Fast central catadioptric line extraction, estimation, tracking and structure from motion

Christopher MEI and Ezio MALIS
INRIA Icare Project-team, Sophia-Antipolis, France
Email : {firstname.lastname@sophia.inria.fr}

Abstract— **In this paper we present an analysis of 3D line projections for central catadioptric cameras from a projective perspective. Most algorithms consider the projection of lines as general conics in the image plane with five degrees of freedom. However, in the calibrated case, only two parameters are needed to represent lines. We describe methods to obtain fast extraction and estimation algorithms. We then explain how classical edge-tracking algorithms can be adapted to these sensors. To this avail, we introduce two parametric equations for lines in central catadioptric images. We then propose a minimal representation for the euclidean transformation in the structure from motion problem and introduce possible metrics between a point and a central catadioptric line. These metrics are evaluated on simulated data. The structure from motion algorithm, from the line extraction process to the tracking and the reconstruction is tested on a real sequence.**

I. Introduction

Panoramic cameras capture a wide field of view that can benefit many robotic applications. Central catadioptric cameras are the specific sensors composed of a mirror and a camera which preserve a single viewpoint [1]. This is a desirable property as it means results from projective geometry can be used directly when the sensor is calibrated. Geyer [2] and Barreto [3] developed a unified projection model for these sensors through the projection of points on the sphere. It has since been used regularly to develop generic algorithms.

In the past, lines have been used extensively with panoramic cameras for the motion estimation and localisation of mobile robots [4], [5] but generally ([6] being an exception) under the assumption that the lines were radially projected in the device. This of course limits the use of the sensor to environments with sufficient vertical lines and imposes the sensor to be in a vertical position. In this article we aim at generalising the use of lines. Work has been done previously for line extraction [7], [8], [9]. Our contribution is to generalise the approach to all central catadioptric sensors from a projective geometry perspective which leads, as we will see, to simple and efficient algorithms. To our knowledge, current line tracking for omnidirectional vision has only been done using quadric approaches such as in [10] or vanishing points [6]. We will see how we can use a parametric approach instead using the minimal amount of parameters.

Structure from motion from lines has been thoroughly studied in the past [11]. More recently, [12] and [13] analyse the non-linear structure from motion equations. In the context of non-linear minimisation, it is desirable to parametrise the problem using the minimum amount of parameters: the minimisation is faster, less subject to noise and consistency constraints can be directly imposed (eg. a rotation matrix must stay a rotation matrix after minimisation). In [13], the authors introduce an orthonormal representation for Plücker coordinates to minimise only the 4 parameters representing a line. They give references to possible methods to obtain a minimal representation of the transformation but do not give specific details. In this article, we put forth the group structure of the line motion matrix [14] that enables the use of Lie algebras for a minimal parametrisation. We detail the specific case of calibrated cameras that applies to central catadioptric sensors. Different point-line distances are proposed.

We will call omni-lines the projection of 3D lines in the image plane (or the normalised plane according to context) to emphasise that they are conics [15] with only two degrees of freedom in the calibrated case.

We will start by reintroducing a slightly modified version of the unified model of Geyer and Barreto. We will then show how the projection model can be re-written to use the properties from *projective geometry* which leads to a linear estimation of the omni-lines and an efficient algorithm for their extraction using hough transforms. The following section details how to obtain equations for line-tracking. This enables in turn automatic structure from motion. In the last section, we apply the described algorithms for the motion estimation of a mobile robot in a situation where point or template-based approaches generally fail.

II. Unified projection model

For sake of completeness, we present here the steps for the projection of 3D points to the image plane for central catadioptric sensors (see [2], [3], [16] for more details):

1) world points in the mirror frame are projected onto the unit sphere S^2, $(\boldsymbol{\mathcal{X}})_{\mathcal{F}_m} \longrightarrow (\boldsymbol{\mathcal{X}}_s)_{\mathcal{F}_m} = \frac{\boldsymbol{\mathcal{X}}}{\|\boldsymbol{\mathcal{X}}\|} = (X_s, Y_s, Z_s)$
2) the points are then changed to a new reference frame centered in $\boldsymbol{\mathcal{C}}_p = (0, 0, \xi)$, $(\boldsymbol{\mathcal{X}}_s)_{\mathcal{F}_m} \longrightarrow (\boldsymbol{\mathcal{X}}_s)_{\mathcal{F}_p} = (X_s, Y_s, Z_s - \xi)$
3) we then project the point onto the normalised plane, $\mathbf{m} = (x, y, 1) = (\frac{X_s}{Z_s - \xi}, \frac{Y_s}{Z_s - \xi}, 1) = \hbar(\boldsymbol{\mathcal{X}}_s)$
4) the final projection involves a generalized camera projection matrix $\mathbf{K}$: $\mathbf{p} = (u, v, 1) = \mathbf{K}\mathbf{m}$

The function $\hbar$ is bijective and

$$\hbar^{-1}(\mathbf{m}) = \begin{bmatrix} \frac{-\xi-\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}x \\ \frac{-\xi-\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}y \\ \frac{-\xi-\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}+\xi \end{bmatrix} \quad (1)$$

III. Omni-lines as projection of planes

A 3D line projected in a monocular imaging device can be parametrised by the normal noted $\mathbf{n}$ ($\mathbf{n} \in S^2$) formed by the line and the center of projection (Fig. 1).

