3D model based tracking for omnidirectional vision: a new spherical approach

Guillaume Caron[1], El Mustapha Mouaddib[1] and Eric Marchand[2]

[1] *Université de Picardie Jules Verne, MIS laboratory, Amiens, France*
{*guillaume.caron, mouaddib*}*@u-picardie.fr*
[2] *Université de Rennes 1, IRISA/INRIA Lagadic, Rennes, France*
eric.marchand@irisa.fr

Abstract

The current work addresses the problem of 3D model tracking in the context of monocular and stereo omnidirectional vision in order to estimate the camera pose. To this end, we track 3D objects modeled by line segments because the straight line feature is often used to model the environment. Indeed, we are interested in mobile robot navigation using omnidirectional vision in structured environments. In the case of omnidirectional vision, 3D straight lines are projected as conics in omnidirectional images. Under certain conditions, these conics may have singularities.

In this paper, we present two contributions. We, first, propose a new spherical formulation of the pose estimation withdrawing singularities, using an object model composed of lines. The theoretical formulation and the validation on synthetic images show thus the new formulation clearly outperforms the former image plane one. The second contribution is the extension of the spherical representation to the stereovision case. We consider in the paper a sensor which combines a camera and four mirrors. Results in various situations show the robustness to illumination changes and local mistracking. As a final result, the proposed new stereo spherical formulation allows to localize online a robot indoor and outdoor whereas the classical formulation fails.

Keywords:
Omnidirectional vision, stereovision, spherical optimization, tracking

1. Introduction

Pose estimation can be done using image features, *i.e.* points, lines, or other geometrical features. When considering the availability of a partial 3D model of an object or a scene, these features need to be extracted from images and associated to the 3D model. This association process is generally done in a first image and features are then tracked over time, while the object and/or the camera is moving.

When using a perspective camera, the spatial volume in which the object can be moved, and still be perceived by the camera, is limited. Hence, objects can appear partially in images. This is particularly the case when the camera is moving within an environment of which a 3D model, even partial, is tracked.

Omnidirectional cameras, thanks to their very wide field of view, allow to track image features during a long period of time, when moving. This is synonym of efficiency, precision and robustness for the tracking and pose estimation processes. Using such a camera allows to keep the object in the image even if it moves in a wide spatial volume surrounding the camera.

A complementary way to increase robustness in a tracking and pose estimation process is to add perception redundancy using several cameras acquiring synchronously images of the object. An interesting idea is thus to use a vision sensor merging stereovision and omnidirectional vision. Different omnidirectional stereovision sensors were designed [1, 2, 3, 4]. We propose to use the *Four On One* sensor (FOO), a catadioptric sensor made of a unique camera and four parabolic mirrors placed in a square at an equal distance from the camera [5, 6] (Fig. 1).

The pose estimation problem using 2D-3D correspondences can be tackled by linear formulations for various features: points [7, 8, 9], lines [10, 11], etc. A problem raised by these formulations is to deal with outliers. They appear in presence of partial occlusions or specular reflections. RANSAC [12] techniques allow to reject outliers in an iterative procedure. Although efficient to reject outliers it is not really well suited to handle noise in image measurements. Recent works affect a weight to image measurements in linear methods to solve this issue [13, 14]. However, these methods currently do not deal with outliers as efficiently as non-linear optimization methods to compute the pose.

The non-linear formulation of the pose computation issue leads to the usual optimization methods such as steepest descent, Gauss-Newton or

Levenberg-Marquardt. These methods consider first order derivatives of a cost function formalizing the error in the image between the observation and the projection of some model for a given pose. The main idea is to align the forward projection of the model to image measurements. Some more recent works use derivatives of the non-linear function of a higher order [15]. Several works about non-linear pose computation has been proposed in the optimization community [16, 17, 18]. As well as in the perspective vision community, many works exist [19, 20, 21], where the main differences rely on the rotation parameterization (Euler angles, quaternion, vector-angle). Reasoning on the velocities of camera pose elements, the virtual visual servoing (VVS) [22, 23], is a full scale non-linear optimization technique which can be used for pose computation [24].

Some works have been done about 3D model based tracking and pose estimation using VVS, for perspective camera [24], stereo perspective rig [25] and even for monocular omnidirectional vision [26, 27]. These works also deal with corrupted data, which is frequent in image feature extraction, using the widely accepted statistical techniques of robust M-estimation [28], giving a weight, dynamically computed, that reflects the confidence we have in each measure [24, 27, 29].

As we are interested by indoor and outdoor mobile robot navigation, we choose to consider straight lines as features because they are frequent (doors, corridors, ...) and an omnidirectional vision sensor to recover the robot pose. The main problem in the case of omnidirectional vision, is that 3D straight lines are projected as conics in the omnidirectional image. This property induces two issues: the conics extraction, which is not easy in noisy images, and the conics representation, which has a singularity when the 3D lines are in the same plane as the optical axis.

In this paper, we present two contributions. We, first, propose a new spherical formulation of the pose estimation by VVS, using an object model composed of lines, withdrawing singularities. The second contribution is an extension of this scheme to an omnidirectional stereovision sensor, which is made with four mirrors and a single camera.

The paper is organized as follows. First, the sensors used in experiments are described and their projection models recalled in section 2. Then the new formulation of this feature and its consequences are presented in section 3. Both formulations are fairly compared on synthetical images (section 3.7). The extension to stereoscopic systems is finally presented and results on real images show the achievement of the proposed approach in section 4. Several

problematic cases are studied and they highlight the superiority of the new formulation for omnidirectional vision and stereovision, particularly in pose estimation experiment of a mobile robot. A discussion in section 5 analyzes, sums up the contributions of the paper and concludes on perspectives of the work.

2. Sensors and models

In this work, an orthographic camera combined with a paraboloidal mirror, in the monocular case, and with several paraboloidal mirrors, in the stereo case, are considered. This configuration has single viewpoint property and is also called a central camera [30].

2.1. Unified central projection model

We propose to use the unified projection model for central cameras [31]. According to this model, a single viewpoint projection can be modeled by a stereographic projection involving a unitary sphere. This is equivalent to ad hoc models of central cameras but has the advantage to be valid for a set of different cameras [32]. Hence, a 3D point $\mathbf{X} = (X, Y, Z)^\mathsf{T}$ is first projected on this unitary sphere as $\mathbf{X}_\mathcal{S}$, thanks to the spherical projection function $pr_\mathcal{S}(.)$:

$$\mathbf{X}_\mathcal{S} = \begin{pmatrix} X_\mathcal{S} \\ Y_\mathcal{S} \\ Z_\mathcal{S} \end{pmatrix} = pr_\mathcal{S}(\mathbf{X}) \quad \text{with} \quad \begin{cases} X_\mathcal{S} = \frac{X}{\rho} \\ Y_\mathcal{S} = \frac{Y}{\rho} \\ Z_\mathcal{S} = \frac{Z}{\rho} \end{cases} \tag{1}$$

where $\rho = \sqrt{X^2 + Y^2 + Z^2}$. Then, $\mathbf{X}_\mathcal{S}$ is projected in the normalized image plane as $\mathbf{x}$ by a perspective projection, *i.e.* $\mathbf{x} = pr(\mathbf{X}_\mathcal{S})$, thanks to a second projection center for which the sphere has coordinates $(0, 0, \xi)^\mathsf{T}$. The direct relationship between $\mathbf{X}$ and $\mathbf{x}$ defines the projection function pr_ξ:

$$\mathbf{x} = pr_\xi(\mathbf{X}) \quad \text{with} \quad \begin{cases} x = \frac{X}{Z+\xi\rho} \\ y = \frac{Y}{Z+\xi\rho} \end{cases} \tag{2}$$

$\mathbf{x}$ is a point of the normalized image plane and is transformed in pixelic coordinates $\mathbf{u}$ thanks to the intrinsic matrix $\mathbf{K}$: $\mathbf{u} = \mathbf{K}pr_\xi(\mathbf{X}) = pr_\gamma(\mathbf{X})$, where $\gamma = \{\alpha_u, \alpha_v, u_0, v_0, \xi\}$ is the set of intrinsic parameters.

The projection from the sphere to the image plane is invertible, allowing to retrieve a spherical point from an image point:

$$\mathbf{X}_{\mathcal{S}} = pr_{\xi}^{-1}(\mathbf{x}) = \begin{pmatrix} \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}x \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}y \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} - \xi \end{pmatrix} \tag{3}$$

2.2. The stereo sensor : Four On One

Our stereo sensor is composed by a unique camera equipped with a telecentric lens and four parabolic mirrors placed in a square to have their axes parallel to the camera main optical axis. They are at the same distance from the camera (Fig. 1). It is nicknamed the FOO (*Four On One* sensor) [5].

A virtual omnidirectional camera, modeled by the stereographic model, can be associated to each mirror, rigidly combined. Each "camera" handles a set of intrinsic parameters γ_j. Four pairs of α_u and α_v are thus used. This is not the minimal parameters set for the FOO since only one real camera is used and, at least, the ratio of the width of a pixel over its height should be unique for the four virtual cameras. However, this modeling is considered to propose a method which is generic to other kinds of omnidirectional stereo rigs.

