \in \mathbb{R}^{2N\times 2}$. Then the remainder of the algorithm follows, only with smaller dimensions.

4. Experiments

In this section we evaluate the performance of proposed linear MFSFM algorithm on sequences of synthetic images. By way of comparison, we implemented the maximum likelihood estimator (MLE) which minimizes the re-projection error in the image plane:

$$F(R, T, q) = \sum_{p=1}^{N}\sum_{i=0}^{F-1} \left\| \begin{bmatrix} x_p^i - \pi_\xi(R^i(q_p + T^i))_x \\ y_p^i - \pi_\xi(R^i(q_p + T^i))_y \end{bmatrix} \right\|^2 \qquad (21)$$

using the Levenberg-Marquardt nonlinear optimization routine in MATLAB. We compared the proposed algorithm

against two runs of the MLE: one initialized by the *ground truth*, while the other initialized by the output of the linear MFSFM algorithm.

We compare the linear and MLE algorithm as a function of the *baseline* of translation, the *noise* in the image plane, and the parameter ξ of the *central panoramic camera*. Each experiment consists of 1000 trials for a given setting of parameters. In our comparison, we use the following error measures between the estimates $R_{\text{est}} \in SO(3)$, $T_{\text{est}} \in \mathbb{R}^3$, $\lambda_{\text{est}} \in \mathbb{R}^N$ and the ground truth $R_{\text{true}} \in SO(3)$, $T_{\text{true}} \in \mathbb{R}^3$, $\lambda_{\text{true}} \in \mathbb{R}^N$, averaged over the number of trials (and frames where appropriate):

$$\begin{aligned}\text{Rotation error} &= \operatorname{acos}\left(\left(\operatorname{trace}(R_{\text{true}}^T R_{\text{est}}) - 1\right)/2\right)\\ \text{Translation error} &= \operatorname{acos}\left(T_{\text{true}}^T T_{\text{est}}\right) / \left(\|T_{\text{true}}\|\|T_{\text{est}}\|\right)\\ \text{Structure error} &= \operatorname{acos}\left(\lambda_{\text{true}}^T \lambda_{\text{est}}\right) / \left(\|\lambda_{\text{true}}\|\|\lambda_{\text{est}}\|\right).\end{aligned}$$

4.1. Experimental Setup

We consider a central panoramic camera given by (1) with $s_x = s_y$ and define $(\xi + m)s_x$ as the focal length. The diameter of the image disk was considered to be 512 pixels. We assume a fixed CCD size for all camera parameters ξ, which corresponds to fixing the image disk to unit radius for all cameras. Since the field of view (FOV) of a central panoramic camera is determined by the back-projection ray at the perimeter of the image disk, the FOV is determined by ξ. Figure 3 shows the experimental setup, from which it is clear that $\text{FOV}(\xi) = \pi + 2\text{atan}(\mathbf{f}_\xi(1,0))$. Notice that $\text{FOV} = 90^\circ$ when $\xi = 0$, and $\text{FOV} = 180^\circ$ when $\xi = 1$. As Figure 3 shows, we also model the fact that catadioptric cameras have a blind spot in the center of the image disk due to the reflection of the CCD in the mirror. In these experiments, we do not image any 3D point whose projection is at a distance less than $r_{\min} = 0.25$ from the image center.

A set of $N = 20$ points was scattered uniformly within the camera field of view at random depths which varied from $Z_{\max} = -10$ to $Z_{\min} = -400$ focal lengths as shown in Figure 3. The points were projected onto the image plane of the base camera frame through the central panoramic projection (2). In each experiment, we used $F = 7$ image frames, where the translation between each camera frame and the base frame was a uniform random variable in the unit sphere, so that the simulation parameter $\tau = \|T\|_{\max}/\lambda_{min}$ is constant over the trials. The rotation between each frame and the base frame was the random variable $\exp([\omega]_\times)$ where $\omega \in \mathbb{R}^3$ was uniform in the unit sphere. The image points in each camera frame were computed using (2) and then adding zero mean Gaussian noise where the standard deviation σ was specified in terms of pixel size and was independent of camera motion. We do not consider the effect of correspondence errors.