Equation (1) relates a point on the normalised plane to a point $\boldsymbol{\mathcal{X}}_s$ on the sphere or projectively to the ray through the mirror center and $\boldsymbol{\mathcal{X}}_s$. Thus, by multiplying by $\frac{x^2+y^2+1}{-\xi-\sqrt{1+(1-\xi^2)(x^2+y^2)}} \neq 0$, we obtain the projective equality:

$$\begin{cases} \hbar^{-1}(\mathbf{m}) \sim \begin{bmatrix} x \\ y \\ f(x,y) \end{bmatrix} \\ f(x,y) = 1 + \xi\frac{x^2+y^2+1}{-\xi-\sqrt{1+(1-\xi^2)(x^2+y^2)}} \end{cases} \quad (2)$$

Equation (2) is valid for any central catadioptric device (it is also valid for an approximation of a non-central projective model). The special case $f(x,y) = 1$ corresponds to the classic perspective projection. When the sensor is calibrated the values for $f(x,y)$ can be pre-calculated and stored in a look-up table to improve the efficiency of the lifting of the points. The relation between $\mathbf{m}$ and $\mathbf{p}$ is linear and not very costly to compute (in particular if $r = 1$ and $s = 0$ which is often the case with modern cameras).

A point $\mathbf{p}$ on an omni-line of parameter $\mathbf{n}$ verifies:

$$\mathbf{n}^\top \begin{bmatrix} x \\ y \\ f(x,y) \end{bmatrix} = 0 \quad (3)$$

A. Line estimation

If we write (3) for n points, we obtain:

$$\begin{bmatrix} x_1 & y_1 & f(x_1,y_1) \\ x_2 & y_2 & f(x_2,y_2) \\ \vdots & \vdots & \vdots \\ x_n & y_n & f(x_n,y_n) \end{bmatrix} \mathbf{n} = \mathbf{A}\mathbf{n} = 0 \quad (4)$$

If we consider the singular value decomposition of $\mathbf{A}$, $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$ and order the eigenvalues of $\mathbf{S}$ in decreasing order, the normalisation of the third column of $\mathbf{V}$ will correspond to the least squares solution to (4). This is an alternative to the method suggested by Barreto [8]. It has the advantage of also being valid for hyperbolic mirrors.

B. Line extraction with the classic Hough transform

Let Φ be the colatitude and Θ the azimuthal angle, the normal can be written in spherical coordinates as:

$$\begin{cases} n_x = \sin\Phi\cos\Theta \\ n_y = \sin\Phi\sin\Theta \\ n_z = \cos\Phi \end{cases}$$

if we assume that $n_z \neq 1$ (ie. $\Phi \neq 0[\pi]$) and note $z = f(x,y)$, from (3) we obtain:

$$\Phi = \mathrm{atan}\left(\frac{x\cos(\Theta) + y\sin(\Theta)}{z}\right) \quad (5)$$

This result was proposed previously in [9]. We discuss a way to adapt the line extraction to the non-uniform pixel resolution in Section III-D. It may also be noted that values for atan cannot be pre-calculate (because the input space $\mathbb{R}$ is not bounded). To improve the efficiency, a Hough space can be built using directly $\tan(\Phi)$ with for example a "linked-list" to represent the Hough space (see [17] for details of different structures related to Hough parameter spaces).

C. Line extraction with the randomized Hough transform

The Randomized Hough transform (RHT) [17] has proved to be an efficient and robust alternative to classic Hough. It shares *convergence mapping* with RANSAC, meaning that we estimate the n parameters of the curve function ($n = 2$ for a line) by randomly extracting n values.

For estimating the parameters of a line, the authors in [17] extend the (ρ, θ) parametrisation from Duda and Hart. A natural and more efficient parametrisation can be obtained by directly estimating the normal: the omni-line joining two points $\mathbf{m}_1$ and $\mathbf{m}_2$ has for normal $\mathbf{n}$ ($\hbar^{-1}(\mathbf{m}) \in S^2$):

$$\mathbf{n} = \hbar^{-1}(\mathbf{m}_1) \times \hbar^{-1}(\mathbf{m}_2) \quad (6)$$

By imposing for example $n_z \geq 0$, we obtain a 2-dimensional buffer in (n_x, n_y).