To model the FOO stereo rig, the first camera/mirror is fixed as the stereo frame origin. The three other cameras are placed relatively to the first one



Figure 1: The omnidirectional stereo sensor we used: orthographic camera, parabolic mirrors, 90 cm height

and their relative poses to the first camera are noted as homogeneous matrices ${}^{c_2}\mathbf{M}_{c_1}$, ${}^{c_3}\mathbf{M}_{c_1}$ and ${}^{c_4}\mathbf{M}_{c_1}$. These poses and the four intrinsic parameters sets model the full stereo system. One can note this model is expendable to more cameras/mirrors, knowing each γ_j and ${}^{c_j}\mathbf{M}_{c_1}$.

3. Model based tracking on the equivalent sphere

The pose estimation method optimizes the pose of the camera relatively to an object, minimizing the error between the forward projection of the object 3D model and image measurements (Fig. 2).

In this work, we consider that indoor and urban outdoor environments are made of 3D segments. Figure 3 details the model of a 3D line, its projection on the equivalence sphere as a portion of great circle (intersection between $\mathcal{P}_1$ and $\mathcal{S}$) and then its projection on image place, as a conic. In our previous work [34], we minimized the point-to-conic distance in the omnidirectional image plane. The error is computed between the latter conic and sample points like in [27]. The main drawback of this approach is the degeneracy of the conic when its 3D line forms a plane with the optical axis of the camera [31]. These lines project as radial straight lines in omnidirectional images. A solution could be to combine two representations, the conic and the line in the image, but the problem is still present when a 3D straight line is near the singular case. Indeed, in that case, the conic is very bad conditioned, leading to perturbations in the computation of the distance (algebraic measure), between a point and the conic.

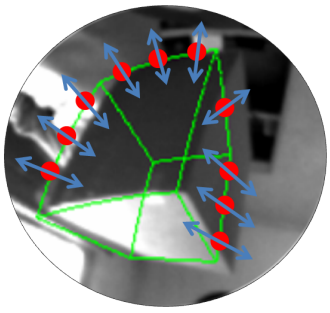


Figure 2: Looking for the corresponding edge along normal vectors (blue) to the conic (green) in each sample (red) of the projected 3D model using the moving edge method [33].

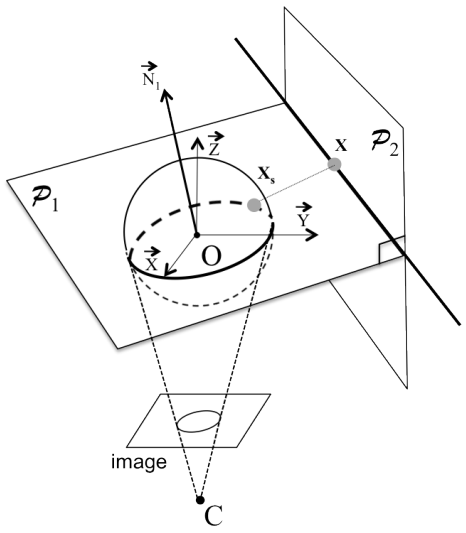


Figure 3: Projection of a 3D straight line, defined by the intersection of planes $\mathcal{P}_1$ and $\mathcal{P}_2$, as a great circle on the sphere $\mathcal{S}$.

Finally, computing a geometric distance between a point and the projection of a 3D straight line, rather than an algebraic one, is more efficient when the conic is badly conditioned and is more discriminant since it is a true distance [35].

To solve these issues, we propose to reformulate the point-to-line feature on the sphere of the unified projection model. This solves the first singularity problem of the projection of a 3D line as a conic. Indeed, even if a 3D straight line forms a plane with the camera axis, its projection on the sphere is not singular. Furthermore, minimizing the point-to-line distance on the equivalent sphere, which is now a *point-to-great circle* distance, rather than in the image plane allows to easily use a geometric distance as feature, contrary to the conic case.

3.1. Algorithm overview

The goal of the proposed algorithm is to track a known moving object in a sequence of images. As in almost all tracking algorithm, we assume that the motion of the object between two images is small. Image measurements are first of all taken and are the input of the following tracking and pose estimation procedure:

1. Consider an initial pose ${}^c\mathbf{M}_o$ of the object, usually the optimal pose of

the previous image (manual selection for the first image of the sequence, in experiments of this paper).

2. Project the 3D model on the sphere and sample it. A set of points is thus generated for each segment of the 3D model.
3. Project these points in the current image for pose ${}^{c}\mathbf{M}_{o}$. They may not fit to the actual object (Fig. 2) since it moved between the two images.
4. Apply the moving edge algorithm [33] to each conic of the 3D model in the image to find the set of sample points on the real corresponding contour.
5. Back-project the obtained points on the sphere.
6. Compute the point-to-great circle distances and interaction matrices.
7. Apply the VVS control law to compute the update vector of the pose.

Both latter items are repeated until convergence. Then, the next image of the sequence is waited and the process restart at the first item of the list.

Following sections are organized as follows. First, the spherical virtual visual servoing is presented (section 3.2) as it introduces what need to be computed to optimize the pose. Then the new point-to-great circle feature is defined in section 3.3 as well as a way to process the image to extract measurements (section 3.4), its Jacobian (sections 3.5 and 3.6), needed for the VVS. Figure 4 graphically sums up the algorithm before tackling its validation on synthetic data in section 3.7.

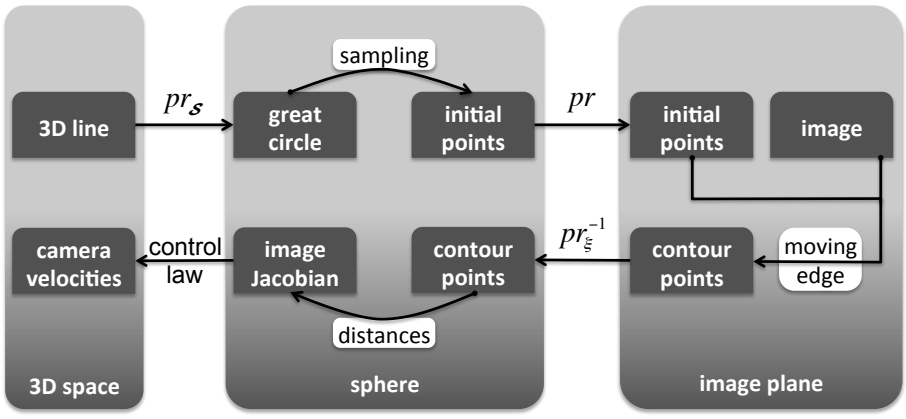


Figure 4: Synopsis of the algorithm.

3.2. Virtual visual servoing

3.2.1. General formulation

The virtual spherical camera is defined by its projection function $pr_{\mathcal{S}}(.)$ and the 3D object by several features ${}^{o}\mathbf{S}$ defined in its own frame. The VVS method estimates the real pose ${}^{c}\mathbf{M}_{o}$ of the object in the camera frame by minimizing the error Δ (eq. (4)). Δ is computed between the set of detected spherical features $\mathbf{s}^{*}$ and the position $\mathbf{s}(\mathbf{r})$ of the same features computed by forward projection. $\mathbf{r} = [t_X, t_Y, t_Z, \theta u_X, \theta u_Y, \theta u_Z]$ is a vector representation of the pose ${}^{c}\mathbf{M}_{o}$ (3 translation and 3 rotation parameters). Considering k features, Δ is defined as:

$$\Delta = \sum_{i=1}^{k} (\mathbf{s}_i(\mathbf{r}) - \mathbf{s}_i^{*})^2, \text{ with } \mathbf{s}(\mathbf{r}) = pr_{\mathcal{S}}({}^{c}\mathbf{M}_{o}, {}^{o}\mathbf{S}_i). \tag{4}$$

i indexes the i-th feature of the set.

With this formulation, a virtual spherical camera, with initial pose ${}^{c}\mathbf{M}_{o}^{0}$, is moved using a visual servoing control law to minimize the error Δ. At convergence, the virtual sphere reaches the pose ${}^{c}\mathbf{M}_{o}^{*}$ which minimizes Δ. Assuming this non-linear estimation process converges, this pose is the real pose. To sum up, this positioning task is expressed as the regulation of the error

$$\mathbf{e} = \mathbf{s}(\mathbf{r}) - \mathbf{s}^{*}. \tag{5}$$

Imposing an exponential decoupled decrease of the error, $\dot{\mathbf{e}} = -\lambda \mathbf{e}$ leads to the optimization of the pose so that the error evolution curve has an exponential decreasing profile. $\dot{\mathbf{e}}$ depends on $\dot{\mathbf{s}}$. The image features motion is linked to the virtual camera velocity $\mathbf{v}$ [36]:

$$\dot{\mathbf{s}} = \mathbf{L}_{\mathbf{s}} \mathbf{v}. \tag{6}$$

$\mathbf{v} = (\upsilon, \omega)^{\mathsf{T}}$ is the velocity vector of the pose with $\boldsymbol{\upsilon} = (\upsilon_X, \upsilon_Y, \upsilon_Z)$, the translation velocity and $\boldsymbol{\omega} = (\omega_X, \omega_Y, \omega_Z)$, the rotation velocity. $\mathbf{L}_{\mathbf{s}}$ is the interaction matrix (or image Jacobian) linking the features motion to the camera velocity.

