Spherical Visual Gyroscope for Autonomous Robots using the Mixture of Photometric Potentials

Guillaume Caron, Fabio Morbidi

Abstract— **In this paper, we present a new direct omnidirectional visual gyroscope for mobile robotic platforms. The gyroscope estimates the 3D orientation of a camera-robot by comparing the current spherical image with that acquired at a reference pose. By transforming pixel intensities into a *Mixture of Photometric Potentials*, we introduce a novel image-similarity measure which can be seamlessly integrated into a classical nonlinear least-squares optimization scheme, offering an extended convergence domain. Our method provides accurate and robust attitude estimates, and it is easy-to-use since it involves a single tuning parameter, the width of the photometric potentials (Gaussian functions, in this work) controlling the power of attraction of each pixel. The visual gyroscope has been successfully tested on spherical image sequences generated by a twin-fisheye camera mounted on the end-effector of a robot arm and on a fixed-wing UAV.**

I. Introduction

A. Motivation and related work

Cameras provide informative clues about the shape and appearance of the surrounding environment, and they are pervasively used today in mobile robotics, because of their small form factor, reduced weight, and low cost. A classical problem in robotics is the estimation of the orientation of a camera between a reference and a query image. *Visual compasses* estimate the yaw angle of the camera-robot, *visual gyroscopes* provide the full attitude (roll, pitch and yaw angles), while *visual odometers* exploit the entire sequence of images between the reference and query, for estimating the robot's pose (position and attitude) [1].

With the advent of *omnidirectional cameras*, the field of view and workspace of an autonomous robot have been enlarged. Omnidirectional images can be obtained by combining a catadioptric system [2] or a fisheye lens [3] with a perspective camera. In particular, the mirrors and lenses can be arranged so that the full sphere around a single viewpoint is covered by a unique matrix of photosensitive sensors [4]. *Spherical images* can be also obtained with motorized cameras (see [5] for a survey), or by mounting a set of identical perspective cameras on the surface of a regular polygon, as in the Immersive Media 2360 Dodeca and in the Point Grey Ladybug 5 cameras, or on the surface of a sphere, as in the Panono 360° camera. Some work has been recently done on new spherical representations for image processing ("Spherepix" data structure) [6] and attitude estimation [7]. Technology companies have also shown a growing interest in the market of consumer omnidirectional cameras (see, e.g. the 360° video stabilization method in [8] or the upright adjustment algorithm for spherical panoramas in [9]), which will likely entail a resurgence of interest in robot-vision applications.

The design of omnidirectional visual gyroscopes is a well-entrenched subject in mobile robotics. The methods in [10]–[13] all rely on image features (e.g. the image projection of 3D parallel lines, vanishing points, SIFT features) for camera attitude estimation. However, feature extraction, matching and tracking are nontrivial image-processing steps which are sensitive to noise and illumination changes. Another limitation of feature-based methods is that they generally rely on strong assumptions about the structure of the 3D environment: for example, the methods in [11], [12] only work in man-made environments where parallel lines are abundant ("Atlanta world" assumption).

To overcome these issues, the *pixel intensity* of the overall image, interpreted as a 2D signal, can be exploited for estimating the attitude of a camera-robot. In fact, processing redundant information leads to superior accuracy and robustness. This approach is referred to as *direct* or global, and two families of strategies have emerged in the literature: "appearance-based" methods in the standard image domain [14], and methods based on harmonic analysis in the spatial frequency domain. By interpreting the rows of the panoramic cylinder as unidimensional signals, the authors in [15] computed the Fourier components of the image and used them for visual navigation. Other researchers have exploited the Fourier transform defined on the two-sphere and on the special Euclidean/orthogonal groups for 2D [16] and 3D rigid-motion estimation [17]. Along these lines, in [18] we proposed a weakly-calibrated omnidirectional visual compass based on the phase correlation method in the 2D Fourier domain, and we showed that it compares favorably with [12] and a brute-force photometric approach.

It is finally worth mentioning here the recent emergence of *hybrid solutions* [19], [20] which inherit some of the advantages of the feature-based and direct methods.