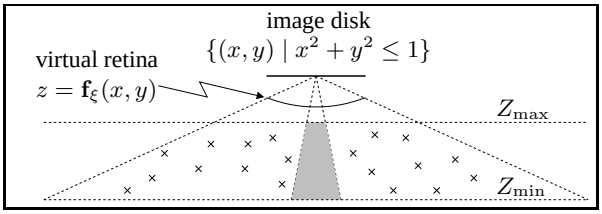

Figure 3: Showing the experimental setup for the simulations. The field of view is determined by ξ through the back-projection ray at the perimeter of the image disk.

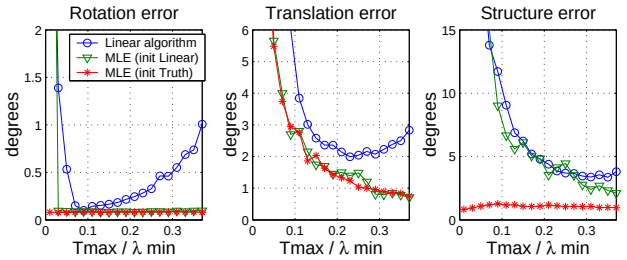

Figure 4: Showing the performance of the linear MFSFM algorithm as a function of baseline. The noise level was $\sigma = 1$ pixel and the camera was paracatadioptric $\xi = 1$.

4.2. Simulation Results

Baseline Dependency. Figure 4 shows the performance the linear MFSFM and the MLE algorithms as a function of the baseline $\|T\|_{\max}/\lambda_{min}$. The noise level was $\sigma = 1$ pixel and the camera was paracatadioptric ($\xi = 1$). We observe that the MFSFM algorithm has very good performance for motion estimation for $0.1 \le \tau \le 0.4$. When $\tau < 0.1$ the signal-to-noise ratio is too small, causing an increase in the estimation error. When $\tau > 0.4$, the small baseline assumption is violated hence the estimation error increases. Notice that we still get good results for $\tau = 0.4$, which corresponds to a relatively large translational motion.

Noise Dependency. Figure 5 shows the performance of the linear MFSFM algorithm compared with the MLE algorithms for different levels of noise. The baseline was fixed at $\tau = 0.2$ and the camera parameter was $\xi = 1$ (paracatadioptric). Notice that rotation estimates were the least sensitive to noise, while structure estimates were the most sensitive. Also, notice that the average error for the MFSFM algorithm appears to grow faster than linearly as a function of the noise level. This is different than the behavior of the corresponding MFSFM algorithm for perspective projection [9, 15] and is a result of the fact that the large field of view in omnidirectional cameras comes at the price of lower image resolution and higher noise sensitivity, due to nonlinear distortion.

Camera Dependency. Figure 6 shows the performance of the linear MFSFM algorithm compared with the MLE algorithm for different camera parameters ξ. The noise level

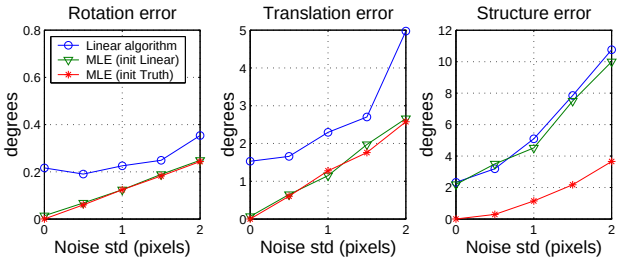

Figure 5: Comparing the performance of the linear MFSFM algorithm and the nonlinear MLE algorithm as a function of noise level. The baseline was $\tau = 0.2$ and the camera was paracatadioptric $\xi = 1$.

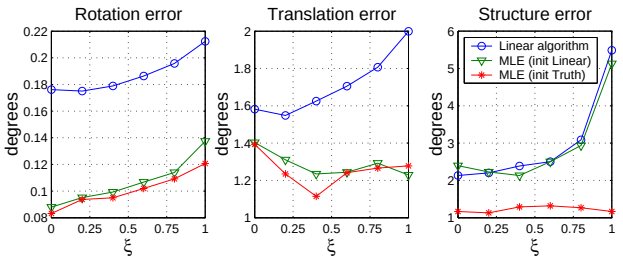

Figure 6: Comparing the performance of the linear MFSFM algorithm and the nonlinear MLE algorithm as a function of camera parameter ξ. The noise level was $\sigma = 1$ pixel and the baseline was $\tau = 0.2$.

was $\sigma = 1$ pixel and the baseline was $\tau = 0.2$. Notice that while the MLE algorithm is relatively insensitive to varying ξ, the linear MFSFM algorithm is much more sensitive to noise as $\xi \rightarrow 1$, or equivalently, the FOV increases. Looking carefully at the plots in Figure 6, we see that the optimal central panoramic camera which minimizes motion estimation error is around $\xi^* \approx 0.2$.