3.2.2. Robust formulation

To make the pose estimation more robust, it has to deal with outliers and the use of M-estimator [28] is an efficient solution in this way. Some functions exist in the literature and apply well to our problem [37]. They allow

uncertain measures to be less likely considered and in some cases completely rejected. To add robust estimation to our objective function Δ (eq. (4)), it is modified as:

$$\Delta_{\mathcal{R}} = \sum_{i=1}^{k} \mathsf{P}(\mathbf{s}_i(\mathbf{r}) - \mathbf{s}_i^*), \tag{7}$$

where $\mathsf{P}(a)$ is a robust function that grows subquadratically and is monotonically non-decreasing with increasing $|a|$. For P, we considered the Tukey's function [28] because it completely rejects outliers and give them a zero weight. This is of interest in virtual visual servoing so that detected outliers have no effect on the virtual camera motion [38].

Iterative Re-Weighted Least Squares is a common method of applying the M-estimator. Thus, the error to be regulated to 0 is redefined, taking the robust function into account:

$$\mathbf{e} = \mathbf{D}(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*), \tag{8}$$

where $\mathbf{D}$ is a diagonal weighting matrix given by $\mathbf{D} = \text{diag}(w_1, ..., w_k)$. Each w_i is a weight given to specify the confidence in each feature location and is computed using statistics on the error vector [24].

A simple control law that allows to move the virtual camera can be designed to try to ensure an exponential decoupled decrease of $\mathbf{e}$:

$$\mathbf{v} = -\lambda \left(\mathbf{D}\mathbf{L}_\mathbf{s}\right)^+ \mathbf{D} \left(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\right). \tag{9}$$

where λ is a gain that tunes the convergence rate and the matricial operator $(\,)^+$ denotes the left pseudo inverse of the matrix inside the parentheses.

The pose ${}^c\mathbf{M}_o$ is then updated using the exponential map $e^{[.]}$ of $SE(3)$ [39]:

$${}^c\mathbf{M}_o^{t+1} = {}^c\mathbf{M}_o^t e^{[\mathbf{v}]}, \tag{10}$$

The feature type choice and hence the interaction matrix expression are a key point of this algorithm. As previously mentioned, we chose the point-to-great circle feature and next sections describe the spherical virtual visual servoing for it.

3.3. Definition of the point-to-great circle feature

As we can see in figure (3), the 3D line can be represented by the intersection of two planes. These two planes $\mathcal{P}_1$ and $\mathcal{P}_2$ are defined, in the sphere frame, by:

$$\begin{aligned} \mathcal{P}_1 &: A_1X + B_1Y + C_1Z \quad\;\; = 0 \\ \mathcal{P}_2 &: A_2X + B_2Y + C_2Z + D_2 = 0 \end{aligned} \tag{11}$$

with the following constraints on the 3D parameters:

$$\begin{cases} A_1^2 + B_1^2 + C_1^2 = 1 \\ A_2^2 + B_2^2 + C_2^2 = 1 \\ A_1A_2 + B_1B_2 + C_1C_2 = 0 \end{cases} \tag{12}$$

so that the two planes are orthogonal with unit normals $\mathbf{N}_1 = (A_1, B_1, C_1)^\mathsf{T}$ and $\mathbf{N}_2 = (A_2, B_2, C_2)^\mathsf{T}$. D_2 is the orthogonal distance of the 3D line to the unified model sphere origin.

The spherical projection of a 3D line is then defined as the intersection of the unitary sphere $\mathcal{S}$ and plane $\mathcal{P}_1$:

$$\begin{cases} X^2 + Y^2 + Z^2 = 1 \\ A_1X + B_1Y + C_1Z = 0 \end{cases} . \tag{13}$$

Since features are lying on the sphere surface, the normal vector $\mathbf{N}_1$ is sufficient to parameterize the great circle.

Considering a point $\mathbf{X}_\mathcal{S}$ on the sphere, *i.e.* an image contour point projected on the sphere, its signed distance to the great circle is expressed by the dot product $d = \mathbf{N}_1.\mathbf{X}_\mathcal{S}$:

$$d = A_1X_\mathcal{S} + B_1Y_\mathcal{S} + C_1Z_\mathcal{S} \tag{14}$$

This signed distance is interesting since $d = 0$ when the point is on the great circle and tends to ± 1 when going away the great circle.

The interaction matrix linked to this distance is expressed considering the time variation of d, with $\mathbf{X}_\mathcal{S}$ constant during the pose optimization of one image :

$$\dot{d} = \dot{A}_1X_\mathcal{S} + \dot{B}_1Y_\mathcal{S} + \dot{C}_1Z_\mathcal{S}. \tag{15}$$

This clearly depends on the interaction matrix of $\mathbf{N}_1$, *i.e.* its variation with respect to the pose. It is described in the following section.

3.4. Measuring edge image points

We mention that edge image points are measured since they are obtained thanks to initial points computed from the 3D model projection at an initial pose. This is the low level tracking method already used in [33]. Let us note that it is different to a point detection method.

Points used to compute the *point-to-great circle* distance are obtained thanks to an adaptation of the moving edge method to omnidirectional images [34]. In the latter work, a 3D segment of the 3D model is projected as a conic in the omnidirectional image. The method needs a regular sampling of the conic in order to use the obtained sample points as initial positions for the edge searching, along normal directions to the conic. However, a conic to circle transformation and its inverse are necessary for the regular sampling. Indeed, the direct regular sampling of the conic is not trivial, but for a circle, it is.

In the current work, as introduced in the algorithm overview (section 3.1), the sampling of the 3D line projection is done on its corresponding portion of great circle, *i.e.* on the sphere. It allows a direct regular sampling, contrary to the conic case. Samples are then projected in the image to find the corresponding contour thanks to the moving edge method.

3.5. Interaction matrix for the great circle

The interaction between the time variation of the normal vector to the great circle and the camera motion is expressed from $\dot{\mathbf{N}}_1 = (\dot{A}_1, \dot{B}_1, \dot{C}_1)^\mathsf{T}$ [40]:

$$\dot{\mathbf{N}}_1 = -\frac{1}{D_2}\mathbf{N}_1\mathbf{N}_2^\mathsf{T}\upsilon - \mathbf{N}_1 \times \omega, \tag{16}$$

with υ and ω, the translation and rotation velocities of the spherical camera (section 3.2.1). The time variation of $\mathbf{N}_1$ leads to the interaction matrix linked to the projection of a 3D line on the sphere:

$$\mathbf{L}_{\mathbf{N}_1} = \begin{bmatrix} \mathbf{L}_{A_1} \\ \mathbf{L}_{B_1} \\ \mathbf{L}_{C_1} \end{bmatrix} = \left[-\frac{1}{D_2}\mathbf{N}_2\mathbf{N}_1^\mathsf{T} \;\middle|\; [\mathbf{N}_1]_\times \right] \tag{17}$$

3.6. Interaction matrix for the point-to-great circle distance feature

From equation (15), and knowing the elements of $\mathbf{L}_{\mathbf{N}_1}$ (eq. (17)), the interaction matrix $\mathbf{L}_d$ related to the point-to-great circle distance is expressed as:

$$\mathbf{L}_d = \begin{pmatrix} X_\mathcal{S} \\ Y_\mathcal{S} \\ Z_\mathcal{S} \end{pmatrix}^\mathsf{T} \begin{pmatrix} \mathbf{L}_{A_1} \\ \mathbf{L}_{B_1} \\ \mathbf{L}_{C_1} \end{pmatrix}. \tag{18}$$

The feature of the VVS is the signed distance d (eq. (14)). So, considering k features, as in equation (7), the stacking of the k point-to-line distances

for current pose gives $\mathbf{s}(\mathbf{r})$. The goal of the optimization is to reach null distances between points to great circle and hence, $\mathbf{s}^* = 0$ (a k-vector with all elements equal to zero). Finally, the stacking of the k interaction matrices $\mathbf{L}_d$ leads to $\mathbf{L}_\mathbf{s}$ and equation (8) allows to compute the update vector of the pose.

3.7. Validation on synthetic images

This section presents results obtained on images synthesized using the Pov-Ray software. The goal of these experiments is to study the behavior and accuracy of our proposed approach (S) comparing to the image plane based (IP) [27], and their impact on the estimated pose. The implementation has been done in C++ using the ViSP library [41].