D. Voting in the Hough space

The solid angle subtended by the surface represented by a pixel can be used to take into account the non-uniform resolution. A pixel $\mathbf{p}$ is bounded by $(u-0.5, v-0.5)$, $(u+0.5, v-0.5)$, $(u-0.5, v+0.5)$ and $(u+0.5, v+0.5)$. The corresponding surface on the unit sphere is then bounded by (Φ_{min}, Φ_{max}) and $(\Theta_{min}, \Theta_{max})$ (obtained through the lifting of the points) and corresponds to the following solid angle s (measured in steradians *sr*):

$$\begin{aligned} s &= \int_{\Theta_{min}}^{\Theta_{max}} \int_{\Phi_{min}}^{\Phi_{max}} \sin(\Phi) d\Theta d\Phi & (7) \\ &= -(\Theta_{max} - \Theta_{min})(\cos(\Phi_{max}) - \cos(\Phi_{min})) & (8) \end{aligned}$$

The precision of the normal estimate will be inversely proportional to the subtended solid angle so $1/s$ can be used in the classic Hough voting scheme. (In the randomized case, we assumed that on average the surface subtended by two pixels was the same.)

IV. Line tracking

Tracking a line $\mathcal{L}$ between two views can be done using classic edge-tracking approaches [18] [19] in the following steps:

1) obtain n points on $\mathcal{L}$ uniformly distributed in the image,
2) for each point calculate the normal to the edge,

3) search (within pre-defined bounds) along the direction given by the normal to the curve for edge points with same normals (using pre-calculated convolution kernels),
4) robustly extract the equation of the new line from the edge points.

For 3) the preferred method is the Bresenham algorithm [20] For 4), M-estimators are often chosen to extract the parameters but in presence of a lot of noise, a RANSAC is a good alternative (this is relatively fast as the model is simple to fit using (6) and the size of the data is small.

We will now derive a parametric equation for omni-lines and calculate the normal in a given point. We will see that using a conic parametric function leads to singularities (but gives information on the nature of the conic). We will then propose a non-singular parametric function.

A. Conic parametric function

Equation (6) can be re-written to obtain a quadric form in the normalised plane [21][15]: $\mathbf{m}^\top \boldsymbol{\Omega}_m \mathbf{m}$ with:

$$\boldsymbol{\Omega}_m = \begin{bmatrix} n_x^2(1-\xi^2) - n_z^2\xi^2 & n_x n_y(1-\xi^2) & n_x n_z \\ n_x n_y(1-\xi^2) & n_y^2(1-\xi^2) - n_z^2\xi^2 & n_y n_z \\ n_x n_z & n_y n_z & n_z^2 \end{bmatrix} \quad (9)$$

$\det(\widehat{\boldsymbol{\Omega}}_m) = \xi^4 n_z^4(n_x^2 + n_y^2 + n_z^2) = (\xi n_z)^4$. For $\widehat{\boldsymbol{\Omega}}_m$ to be a proper conic, we will from now on assume that $\xi \neq 0$ (non-planar/perspective mirror) and $n_z \neq 0$.

The nature of the conic depends on the number of intersections with the line at infinity ie. the sign of $\Delta = 1 - \xi^2 - n_z^2$ (we removed $n_z^2\xi^2 > 0$). $\Delta > 0$ corresponds to a hyperbola, $\Delta = 0$ to a parabola and $\Delta < 0$ to an ellipse.

From the *Joachimsthal* equations, we obtain the four focal points (2 real and 2 complex) [22]. The two real values are:

$$\mathbf{f}_1 = \begin{bmatrix} n_x \\ n_y \\ n_z + \sqrt{1-\xi^2} \end{bmatrix} \quad \mathbf{f}_2 = \begin{bmatrix} n_x \\ n_y \\ n_z - \sqrt{1-\xi^2} \end{bmatrix} \quad (10)$$

(We can note that $\mathbf{f}_2$ is at infinity if the conic is a parabola.) If we now center the conic in $\mathbf{f}_1$ and rotate it (if $n_z \neq 1$) by an angle Θ, we obtain ($\Delta \geq -1$):

$$\boldsymbol{\Omega}'_m = \begin{bmatrix} \Delta & 0 & \sqrt{(1-n_z^2)(1-\xi^2)} \\ 0 & -\xi^2 n_z^2 & 0 \\ \sqrt{(1-n_z^2)(1-\xi^2)} & 0 & 1 \end{bmatrix} \quad (11)$$

Let $x = \rho\cos(\theta)$ and $y = \rho\sin(\theta)$, the polar equation of the omni-line centered in $\mathbf{f}_1$, valid for $n_z \neq 0$ and $\xi \neq 0$, is:

$$\rho = \frac{1}{\xi n_z - \sqrt{(1-n_z^2)(1-\xi^2)}\cos(\theta - \Theta)} \quad (12)$$

For $n_z = 0$, the conic is a straight line that goes through the origin and is parameterised by (ρ, Θ).

When $n_z \to 0$, we get closer to a degenerate conic as $\mathbf{f}_1 \to 0$. This means we will not be able to represent and sample curves when $n_z \to 0$ using the angle θ.