The results of this experiment show a very interesting phenomenon in structure and motion estimation as a function of the FOV of omnidirectional cameras. One one hand, the large FOV of an omnidirectional camera implies that one can more easily distinguish between translational and rotational motion, which should make SFM algorithms more robust in the presence of noise [13]. On the other hand, the effective resolution of the camera decreases as the FOV increases, and hence for a given noise level in the image plane, a larger FOV camera will have larger errors in the triangulation angles and hence larger 3D motion and structure errors. The results of this experiment show that the optimal central panoramic camera which balances these competing factors is achieved around $\xi^* \approx 0.2$.

However, one should not forget that the large FOV in omnidirectional cameras comes with many other benefits, such as being able to track feature points across many more frames than with a small FOV camera.

5. Conclusions

To the best of our knowledge, we have presented the first multi-frame structure from motion algorithm for discrete measurements with central panoramic cameras. Simulation results show that the algorithm performs well in the domain for which it was designed, namely when the translation between image frames is small relative to the scene depth. This assumption is easily satisfied in applications such as outdoor navigation.

Future work will consider relaxing the general motion assumption that the translation be fully three dimensional. While the case of translation in the X-Y plane was easily handled with minor modifications of the general algorithm, the case of planar motion in an arbitrary plane is much more complex as demonstrated in [15]. We will also compare our algorithm with the method of first unwarping the omnidirectional image into a perspective image and then applying algorithm [9]. We suspect that as in [12], the accuracy of reconstruction using the two methods will depend on the type of camera motion.

References

[1] S. Baker and S. Nayar. A theory of single-viewpoint catadioptric image formation. *IJCV*, 35:175–196, 1999.

[2] J. Barreto and H. Araujo. Geometric properties of central catadioptric line images. In *ECCV*, pages 237–251, 2002.

[3] P. Chang and M. Hebert. Omni-directional structure from motion. In *OMNIVIS*, pages 127–133, 2000.

[4] P. Doubek and T. Svoboda. Reliable 3d reconstruction from a few catadioptric images. In *OMNIVIS*, pages 71–78, 2002.

[5] C. Geyer and K. Daniilidis. A unifying theory for central panoramic systems and practical implications. In *ECCV*, pages 445–461, 2000.

[6] C. Geyer and K. Daniilidis. Structure and motion from uncalibrated catadioptric views. In *CVPR*, pages 279–286, 2001.

[7] C. Geyer and K. Daniilidis. Paracatadioptric camera calibration. *IEEE Transactions on PAMI*, 4(24):1–10, 2002.

[8] J. Gluckman and S. Nayar. Ego-motion and omnidirectional cameras. In *ICCV*, pages 999–1005, 1998.

[9] J. Oliensis. A multi-frame structure-from-motion algorithm under perspective projection. *IJCV*, 34(2-3):163–192, 1999.

[10] O. Shakernia, R. Vidal, and S. Sastry. Infinitesimal motion estimation from multiple central panoramic views. In *IEEE Workshop on Motion and Video Computing*, pages 229-234, 2002.

[11] O. Shakernia, R. Vidal, and S. Sastry. Multi-body motion estimation and segmentation from multiple central panoramic views. To appear: *ICRA*, 2003.

[12] O. Shakernia, R. Vidal, and S. Sastry. Omnidirectional egomotion estimation from back-projection flow. In *OMNIVIS*, 2003.

[13] T. Svoboda, T. Pajdla, and V. Hlavac. Motion estimation using panoramic cameras. In *IEEE Conference on Intelligent Vehicles*, pages 335–350, 1998.

[14] R.F. Vassallo, J. Santos-Victor, and J. Schneebeli. A general approach for egomotion estimation with omnidirectional images. In *OMNIVIS*, pages 97–103, 2002.

[15] R. Vidal and J. Oliensis. Structure from planar motions with small baselines. In *ECCV* , pages 383–398, 2002.