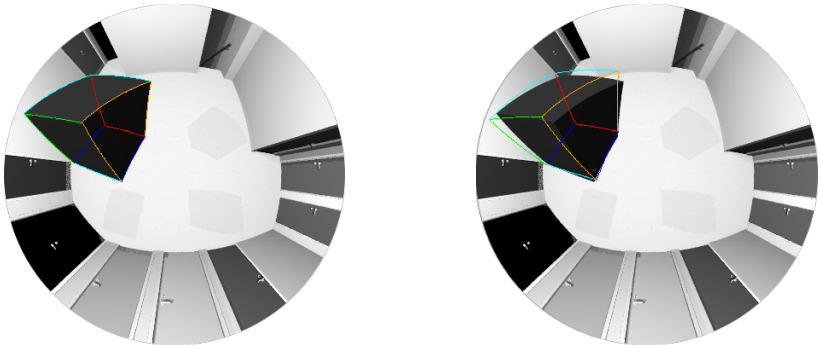
To do this evaluation, two kinds of omnidirectional images are processed. One kind presents a box where lines are not radial in the image (Fig. 5) and the other presents a majority of radials (Fig. 6). The latter case is a near-singular situation for the image plane (IP) formulation in which it should be less efficient than in the former case, and also less efficient than the spherical formulation (S).

The experiments done with synthetical images allow to evaluate the convergence rate of the algorithms and their robustness to noise in the previously mentioned situations. The pose estimation error at convergence and the conditioning of interaction matrices are good tools to better understand the behavior of algorithms.

Three sets of experiments are led to evaluate the:

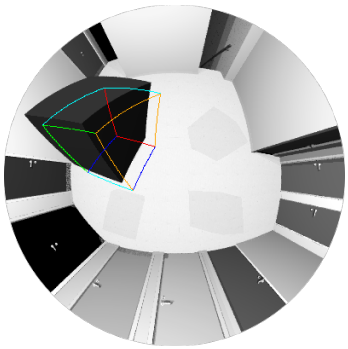
1. robustness to initialization: several different initial poses with a unique desired pose (robust estimation or not) (128 different initial poses). Random variations of the optimal pose are computed in order to be not too far in images for the low level moving edge to succeed.
2. robustness to low quality camera parameters: different level of noise (1 %, 2.5 %, 5 %) applied to the camera intrinsic parameters in terms of percentage of parameter values (1000 random choices each)
3. robustness to low quality 3D model of the object (*i.e.* the model is not exactly the same as the real object, due to fabrication or measurement errors of the object): different level of noise (1 %, 2.5 %, 5 %) applied to the vertices of the object 3D model (1000 random choices each)

The two latter sets of experiments are led considering the robust estimation. For experiments with noisy camera parameters or 3D model vertices, the

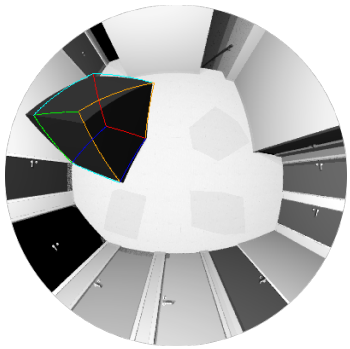


(a) optimal

(b) one of the initial poses



(c) camera noised case

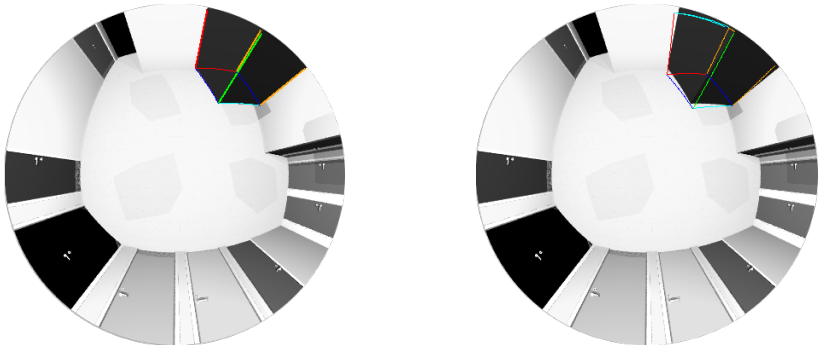


(d) model noised case

Figure 5: Projection of the 3D model of the box in images, in the perfect case (a), for another pose (b), for the "optimal pose" but with noisy intrinsic parameters (c) and finally, for the optimal pose, with a noise applied on the summits of the 3D model (d).

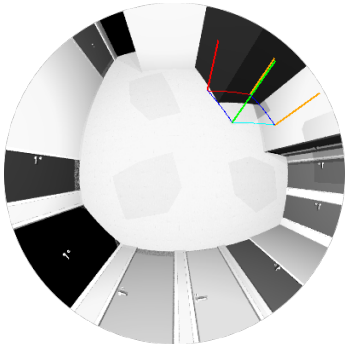
initial pose of image plane and spherical VVS is the optimal one, for the perfect camera parameters and perfect model. The robustness is evaluated with respect to (i) the convergence rate, (ii) the orientation and position precision of the optimal pose and (iii) the interaction matrix conditioning.

Figure 7 shows convergence rates. We first observe that, when no 3D line segments of the model projects as a radial, image plane and spherical formulations lead to quasi identical convergence rates. A thin advantage for the spherical formulation is however noted when the object 3D model is importantly noised (Fig. 7(a), on the right side). Secondly, when the model is mainly projected as radials, the spherical formulation clearly outperforms the image plane one. This was expected from the theoretical developments and is confirmed by experiments as we can see in the figure 7(b), for various

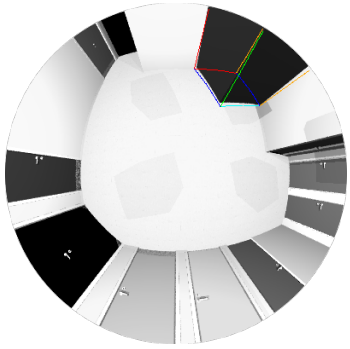


(a) optimal

(b) one of the initial poses



(c) camera noised case

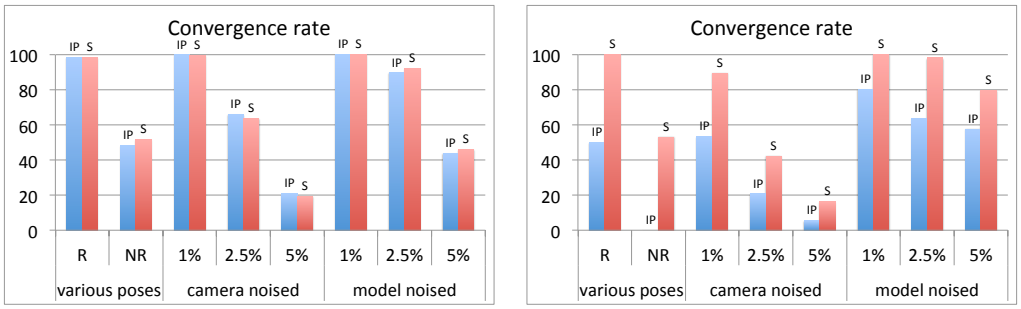


(d) model noised case

Figure 6: Same different cases than in figure 5 but the pose is such that the box edges are mainly projected as radials in the image.

initial poses with or without robust estimation. In that case, the IP formulation converges 50% of the time when the M-Estimator is used whereas it never converges when it is not used. The former rate is due to the fact that since a conic close to be a radial is badly conditioned, important algebraic errors are encountered, even for a small geometric one, and the M-Estimator rejects these measures. Hence, no measures are kept on radials and only two segments of the model can be used to estimate the pose, which is subject to ambiguities. When the M-Estimator is not used, sample points on radials are kept in the optimization process and make the optimization unstable that finishes to diverge or to converge in a local minimum.

Still about the various initial poses experiment in the presence of radials, the spherical formulation allows to converge in all the tested cases when the



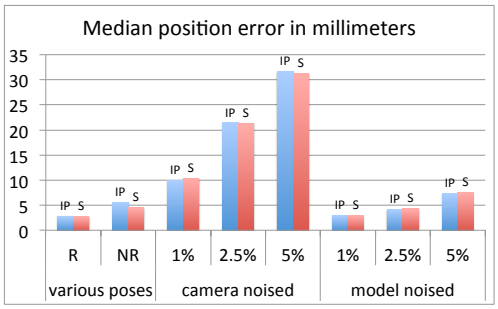
(a) no radials case (b) radials case

Figure 7: Comparison of convergence rates, expressed as percentages of converged optimization over number of initializations, for the image plane (IP) and spherical (S) formulations with radials or not. "R" stands for "Robust" and "NR" for "No Robust". For the camera noised and model noised experiments, the robust estimator is always used.

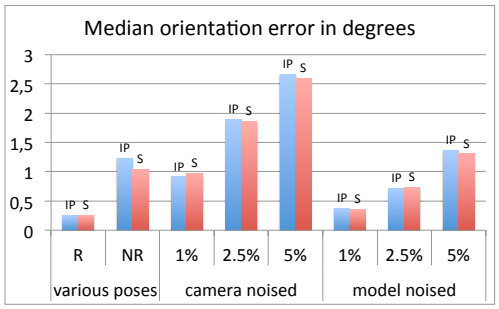
robust estimation is used. However, it only converges at a rate of 50% to the actual pose when the M-Estimator is not used, because some sample points are attracted by shadows of the box in the image or samples points of a line are attracted by the contour of other object edges.