B. Unified non-singular parametric function

Let $B = \{\boldsymbol{\mathcal{X}}_s = (X_s, Y_s, \xi) | \boldsymbol{\mathcal{X}}_s \in S^2\}$. B is the natural boundary between the two sheets of S^2 covering $\mathbb{P}^2$ through the unified projection.

Let $\mathcal{C}$ be the arc of the great circle corresponding to $\mathcal{L}$ and parametrised by $\mathbf{n}$. $\mathbf{n}$ can be seen as an axis of rotation for the points of $\mathcal{C}$ on the sphere (Fig. 1). Let $\mathbf{w}$ be one of these points ($\mathbf{n}^\top \mathbf{w} = 0$). $\mathcal{C}$ can be parametrised with an angle θ without a singularity using Rodrigues' formula:

$$\mathbf{w}(\theta) = e^{[\mathbf{n}]_\times \theta}\mathbf{w} \quad (13)$$

We do not obtain a singularity because a finite 3D line spans an angle strictly inferior to π.

$\mathbf{w}(\theta)$ is a point on the sphere so its projection $\mathbf{m}(\theta)$ (defined for $\mathbf{m}(\theta) \notin B$) on the normalised plane is simply:

$$\mathbf{m}(\theta) = \frac{1}{\mathbf{w}_z(\theta) - \xi} \begin{bmatrix} \mathbf{w}_x(\theta) \\ \mathbf{w}_y(\theta) \end{bmatrix} \quad (14)$$

C. Curve sampling

Let s be the arc length of the parametric curve (14) between two points $\mathbf{m}(\theta_1)$ and $\mathbf{m}(\theta_2)$ with $\theta_2 > \theta_1$. We will make the assumption that we have a similarity between the normalised plane and the image plane (ie. $r \approx 1$ and $s \approx 0$). (In other words a uniform sampling in the normalised plane corresponds to a uniform sampling in the image plane.)

If we wish to sample the curve in n values (to guarantee constant time), the increment arc length is $\delta s = \frac{s}{n}$ with $s = \int_{\theta_1}^{\theta_2} ds$. If we wish to obtain values separated by p pixels, the increment is $\delta s = \frac{p}{\gamma}$.

In the general case, the calculation of arc lengths for conics involve elliptic integrals of the second kind, so we cannot obtain a simple formulation for s. We may note that in the case of a paraboloid mirror, the conic is simply a circle which can be uniformly sampled by an angular increment. However we loose this property with the non-singular representation.

For an approximate calculation of s, we may use the differential form of the curve length and a small increment for θ:

$$ds = \sqrt{dx'^2 + dy'^2}d\theta \quad (15)$$

$$\text{with : } \left\{ \begin{array}{ccc} dx' & = & \frac{\mathbf{w}'_x(\theta)(\mathbf{w}_z(\theta)-\xi)-\mathbf{w}_x\mathbf{w}'_z(\theta)}{(\mathbf{w}_z(\theta)-\xi)^2} \\ dy' & = & \frac{\mathbf{w}'_y(\theta)(\mathbf{w}_z(\theta)-\xi)-\mathbf{w}_y\mathbf{w}'_z(\theta)}{(\mathbf{w}_z(\theta)-\xi)^2} \\ \mathbf{w}'(\theta) & = & e^{[\mathbf{n}]_\times \theta}[\mathbf{n}]_\times \mathbf{w} \end{array} \right. \quad (16)$$

D. Normal to an omni-line

The angle ϕ of the normal in $\mathbf{m}(\theta)$ is simply:

$$\phi = \left\{ \begin{array}{ll} \text{atan}\,(-\frac{dx'}{dy'}) & \text{if } dy' \neq 0 \\ \frac{\pi}{2} & \text{if } dy' = 0 \end{array} \right. \quad (17)$$

V. Structure from motion

Points and templates are generally chosen for motion estimation in robotic application because they offer robust and accurate results. However in cases of low-textured environments, lines can play a key role to improve estimates and provide a partial 3D reconstruction.

In real-time robotic applications, iterative approaches are generally preferred to batch algorithms as they are often faster. However in the case of lines that have been automatically extracted, the motion and 3D line estimates are generally not well constrained and sensitive to the 3D position of the lines. On the other hand, the minimisation is not computationally expensive which encourages a two step approach: 1) bundle adjustment 2) global filtering (eg. Extended Kalman Filter) when the covariance enables confidence to be put into the line and motion estimates. In this article, we focus on bundle adjustment.

A. Line representation

We will represent 3D lines by the following Plücker coordinates: $\mathcal{L}^\top \sim \begin{bmatrix} \mathbf{n}^\top & \mathbf{v}^\top \end{bmatrix}$. Plücker coordinates are defined up to a scale factor. We choose to normalise the first component to simplify the equations on the sphere. $\mathbf{n}$ is defined as previously in the article and $\mathbf{v}$ is the direction of the 3D line. In order to obtain a valid line representation, the constraint $\mathbf{n}^\top\mathbf{v} = 0$ must be imposed.