B. Original contributions, organization and notation

In this paper, we propose a direct omnidirectional visual gyroscope which relies on a novel representation of spherical image intensities based on the *Mixture of Photometric Potentials* (MPP). This representation is founded on [21], where 2D photometric Gaussian mixtures were used to extend the convergence domain of a dense visual-servoing algorithm. Thanks to the simple analytical form of an MPP, an image-

The authors are with the MIS laboratory, Université de Picardie Jules Verne, 33 Rue Saint-Leu, 80039 Amiens Cedex, France. Emails: {guillaume.caron,fabio.morbidi}@u-picardie.fr

This work is carried out as part of the INTERREG VA FMA ADAPT project "*Assistive Devices for empowering disAbled People through robotic Technologies*", http://adapt-project.com/indexen.php. The Interreg FCE Programme is a European Territorial Cooperation programme that aims to fund high quality cooperation projects in the Channel border region between France and England. The Programme is funded by the European Regional Development Fund (ERDF).

similarity measure built upon it can be minimized via a classical Gauss-Newton or Levenberg-Marquardt optimization algorithm (in conjunction with an M-estimator, for increased robustness). The proposed visual gyroscope has some attractive properties. It is accurate, even with low-resolution spherical images, it has a large convergence domain, and it is easy to tune. In fact, for a fixed image resolution, the user has to select a single positive parameter controlling the width of the photometric potentials (Gaussian functions in this paper), and thus the "degree of influence" of each pixel on its neighbors. Unlike the majority of existing methods, our visual gyroscope works in unstructured environments, and it requires no image processing (e.g. spatial gradients are not needed). Finally, extensive real-world experiments suggest that it is robust to illumination changes and partial image occlusions.

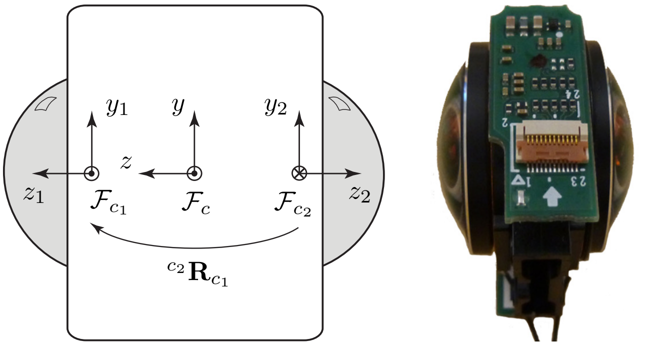
The rest of this paper is organized as follows. In Sect. II, we present the mathematical model of a spherical camera used in this work. In Sect. III, we show how to represent spherical image intensities as a mixture of photometric potentials. In Sect. IV, we introduce a cost function based on the MPP representation and describe the visual gyroscope. The results of real-world experiments conducted with a Ricoh Theta S camera mounted on the end-effector of an industrial robot and on a fixed-wing UAV are discussed in Sect. V. Finally, in Sect. VI, the main contributions of the paper are summarized and some possible directions for future research are outlined.

Notation: Throughout this work, we use the symbol $\mathbb{R}^n$ to denote the n-dimensional Euclidean space, $\mathbb{R}^{m\times n}$ the space of $m \times n$ matrices, $\mathbf{I}_{n\times n}$ the $n \times n$ identity matrix, and $\mathrm{SO}(3)$, $\mathrm{SE}(3)$ the 3D special orthogonal and special Euclidean groups, respectively. Given a square nonsingular matrix $\mathbf{A}$, we use $\mathbf{A}^{-1}$ to denote its inverse and $\det(\mathbf{A})$ its determinant. Finally, $\|\mathbf{x}\|$ indicates the Euclidean norm of $\mathbf{x} \in \mathbb{R}^n$, $[\mathbf{w}]_\times$ the skew-symmetric matrix associated with vector $\mathbf{w} \in \mathbb{R}^3$, and $\triangleq$ the equality by definition. ◇

II. Modeling of the spherical camera

It is well-known that some classes of fisheye cameras are approximately equivalent to a central catadioptric system [22]. They can then be modeled by using the unified central projection model, which involves two successive projections [23]: first, on a unit sphere and, then, perspectively, on the image plane.