When applying a Gaussian noise on camera parameters or vertices of the 3D model, at 1%, 2.5% or 5% of their value, the convergence rate decreases when the level of noise is increasing, which is common. Of course, the noise on camera parameters has a major impact on the convergence rate. Indeed, the generalized focal length is noised up to ± 10, for an original value of 200, the principal point too and ξ is noised up to ± 0.05 for an original value of 1. For instance, figures 5(c) and 6(c) show the projection of the 3D model when camera parameters are noised, for the optimal pose (corresponding to noiseless parameters). The spherical formulation outperforms the image plane one, nearly doubling the convergence rate. The next diagrams have to be cross-interpreted with the latter convergence survey.

Working with synthetic images allows to evaluate the pose estimation precision itself without other unknown perturbations. We first study the position and orientation estimation errors in the non radial case (Fig. 8). In this case, as in the convergence survey, both formulations (IP and S) lead to similar results with a slightly better precision in position, and more particularly in orientation, of the spherical formulation. The "various initial poses" experiments of figure 8 show the estimation precision at convergence is, at least, doubled when using the robust estimation: a position error of



(a) position errors (no radials case)



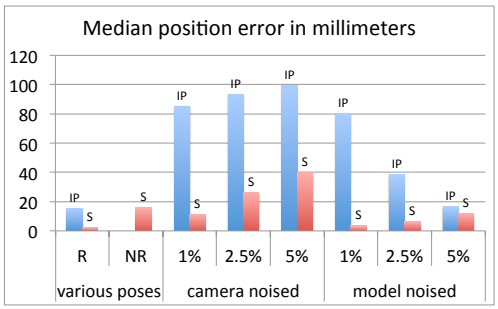
(b) orientation errors (no radials case)

Figure 8: Position and orientation errors at convergence for the image plane (IP) and spherical (S) formulations without radials. "R" stands for "Robust" and "NR" for "No Robust". For the noised experiments, the robust estimator is always used.

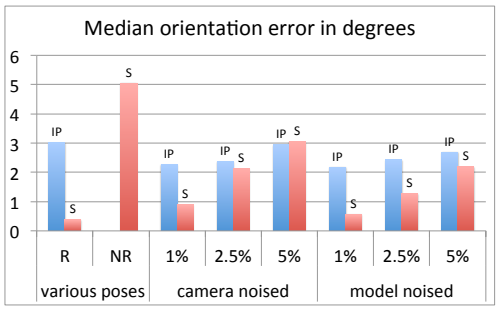
around 2.5 cm with the robust estimation against 5 cm without.

In the presence of radial lines, the pose estimation precision is clearly better, as compared with the known groundtruth, for the spherical formulation than for the image plane one, even only considering the cases of convergence. Figure 9 shows that the spherical formulation allows not only to converge more frequently but the pose estimation is also more precise than with the image plane formulation, especially when the model is projected as radials in the image.

Another tool to study the behavior of non linear optimization algorithms



(a) position errors (radials case)



(b) orientation errors (radials case)

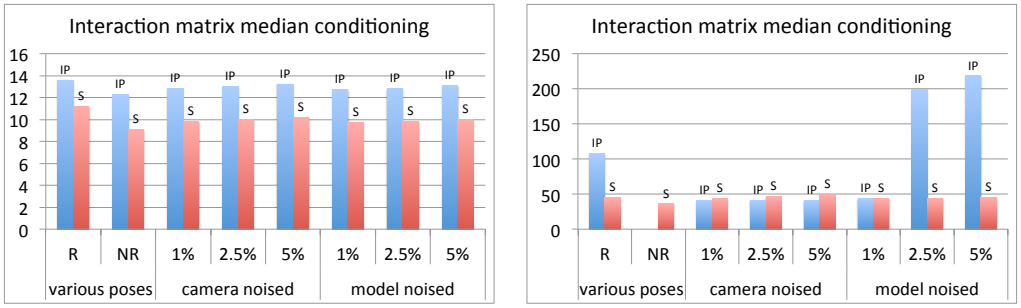
Figure 9: Position and orientation errors at convergence for the image plane (IP) and spherical (S) formulations with radials. "R" stands for "Robust" and "NR" for "No Robust". For the noised experiments, the robust estimator is always used.

is the conditioning of the Jacobian so, in the current work, the conditioning of the interaction matrix. The closer to 1 the condition number, the better conditioned it is. Figure 10(a) shows the median condition number of the interaction matrix in various experiments. The median is chosen and not the mean because, in some cases, the interaction matrix condition number for IP formulation is extremely high (about 10^6). The mean comparison with S formulation would then not be fair. Similar results are obtained by the two studied formulations of the 3D tracking, with a similar median condition number in all experiments where the optimization converged.

On the contrary, when looking at the median conditioning number of the robust interaction matrix for the various initial poses experiment (Fig. 10(b), left part), it helps well to explain the difference of convergence rate between both formulations. Indeed, the interaction matrix of the spherical formulation is twice better conditioned than the image plane formulation.

However, when using a noised set of camera parameters in the presence of radials, the condition number of both methods is similar, and even slightly better for the IP formulation (Fig. 10(b), center part). It helps to understand that, this is precisely the error computation as an algebraic distance in the IP formulation which leads to a low convergence rate, in this case, and a low precision when converging.

Finally, the noised model experiments in the presence of radial lines (Fig. 10(b), right part) clearly show the image plane formulation is more



(a) no radials case (b) radials case

Figure 10: Comparison of the conditioning of interaction matrices of the VVS for the image plane (IP) and spherical (S) formulations with radials in the image plane or not. "R" stands for "Robust" and "NR" for "No Robust". For the camera noised and model noised experiments, the robust estimator is always used.

sensitive to a noised 3D model than the spherical one. And, more globally, about the conditioning of the Jacobian in all experiments, the spherical formulation shows to be very robust to noise, applied on the camera parameters or on the 3D model, with a quasi constant median conditioning, despite noise.

To conclude, this survey experimentally highlights the interest of the spherical modeling of the *point-to-line* distance, with the improvement of convergence rate and pose estimation precision over the image plane formulation.

4. Stereoscopic extension of the model based tracking in omnidirectional vision

4.1. Robust stereoscopic virtual visual servoing

The aim of this section is to present how to adapt the VVS method to an omnidirectional stereovision sensor considered as a rig of four single viewpoint omnidirectional cameras. The criterion Δ (eq. (4)) has to be rewritten to take into account several cameras, assuming known stereo system calibration parameters [42]. So, for N cameras, Δ is extended to:

$$\Delta_{\mathtt{S}} = \sum_{j=1}^{N} \sum_{i=1}^{k_j} (pr_{\mathcal{S}}({}^{c_j}\mathbf{M}_{c_1}\, {}^{c_1}\mathbf{M}_o, {}^{o}\mathbf{P}_i) - {}^{c_j}\mathbf{s}_i^*)^2 \tag{19}$$

with, obviously, ${}^{c_1}\mathbf{M}_{c_1} = \mathbf{I}_{4\times 4}$ and knowing the $N-1$ sets of relative camera poses with respect to the reference one. With this formulation, only six parameters have to be estimated, as for the monocular pose estimation problem. Setting $N = 2$, we retrieve a two cameras case ([25], in perspective vision) and for the FOO, $N = 4$ (Fig. 1).

To express the features motion in images of cameras 2, 3 and 4 with respect to the velocity $\mathbf{v}_1$ of camera 1, *i.e.* the stereo rig velocity, a frame change is applied to velocity vectors $\mathbf{v}_j$ of cameras $j = 2, 3, 4$. Velocity vector frame changes are computed thanks to the twist transformation matrix:

$${}^{c_j}\mathbf{V}_{c_1} = \begin{bmatrix} {}^{c_j}\mathbf{R}_{c_1} & [{}^{c_j}\mathbf{t}_{c_1}]_\times \\ 0 & {}^{c_j}\mathbf{R}_{c_1} \end{bmatrix} \tag{20}$$

where ${}^{c_j}\mathbf{R}_{c_1}$ and ${}^{c_j}\mathbf{t}_{c_1}$ are respectively the rotational bloc and the translation vector extracted from the homogeneous matrix ${}^{c_j}\mathbf{M}_{c_1}$. Then, we can express $\mathbf{v}_j$ w.r.t. $\mathbf{v}_1$:

$$\mathbf{v}_j = {}^{c_j}\mathbf{V}_{c_1}\mathbf{v}_1, j = 2..4. \tag{21}$$

So the feature velocity in the spherical image of the camera j is related to the motion of camera 1:

$$\dot{\mathbf{s}}_j = \mathbf{L}_j \mathbf{v}_j = \mathbf{L}_j {}^{c_j}\mathbf{V}_{c_1} \mathbf{v}_1. \tag{22}$$

For four cameras, considering a M-Estimator is computed on each image, leading to four weighting matrices ($\mathbf{D}_{1..4}$), the pose velocity vector of the rig, expressed in camera 1 frame, is computed as:

$$\mathbf{v}_1 = -\lambda \begin{bmatrix} \mathbf{D}_1\mathbf{L}_1 \\ \mathbf{D}_2\mathbf{L}_2 {}^{c_2}\mathbf{V}_{c_1} \\ \mathbf{D}_3\mathbf{L}_3 {}^{c_3}\mathbf{V}_{c_1} \\ \mathbf{D}_4\mathbf{L}_4 {}^{c_4}\mathbf{V}_{c_1} \end{bmatrix}^{+} \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \\ \mathbf{D}_3 \\ \mathbf{D}_4 \end{bmatrix} \begin{bmatrix} \mathbf{s}_1(\mathbf{r}_1) - \mathbf{s}_1^* \\ \mathbf{s}_2(\mathbf{r}_2) - \mathbf{s}_2^* \\ \mathbf{s}_3(\mathbf{r}_3) - \mathbf{s}_3^* \\ \mathbf{s}_4(\mathbf{r}_4) - \mathbf{s}_4^* \end{bmatrix}. \tag{23}$$

Four M-Estimators are individually computed since the moving edges procedure is done individually in each of the four views of the FOO. Finally, $\mathbf{L}_j$ is substituted by the stacking of each feature interaction matrix $\mathbf{L}_d$, computed for the j-th spherical camera of the rig. The same reasoning is followed to define current and desired features for each camera (see the end of section 3.6).