B. Minimal representation of Plücker coordinates

We will use the method proposed by Bartoli and Sturm [13] to obtain a minimal representation (4 parameters) of Plücker coordinates through an orthonormal representation.

Let QR be the orthogonal/upper triangular decomposition. To briefly summarise, Plücker coordinates can be decomposed as:

$$\begin{bmatrix} \mathbf{n} & \mathbf{v} \end{bmatrix}_{3\times 2} \overset{QR}{=} \mathbf{U}_{3\times 3} \begin{bmatrix} \sigma_1 & \\ & \sigma_2 \\ & \end{bmatrix}_{3\times 2}, \mathbf{W} = \frac{1}{\|\sigma\|} \begin{bmatrix} \sigma_1 & -\sigma_2 \\ \sigma_2 & \sigma_1 \end{bmatrix} \tag{18}$$

with $(\mathbf{U}, \mathbf{W}) \in \mathbb{SO}(3) \times \mathbb{SO}(2)$. Let $\mathbf{x}_L$ contain the 3+1 parameters representing the matrices, we can recover the Plücker coordinates through (with $\mathbf{u}_i$ the $i-$th column of $\mathbf{U}$):

$$\mathcal{L}(\mathbf{U}(\mathbf{x}_L), \mathbf{W}(\mathbf{x}_L))^\top \rightarrow \begin{bmatrix} \mathbf{u}_1^\top & \frac{w_{21}}{w_{11}}\mathbf{u}_2^\top \end{bmatrix} \tag{19}$$

C. Line motion matrix

In [14], the authors define the 6×6 matrices that act on Plücker coordinates in projective, affine and Euclidean spaces. We can prove that we have in fact a group homomorphism between the transformation groups and the line motion matrix spaces for the matrix product. This result is important as it indicates that we can obtain a minimal representation through the associated Lie algebras and recover the transformations directly. We will detail the Euclidean case which is of interest for this study.

For calibrated central catadioptric cameras, the transformation between 3D lines in two views can be represented by a rotation matrix $\mathbf{R} \in \mathbb{SO}(3)$ and a translation $\mathbf{t} \in \mathbb{R}^3$ by:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & [\mathbf{t}]_\times \mathbf{R} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \tag{20}$$

We will call $\mathbb{LE}(3)$ the group formed of matrices of the previous type and $\mathfrak{le}(3)$ its associated Lie algebra. Let $\mathbf{A}_i$, with $i \in \{1, 2, ..., 6\}$, be a basis of $\mathfrak{le}(3)$. Any matrix $\mathbf{A} \in \mathfrak{le}(3)$ can be written as a linear combination of the matrices $\mathbf{A}_i$.

Let the (3×1) vectors $\mathbf{b}_x = (1, 0, 0)$, $\mathbf{b}_y = (0, 1, 0)$ and $\mathbf{b}_z = (0, 0, 1)$ be the natural orthonormal basis of $\mathbb{R}^3$. The $\mathbf{A}_i$ matrices are of dimension (6×6). The generators for the translation are:

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{0} & [\mathbf{b}_x]_\times \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \mathbf{A}_2 = \begin{bmatrix} \mathbf{0} & [\mathbf{b}_y]_\times \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \mathbf{A}_3 = \begin{bmatrix} \mathbf{0} & [\mathbf{b}_z]_\times \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{21}$$

The generators for the rotation are:

$$\mathbf{A}_4 = \begin{bmatrix} [\mathbf{b}_x]_\times & \mathbf{0} \\ \mathbf{0} & [\mathbf{b}_x]_\times \end{bmatrix}, \mathbf{A}_5 = \begin{bmatrix} [\mathbf{b}_y]_\times & \mathbf{0} \\ \mathbf{0} & [\mathbf{b}_y]_\times \end{bmatrix}, \mathbf{A}_6 = \begin{bmatrix} [\mathbf{b}_z]_\times & \mathbf{0} \\ \mathbf{0} & [\mathbf{b}_z]_\times \end{bmatrix} \tag{22}$$

The exponential map links the Lie algebra to the Lie Group. $\mathbf{T}$ can be locally parameterized as:

$$\mathbf{T}(\mathbf{x}_T) = \exp\left(\sum_{i=1}^{6} x_i \mathbf{A}_i\right) \tag{23}$$

Thanks to the group homomorphism we can recover the euclidean transformation $\mathbf{T}_e$ directly from x_T through the 6 generators $\mathbf{B}_i$ of $\mathbb{SE}(3)$:

$$\mathbf{T}_e(\mathbf{x}_T) = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} = \exp\left(\sum_{i=1}^{6} x_i \mathbf{B}_i\right)$$

D. Distance functions

Several distance functions between a point $\boldsymbol{\mathcal{X}}_s$ and a line parametrised by $\mathbf{n}$ can be considered in the case of the sphere:

$$\begin{cases} d_A = \boldsymbol{\mathcal{X}}_s^\top \mathbf{n} \\ d_R = \arccos(\sqrt{1 - (\boldsymbol{\mathcal{X}}_s^\top \mathbf{n})^2}) \\ d_r = d_e(\hbar(\boldsymbol{\mathcal{X}}_s^\perp), \hbar(\boldsymbol{\mathcal{X}}_s)) \\ \boldsymbol{\mathcal{X}}_s^\perp = \frac{\boldsymbol{\mathcal{X}}_s - (\boldsymbol{\mathcal{X}}_s^\top \mathbf{n})\mathbf{n}}{\sqrt{1 - (\boldsymbol{\mathcal{X}}_s^\top \mathbf{n})^2}} \end{cases} \tag{24}$$

$\boldsymbol{\mathcal{X}}_s^\perp$ is the closest point to the line defined by $\mathbf{n}$ from $\boldsymbol{\mathcal{X}}_s$ (Fig. 1). d_A is an algebraic distance. d_R is the distance on the sphere (Riemann distance) between $\boldsymbol{\mathcal{X}}_s^\perp$ and $\boldsymbol{\mathcal{X}}_s$. d_r is the distance between $\boldsymbol{\mathcal{X}}_s$ and $\boldsymbol{\mathcal{X}}_s^\perp$ reprojected onto the normalised image plane. It corresponds to the standard distance between a point and a line in the perspective case if $n_z = 0$.

E. Global cost function

Let $\mathbf{P}$ be the projection matrix for the 3D lines $\mathbf{P} = \begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} \end{bmatrix}$

Let $\boldsymbol{\mathcal{X}}_s^{ij}$ and be $\boldsymbol{\mathcal{Y}}_s^{ij}$ be the two endpoints of the j-th lines in the i-th view. The cost function can be written as:

$$c_{ij}(\mathbf{x}_T^i, \mathbf{x}_L^j) = \left(d_.(\boldsymbol{\mathcal{X}}_s^{ij}, \mathbf{x}_T^i, x_L^j)\right)^2 + \left(d_.(\boldsymbol{\mathcal{Y}}_s^{ij}, \mathbf{x}_T^i, \mathbf{x}_L^j)\right)^2 \tag{25}$$

Let $\widehat{\mathbf{T}}$ be an approximation of the real transformation and $(\widehat{\mathbf{U}}, \widehat{\mathbf{W}})$ an approximation of the line parameters. In the case of the algebraic distance, for example, the problem is to find

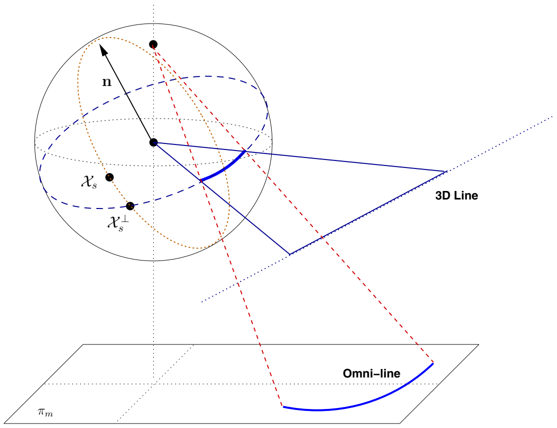

Fig. 1. Closest point to a great circle on the sphere

the incremental transformations $\mathbf{T}(\mathbf{x}_T)$, $\mathbf{U}(\mathbf{x}_L)$ and $\mathbf{W}(\mathbf{x}_L)$ such that the following value is minimised:

$$d_A(\boldsymbol{\mathcal{X}}_s, \mathbf{x}_T, \mathbf{x}_L) = \boldsymbol{\mathcal{X}}_s^\top \mathbf{P}\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}_T)\mathcal{L}(\widehat{\mathbf{U}}\mathbf{U}(\mathbf{x}_L), \widehat{\mathbf{W}}\mathbf{W}(\mathbf{x}_L)) \tag{26}$$

The global cost function is then for m views and l lines (with $\mathbf{X}_T$ and $\mathbf{X}_L$ the list of the parameters and $\mathbf{x}_T^1 = \mathbf{0}$) :

$$F(\mathbf{X}_T, \mathbf{X}_L) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{l} \|c_{ij}(\mathbf{x}_T^i, \mathbf{x}_L^j)\|^2 \tag{27}$$

The equation has $6(m-1)+4l$ unknowns. Each line in a given view adds 2 constraints. Therefore the minimal number of lines needed to constrain the system can be deduced from (if we consider that each line is visible in each view):

$$l \geq \frac{6(m-1)}{2m-4} \tag{28}$$

VI. Experimental results

A. Simulated data

Our experimental setup consists of a parabolic mirror ($\xi = 1$) with a generalised focal length of $\gamma = 270$ (this value was chosen from a real camera). Lines were randomly generated at a distance of the camera between 0 and 8 m. The images were spaced by a random transformation with a translation between $[0; 10]$ cm and a rotation between $[0; \pi/2]$ rad to simulate an incremental motion. We added Gaussian noise to the end-points of each line projected in the image. The given values are the mean over 40 trials. The aim of these experiments was to assess the quality of the distances on the sphere. We also wanted to answer the questions: is it better to have a lot of lines with few images or a lot of images with few lines ? (ie. the trade-off between frame rate and the processing time taken for the line extraction and tracking)

Figure 2 shows the effect of errors in the image on the estimation of the translation for 10 lines seen in 15 images (the rotation gave similar results). The reprojection distance d_r gave a far better accuracy than the Riemann d_R and algebraic d_A distances with similar results.