$\mathbf{v}_1$ is used to update the stereo rig pose, as in the unique camera case, and poses of the three other cameras are then updated using the stereo rig calibration parameters ${}^{c_j}\mathbf{M}_{c_1}$: ${}^{c_j}\mathbf{M}_o = {}^{c_j}\mathbf{M}_{c_1} {}^{c_1}\mathbf{M}_o$. These pose matrices are used in equation (23) to compute $\mathbf{s}_j(\mathbf{r}_j)$ for a new iteration.

4.2. Experimental results

The algorithm has been applied on several image sequences either with a static sensor and mobile objects in the scene and with mobile ones, embedded on a mobile robot.

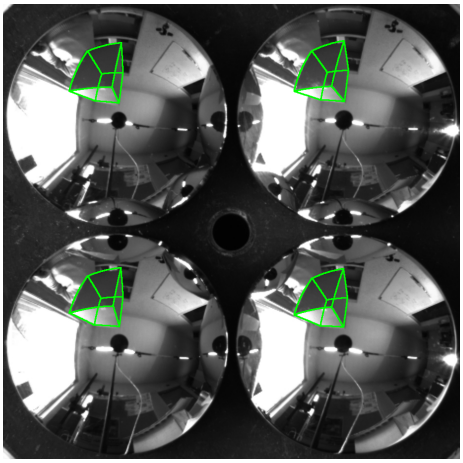
The first handheld box tracking experiment is led to compare image plane based and spherical based VVS methods using the FOO sensor. Pose estimation stability of image plane and spherical formulations in the stereo case are then compared. Then, a localization experiment tracking doors of a scene shows the precision of estimation when using the spherical formulation. Finally, an experiment is led outdoor to track buildings in a more challenging situation than indoor.

4.2.1. Box tracking: sphere versus image plane

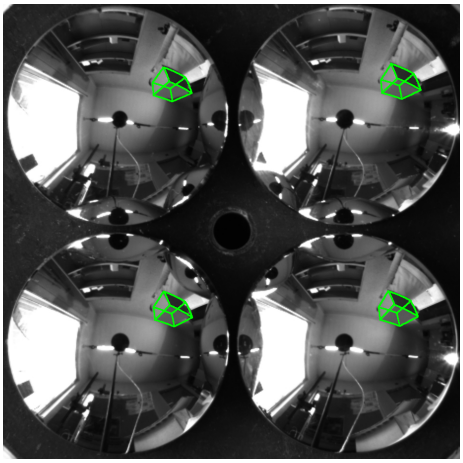
A box (30 cm × 25 cm × 20 cm) is manually moved in a major part of the static sensor field of view (Fig. 11), at a distance range from 35 cm to



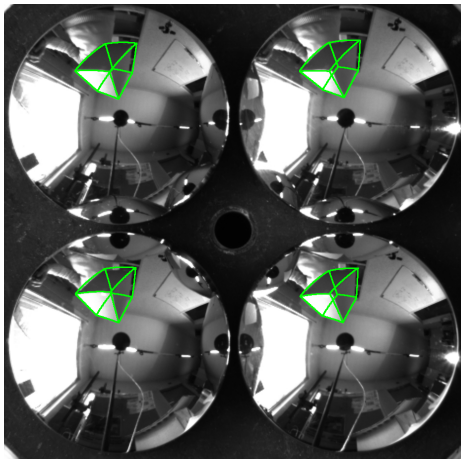
(a) Experimental setup



(b) Image 1



(c) Image 149

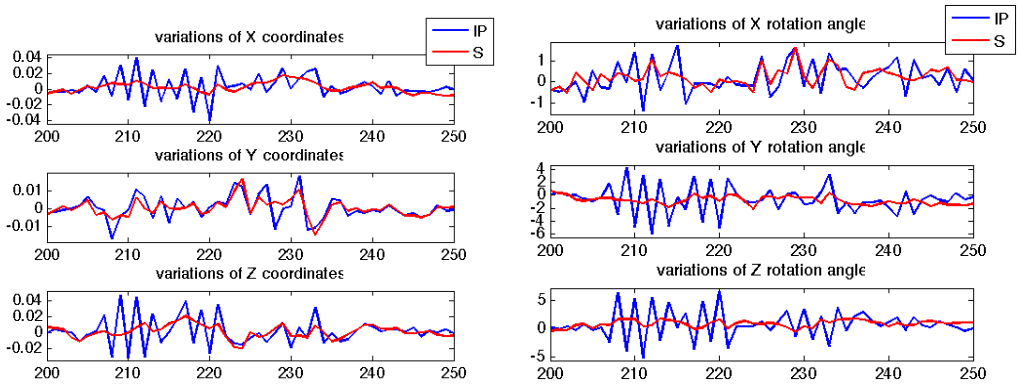


(d) Image 312

Figure 11: Pose estimation for the handheld box. (a) an external view of the setup. (b-d) FOO images with the projected model (green), showing its pose is well computed all along the sequence.

75 cm, in this experimentation. The small size of the box, and hence its small size in images, even if it is not so fare, leads to challenging conditions. The tracking succeeds even in the image plane formulation thanks to the redundancy compensating the occurrence of singularities (Fig. 11). The spherical stereo tracking succeeds too and the tracking results are similar to the image plane formulation. It is however noticeable that the spheri-

cal formulation is more stable than the image plane one, mainly due to the fact that it uses more image measurements, without discarding them due to a near-singularity. Figure 12 presents temporal variations for each position and orientation coordinates estimated from the box stereo sequence, and particularly on a subset of this sequence where the differences between IP and S are the most visible. The instability of the image plane formulation w.r.t. the spherical one is particularly clear, in this experiment, for the X and Z coordinates and Y and Z rotation angles.



(a) position variations in meters (b) orientation variations in degrees

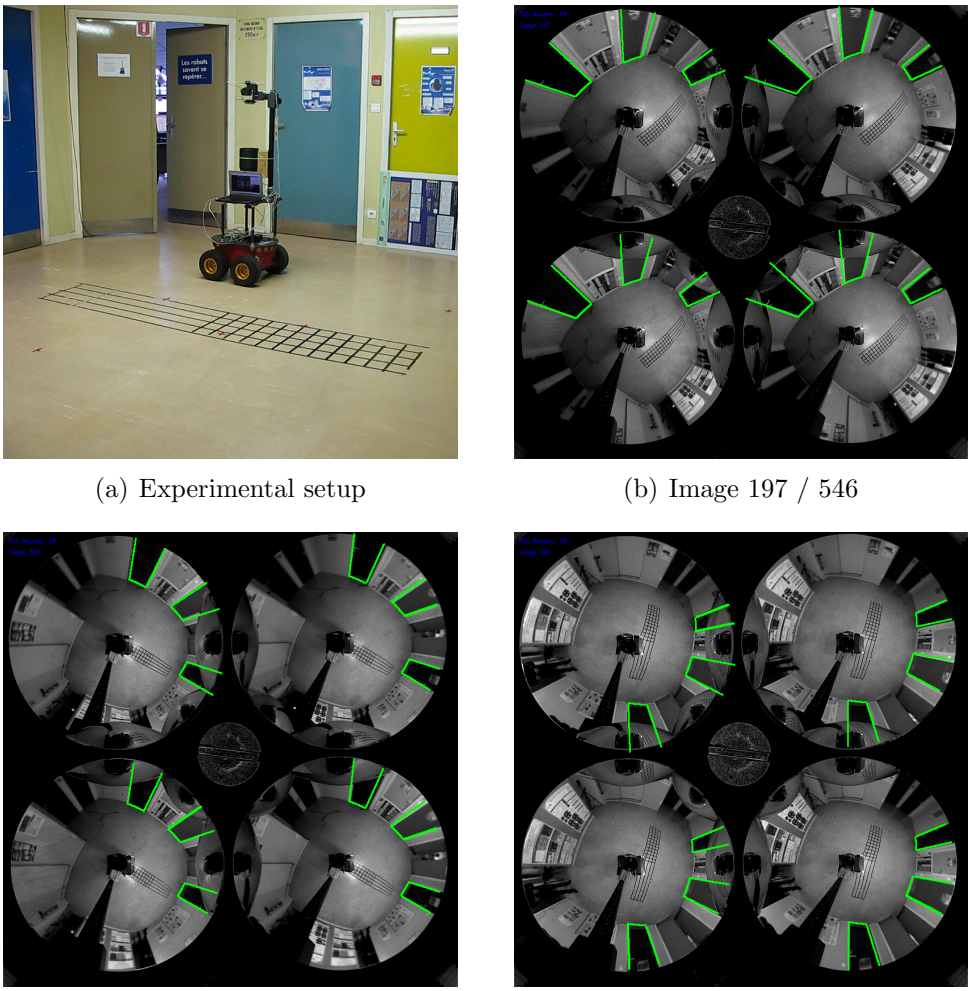
Figure 12: Pose variations between frames 200 and 250 of the sequence presented in figure 11, using the four mirrors. "IP" stands for "Image Plane" formulation and "S" for "Spherical" formulation.