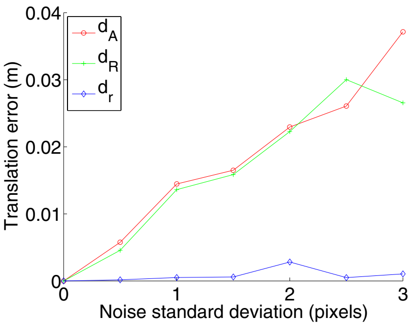

Fig. 2. Translation error for different distances when varying the added noise on the line end-points

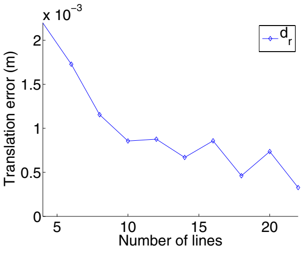

Fig. 3. Translation error when varying the number of lines

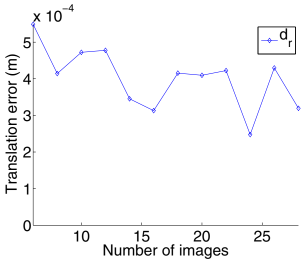

Fig. 4. Translation error when varying the number of images

Figure 3 shows the effect of the number of lines over the quality of the estimation for a fixed noise of 2 pixels and 10 images (only d_r is shown, the other distances gave similar results). The number of lines has a strong influence over the accuracy of the estimates. Figure 4 assesses the improvement in accuracy as the amount of images increases for 3 lines and a fixed noise of 2 pixels. The number of images only improves the estimates slightly. For robotic applications, these results indicate that it might be preferable to estimate and track as many lines as possible rather than obtain many images (with for example a high frame rate). (This was coherent with (28).)

B. Real data

1) Technical details: The RHT was used for the line extraction. In the case of non-perspective omnidirectional sensors, the large field of view and the relative low resolution means that we obtain strong gradient responses typically around the mirror border. To improve the omni-line extraction, the voting scheme used a measure of confidence based on the expected gradient direction (from (3) and (16)) and the observed gradient direction in the image.

For the tracking, after a search along the normals, the omni-line parameters were extracted with RANSAC followed by a least-square minimisation (4).

To avoid omni-lines “jumping” between two potential lines, we only considered lines with relatively few outliers ($\sim 40\%$) and with “enough” supporting points.

2) Experiment: The validation was done on a sequence of 35 images where point or template-based approaches gave unsatisfying results. The sensor used is a parabolic mirror with a telecentric lens and a perspective camera of resolution 1280×1024^1. The motion was constrained in a plane by only estimating 1 rotation and 2 translation parameters in the Lie algebra. The initial values given were the identity for the transformations and the cross product between $\mathbf{n}$ and a point $\boldsymbol{\mathcal{X}}_s$ for the second component of the Plücker coordinates. The initial pixel reprojection error was of 36.3 pixels. After minimisation of the cost function with the Levenberg-Marquardt algorithm, it was reduced to 0.86 pixels.

Figure 5 shows the first and last image of the corridor sequence (the images were flipped to ease the comparison with the 3D model). No new lines were added during the tracking. Not all lines could be tracked through the entire sequence. Figure 6 shows two views of the reconstructed scene with the robot motion. Without being entirely satisfying, the results are sufficiently accurate ($\sim$ 5 cm over 1 m) to use in combination with, for example, odometry.

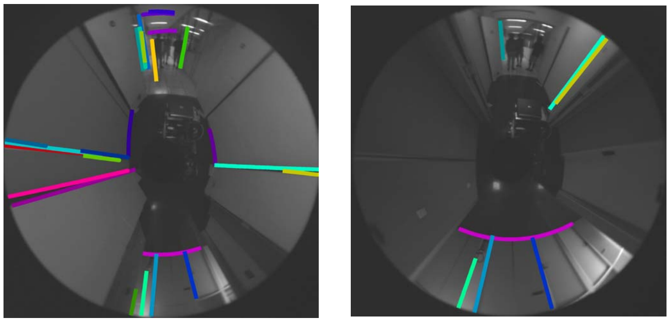

Fig. 5. First and last image of the corridor sequence

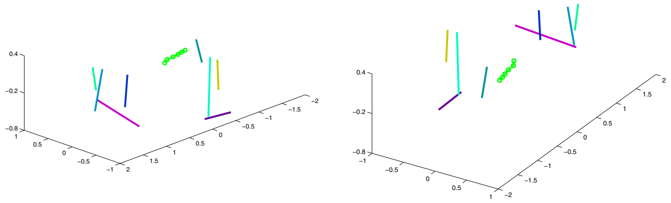

Fig. 6. Two views of the 3D reconstruction of the scene with the robot motion depicted by the green line with circles

VII. Conclusion

In this article we have presented algorithms for automatic structure from motion from lines for central catadioptric sensors. We focused on important aspects for robotic applications such as robustness and minimal parametrisation. From our experimental and simulation results, we do not believe lines constrain the motion sufficiently to be used alone. They can however provide additional robustness to mapping and navigational tasks in low-textured man-made environments.

This work has implications in visual servoing from lines. In [23], the authors base their study on the assumption that omnilines would be extracted and tracked as conics. As we have seen, the tracking can be done through the parametrisation of lines by their normals in a given view. This indicates that there is no formal difference between the central catadioptric case and the perspective case when using projective properties. Through the Plücker coordinates, the results of studies such as [24] can be applied as such.

Acknowledgments

The authors would like to thank Eric Marchand and Andrew Comport for their advice on edge tracking.

References

[1] S. Baker and S. Nayar, "A theory of catadioptric image formation," in *ICCV*, 2001, pp. 422–427.

[2] C. Geyer and K. Daniilidis, "A unifying theory for central panoramic systems and practical implications," in *ECCV*, 2000, pp. 445–461.

[3] J. P. Barreto and H. Araujo, "Issues on the geometry of central catadioptric image formation," in *CVPR*, vol. 2, 2001, pp. 422–427.

[4] Y. Yagi and M. Yachida, "Real-time generation of environmental map and obstacle avoidance using omnidirectional image sensor with conic mirror," in *CVPR*, 1991.

[5] L. Delahoche, C. Pgard, B. Marhic, and P. Vasseur, "A navigation system based on an omnidirectional vision sensor," in *IROS*, 1997.

[6] M. Bosse, R. Rikoski, J. Leonard, and S. Teller, "Vanishing points and 3d lines from omnidirectional video," in *ICIP*, 2002.

[7] Y. Yamazawa, K.and Yagi and M. Yachida, "3d line segment reconstruction by using hyperomni vision and omnidirectional hough transforming," in *ICPR*, 2000.

[8] J. P. Barreto and H. Araujo, "Direct least square fitting of paracatadioptric line images," in *OMNIVIS*, 2003.

[9] P. Vasseur and E. M. Mouaddib, "Central catadioptric line detection," in *BMVC*, 2004.

[10] F. M. J. P. Barreto and R. Horaud, "Visual servoing/tracking using central catadioptric cameras," in *ISER*, ser. Advanced Robotics Series, 2002.

[11] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[12] C. J. Taylor and D. J. Kriegman, "Structure and motion from line segments in multiple images," *PAMI*, vol. 17, no. 11, pp. 1021–1032, 1995.

[13] A. Bartoli and P. Sturm, "Structure from motion using lines: Representation, triangulation and bundle adjustment," *CVIU*, vol. 100, no. 3, pp. 416–441, dec 2005.

[14] ——, "The 3d line motion matrix and alignement of line reconstructions," *IJCV*, vol. 57, no. 3, pp. 159–178, 2004.

[15] J. P. Barreto, "General central projection systems, modeling, calibration and visual servoing," Ph.D. dissertation, University of Coimbra, 2003.

[16] C. Mei, S. Benhimane, E. Malis, and P. Rives, "Homography-based tracking for central catadioptric cameras," in *IROS*, 2006.

[17] L. Xu and E. Oja, "Randomized hough transform (rht) : Basic mechanisms, algorithms, and computational complexities," *CVGIP*, vol. 57, no. 2, pp. 131–154, 1993.

[18] E. Marchand and F. Chaumette, "Feature tracking for visual servoing purposes," *Robotics and Autonomous Systems*, vol. 52, no. 1, pp. 53–70, 2005.

[19] T. D. P. Smith and R. Cipolla, "Layered motion segmentation and depth ordering by tracking edges," *PAMI*, vol. 24, no. 4, pp. 479–494, 2004.

[20] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IDM Systems Journal*, vol. 4, no. 1, 1965.

[21] C. Geyer and K. Daniilidis, "Catadioptric projective geometry," *IJCV*, vol. 45, no. 3, pp. 223–243, 2001.

[22] J. Semple and G. Kneebone, *Algebraic Projective Geometry*. Clarendon Press, 1979.

[23] Y. Mezouar, H. Haj Abdelkader, P. Martinet, and F. Chaumette, "Central catadioptric visual servoing from 3d straight lines," in *IROS*, 2004.

[24] N. Andreff, B. Espiau, and R. Horaud, "Visual servoing from lines," *IJRR*, vol. 21, no. 8, pp. 679–700, 2002.

[1] the toolbox used for the calibration is available as open-source software on the author's website