4.2.2. Application to mobile robot localization

Indoor experimentation. In this experiment, we consider a massively radial model projection and mutual occlusion case. The FOO sensor is vertically mounted on a mobile robot with mirrors reflecting the ground and walls. This usual placement of the omnidirectional sensor allows to actually sense information 360° all around the robot. Hence, it leads to the projection of vertical lines as radial lines in the image plane (Fig. 13(b)), the degenerate and even singular case of the normalized conic representation. Indeed, when using doors as the 3D model for VVS in the image plane, the tracking is lost from the first images. However, the spherical formulation of the model based VVS allows to naturally deal with the projection of vertical lines when the sensor axis is also vertical. Figure 13 present tracking results over several

images of a sequence where the robot is moved along a loop trajectory. Despite important rotations of the robot and important size variations of doors in images, the tracking succeeds all along this sequence.

Estimated poses are used to plot a raw estimation of the camera/robot trajectory in figure 14. Comparison is made between the onboard estimation



(a) Experimental setup

(b) Image 197 / 546

(c) Image 255 / 546

(d) Image 343 / 546

Figure 13: Stereo spherical tracking and pose estimation of an environment mainly composed of radial lines in the image. The FOO camera is mounted on a mobile robot with mirrors reflecting the ground and walls. The spherical tracking of scene doors succeeds all along the sequence of 546 images where the robot makes a loop.

using the spherical VVS and the trajectory estimated with an external tool, which is a camera placed at the ceiling, tracking and estimating the pose of an IR target placed on the robot. The overlapping of the two trajectories shows qualitatively the estimation precision, which is important, with stable pose estimations over time, without any filtering. Even if the trajectory obtained with an external vision tool cannot be actually considered as a "ground truth", it gives an idea of it. We then computed the error between these two trajectories and obtained 6.72 cm as mean position estimation error. This mean error leads to the mean error ratio over the trajectory length (around 11 m) of 0.61 %.

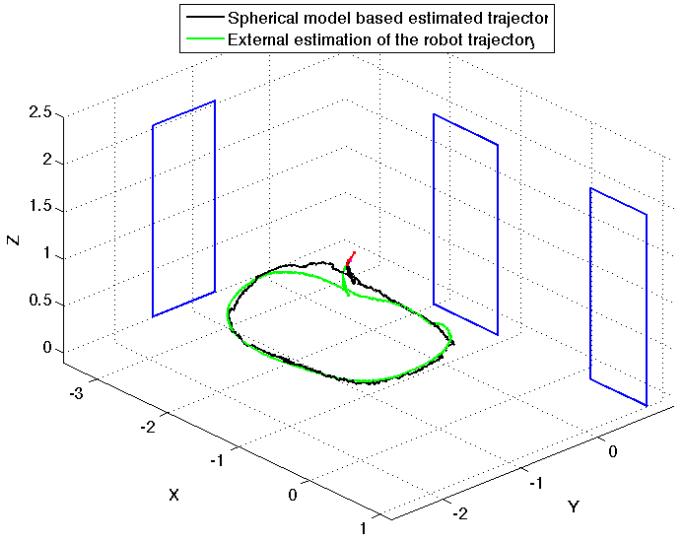


Figure 14: Estimated trajectory using poses obtained from the stereo spherical VVS. The "ground truth" is obtained using an external localization tool. The important overlapping of these trajectories shows the precision of the estimation. The unit is the meter.

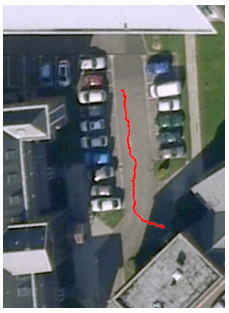
Outdoor experimentation. This experiment shows the behavior of our method in more challenging conditions, traveling the robot outdoor. The robot embeds a small configuration of the FOO sensor, still with the four mirrors but with a smaller lens in order to have a sensor of the same size as a monocular one : 30 cm (fig. 15(b)). A sequence of 1500 images for a 21 m path of the robot is acquired. A building is tracked with a partial and imprecise model of its edges. In the image sequence, the building moves in a large part of images leading to self-occlusion of the building (fig. 15). Despite these diffi-



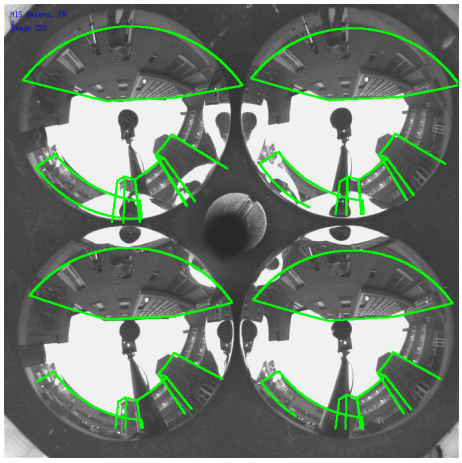
(a) environment; red: an idea of the robot path; green: tracked edges in (d-f)



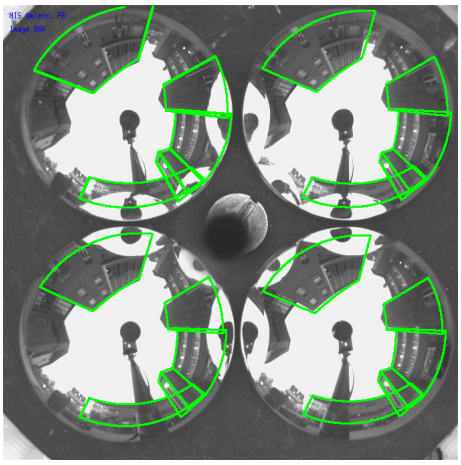
(b) small FOO



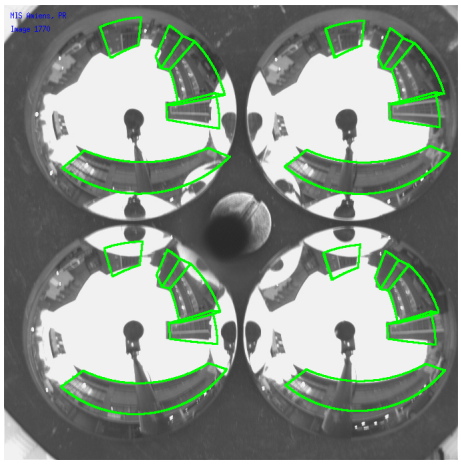
(c) Estimated poses



(d) Image 0 / 1500



(e) Image 540 / 1500



(f) Image 1450 / 1500

Figure 15: Tracking buildings. (a) The outdoor environment. (b) A compact configuration of the FOO is mounted on the robot. (c) Estimated poses (red) using the stereo spherical VVS. The robot is driven along a trajectory of 21 m with 1500 acquired images. Tracking results are shown at (d) the beginning, (e) the middle and (f) the end of the robot path.

culties, the tracking succeeds all along the sequence showing the robustness of the method in real conditions with partial occlusion of the building bottom by some cars, self occlusion of the building, which also present other strong gradients on its surface. One can finally note that, in a mean, the distance between the robot and the building is around 12 m.

A video presenting the two latter experiments is available at the following URL: http://home.mis.u-picardie.fr/∼g-caron/videos/MBTMobileFOO.mp4.

5. Discussion and conclusion

Experiments aimed to evaluate the new spherical formulation of the 3D model based tracking in omnidirectional vision and its extension to stereovision. We compared this new formulation with the image plane based one. Dealing with this spherical representation allows (i) to withdraw singular and nearly singular conic representations, (ii) to reduce the number of parameters to represent a line and, finally, (iii) to easily use an actual geometrical distance. All these theoretical interests are confirmed by the results.

Experimental results on synthetic images show exactly the situation in which the image plane formulation is defeated whereas the spherical formulation is always efficient. Based on this experimental demonstration, the results on real images show two things. First, using the image plane formulation, the redundancy brought by the FOO stereovision sensor adds enough robustness to the tracking, even if it is not perfectly stable. Then, experimental results on real images show that the use of the spherical formulation stabilizes estimations in the tracking of a box. This is furthermore the only formulation allowing to succeed to track an environment for the onboard localization of a mobile robot, when the 3D model is mainly projected as radials, which is really often the case indoor as well as outdoor. The redundancy of the four views is not enough constraining for the image plane formulation in the latter case, highlighting again the interest of the spherical formulation.

Another advantage of the technique is the processing time since, even if no particular code optimization has been led, processing is real time w.r.t. the acquisition frame rate. The tracking of a box in the monocular VVS case for images of 640×480 pixels (450×450 useful) takes around 25 ms and the tracking of doors in the last experiment with the FOO takes 45 ms for 800×800 pixels FOO images, with 250 moving edge sites in each view. Furthermore, the latter computation time can even be reduced since processes are parallelizable for the stereo case. Indeed, the moving edge stage is done on each image individually, as the weights computation of the M-Estimator, and these steps could be done in parallel for each view.

Of course, if image acquisition conditions are really hard for a tracking method (extremely small object in images or important displacement of the tracked object/scene between two consecutive frames), the proposed algorithm may diverge. But if the new spherical method diverges, the former image plane one will diverge too, whereas the opposite has never been observed in any of our experiments. A better behavior of the former image plane formulation over the new spherical one is, furthermore, never waited from the theoretical expressions.

To sum up, the new spherical formulation of model based pose estimation and the redundancy brought by the FOO sensor allow to make the tracking robust and stable in static or mobile situations, indoor and outdoor. Experiments show that the use of this sensor is adapted, efficient and robust, for real time onboard mobile robot localization applications.

Future works will be focused on the low level image processing side to fully adapt the moving edge to the spherical geometry and to study the interests brought by this adaptation. Finally, the mix of the proposed contour-based method with a textured-based one [43] is one of the perspectives of this work too, in order to merge their efficiency in textured and non-textured areas.

References

[1] Nayar S. Sphereo: Determining Depth using Two Specular Spheres and a Single Camera. In: SPIE Conf. on Optics, Illumination, and Image Sensing for Machine Vision III. Cambridge, Massachusetts, USA; 1988, p. 245–54.

[2] Gluckman J, Nayar S, Thoresz K. Real-time omnidirectional and panoramic stereo. In: DARPA Image Understanding Workshop. Monterey, USA; 1998,.

[3] Jang G, Kim S, Kweon I. Single camera catadioptric stereo system. In: OmniVis, workshop of ICCV 2005. Beijing, China; 2005,.

[4] Luo C, Su L, Zhu F. A novel omnidirectional stereo vision system via a single camera. In: Scene Reconstruction Pose Estimation and Tracking. InTech; 2007, p. 19 – 38.

[5] Mouaddib E, Sagawa R, Echigo T, Yagi Y. Stereo vision with a single camera and multiple mirrors. In: IEEE Int. Conf. on Robotics and Automation. Barcelona, Spain; 2005, p. 800–5.

[6] Dequen G, Devendeville L, Mouaddib E. Stochastic local search for omnidirectional catadioptric stereovision design. In: Pattern Recognition and Image Analysis, Lecture Notes in Computer Science; vol. 4478. Girona, Spain: Springer; 2007, p. 404–11.

[7] Haralick RM, Lee C, Ottenberg K, Nolle M. Analysis and solutions of the three point perspective pose estimation problem. Tech. Rep.; Universitaet Hamburg; Hamburg, Germany; 1991.

[8] Quan L, Lan Z. Linear n-point camera pose determination. IEEE Trans Pattern Analysis and Machine Intelligence 1999;21(8):774–80.

[9] Paulino A, Araujo H. Pose estimation for central catadioptric systems: An analytical approach. In: IEEE Int. Conf. on Pattern Recognition; vol. 3. Washington, DC, USA; 2002, p. 696–9.

[10] Dhome M, Richetin M, Lapreste JT. Determination of the attitude of 3d objects from a single perspective view. IEEE Trans on Pattern Analysis and Machine Intelligence 1989;11:1265–78.

[11] Ansar A, Daniilidis K. Linear pose estimation from points or lines. In: European Conf. on Computer Vision; vol. 25. Copenhagen, Denmark; 2002, p. 209–13.

[12] Fischler M, Bolles R. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 1981;24(6):381–95.

[13] Hmam H, Kim J. Optimal non-iterative pose estimation via convex relaxation. Image and Vision Computing 2010;11:1515–23.

[14] Lepetit V, Moreno-Noguer F, Fua P. EPnP: An accurate O(n) solution to the PnP problem. Int Journal on Computer Vision 2009;81(2):155–66.

[15] Malis E. Improving vision-based control using efficient second-order minimization techniques. In: IEEE Int. Conf. on Robotics and Automation; vol. 2. New Orleans, USA; 2004, p. 1843–8.

[16] Breitenreicher D, Schnörr C. Model-based multiple rigid object detection and registration in unstructured range data. Int J Comput Vision 2011;92:32–52.

[17] Pottmann H, Huang QX, Yang YL, Hu SM. Geometry and convergence analysis of algorithms for registration of 3d shapes. Int J Comput Vision 2006;67:277–96.

[18] Hartley RI, Kahl F. Global optimization through rotation space search. Int J Comput Vision 2009;82:64–79.

[19] Lowe D. Fitting parameterized three-dimensional models to images. IEEE Trans on Pattern Analysis and Machine Intelligence 1991;13:441–50.

[20] DeMenthon D, Davis L. Model-based object pose in 25 lines of code. Int Journal of Computer Vision 1995;15:123–41.

[21] Phong T, Horaud R, Yassine A, Tao P. Object pose from 2-D to 3-D point and line correspondences. Int Journal on Computer Vision 1995;15:225–43.

[22] Sundareswaran V, Behringer R. Visual servoing based augmented reality. San-Franscisco, USA; 1998, p. 193–200.

[23] Marchand E, Chaumette F. Virtual visual servoing: A framework for real-time augmented reality. Computer Graphics Forum 2002;21(3):289–98.

[24] Comport A, Marchand E, Pressigout M, Chaumette F. Real-time markerless tracking for augmented reality: the virtual visual servoing framework. IEEE Trans on Visualization and Computer Graphics 2006;12(4):615–28.

[25] Dionnet F, Marchand E. Robust stereo tracking for space robotic applications. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. San Diego, California; 2007, p. 3373–8.

[26] Barreto J. General central projection systems: modeling, calibration and visual servoing. Ph.D. thesis; University of Coimbra; 2003.

[27] Marchand E, Chaumette F. Fitting 3D models on central catadioptric images. In: IEEE Int. Conf. on Robotics and Automation. Roma, Italy; 2007, p. 52–8.

[28] Huber PJ. Robust statistics. New York, USA: Wiley; 1981.

[29] Drummond T, Cipolla R. Real-time visual tracking of complex structures. IEEE Trans on Pattern Analysis and Machine Intelligence 2002;24(7):932 – 46.

[30] Baker S, Nayar SK. A theory of single-viewpoint catadioptric image formation. Int Journal on Computer Vision 1999;35(2):175–96.

[31] Barreto JP, Araujo H. Issues on the geometry of central catadioptric imaging. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition; vol. 2. Hawaii, USA; 2001, p. 422–7.

[32] Geyer C, Daniilidis K. A unifying theory for central panoramic systems and practical applications. In: European Conf. on Computer Vision. Dublin, Ireland; 2000,.

[33] Bouthemy P. A maximum likelihood framework for determining moving edges. IEEE Trans on Pattern Analysis and Machine Intelligence 1989;11(5):499–511.

[34] Caron G, Marchand E, Mouaddib E. 3D model based pose estimation for omnidirectional stereovision. In: IEEE/RSJ International Conference on Intelligent RObots and Systems. St. Louis, Missouri, USA; 2009, p. 5228–33.

[35] Sturm P, Gargallo P. Conic fitting using the geometric distance. In: Asian Conf. on Computer vision. Berlin, Heidelberg: Springer-Verlag. ISBN 3-540-76389-9, 978-3-540-76389-5; 2007, p. 784–95.

[36] Espiau B, Chaumette F, Rives P. A new approach to visual servoing in robotics. IEEE Trans on Robotics and Automation 1992;8(3):313–26.

[37] Stewart C. Robust parameter estimation in computer vision. SIAM Rev 1999;41:513–37.

[38] Comport A, Marchand E, Chaumette F. Statistically robust 2d visual servoing. IEEE Trans on Robotics 2006;22(2):415–21.

[39] Ma Y, Soatto S, Košecká J, Sastry S. An invitation to 3D vision. Springer; 2004.

[40] Andreff N, Espiau B, Horaud R. Visual servoing from lines. Int Journal of Robotics Research 2002;21(8):669–99.

[41] Marchand E, Spindler F, Chaumette F. Visp for visual servoing: a generic software platform with a wide class of robot control skills. IEEE Robotics and Automation Magazine 2005;12(4):40–52.

[42] Caron G, Marchand E, Mouaddib E. Single viewpoint stereoscopic sensor calibration. In: International Symposium on I/V Communications and Mobile Networks, ISIVC. Rabat, Morocco; 2010, p. 1–4.

[43] Caron G, Marchand E, Mouaddib E. Tracking planes in omnidirectional stereovision. In: IEEE Int. Conf. on Robotics and Automation, ICRA'11. Shanghai, China; 2011, p. 6306–11.