In this paper, we will focus on spherical cameras including two identical ("twin") fisheye lenses facing in opposite directions, but whose axes are aligned (as in the Ricoh Theta and Samsung Gear 360 cameras, and as in [24]). Two sets of *intrinsic parameters* are then considered, one for each fisheye camera, $\mathscr{P}_{c_j} = \{\alpha_{u_j}, \alpha_{v_j}, u_{0_j}, v_{0_j}, \xi_j\}$, $j \in \{1, 2\}$, where α_{u_j} and α_{v_j} are the focal lengths in pixels in the horizontal and vertical direction, respectively, (u_{0_j}, v_{0_j}) are the coordinates of the principal point in pixels, and ξ_j is the distance between the unit sphere's first projection center and the perspective second projection center of fisheye camera j. Finally, *extrinsic parameters* are introduced to describe the 3D rotation between the reference frames $\mathcal{F}_{c_1}$ and $\mathcal{F}_{c_2}$ of the two fisheye cameras (see Fig. 1). By using an axis-angle representation with unit vector $\mathbf{w}_{1,2} \in \mathbb{R}^3$ and angle $\theta_{1,2}$ of the rotation, we collect the extrinsic parameters as the components of the axis-angle vector $\mathbf{r}_{1,2} = \theta_{1,2}\,\mathbf{w}_{1,2}$. Note that thanks to the Rodrigues' formula, the rotation matrix ${}^{c_2}\mathbf{R}_{c_1} \in \mathrm{SO}(3)$ between $\mathcal{F}_{c_1}$ and $\mathcal{F}_{c_2}$, can be easily computed from $\mathbf{w}_{1,2}$ and $\theta_{1,2}$.



(a) (b)

Fig. 1. A twin-fisheye camera, the Ricoh Theta: (a) Schematic representation (top view): To guarantee the single-viewpoint property, we assume that the translation vector between the frames $\mathcal{F}_{c_1}$ and $\mathcal{F}_{c_2}$ is *zero* (a non-zero baseline is shown in the figure for illustration purposes only), and that the camera frame $\mathcal{F}_c$ coincides with $\mathcal{F}_{c_1}$; (b) Top view of the actual camera without housing.

The ten intrinsic and three extrinsic parameters of the spherical camera can be simultaneously estimated from corner points extracted from images of a known calibration rig (see Sect. V-A). Once the calibration parameters have been determined, assuming that the reference frame $\mathcal{F}_c$ of the camera coincides with $\mathcal{F}_{c_1}$, the twin-fisheye to spherical warping is straightforward since any point ${}^c\mathbf{X}_\mathcal{S} = [{}^cX_\mathcal{S}, {}^cY_\mathcal{S}, {}^cZ_\mathcal{S}]^T$ lying on the unit sphere $\mathcal{S}^2$ can be expressed in $\mathcal{F}_{c_2}$ via ${}^{c_2}\mathbf{R}_{c_1}$, and projected, with parameters $\mathscr{P}_{c_j}$, $j \in \{1, 2\}$, into that part of the image which corresponds to the fisheye lens in which it is visible. Note that since the spherical projection of a 3D point ${}^c\mathbf{X} = [{}^cX, {}^cY, {}^cZ]^T$ in $\mathcal{F}_c$ onto a point ${}^c\mathbf{X}_\mathcal{S}$ belonging to the spherical image, can be expressed as ${}^c\mathbf{X}_\mathcal{S} = {}^c\mathbf{X}/\|{}^c\mathbf{X}\|$, without loss of generality, we can restrict ourselves to a sphere with *unit* radius. In particular, we can assume that ${}^c\mathbf{X}_\mathcal{S}$ lies directly on the surface of the sphere of the unified central projection model [23].

In this work, a spherical image is represented as a *uniformly-spaced* set of points (pixels) on the surface of a sphere (i.e. in a geodesic sense). For the discrete set of points ${}^c\mathbf{X}_{\mathcal{S}i}$, $i \in \{1, 2, \ldots, P\}$, to be uniformly spaced on the surface of the sphere, we start with a convex regular icosahedron (a polyhedron with 20 equilateral triangle faces and 12 vertices, see Fig. 2(a)). Its faces undergo a number of subdivisions N which is proportional to the desired resolution of the spherical image [25]. Note that the vertices of a convex regular icosahedron are evenly distributed over a unit sphere circumscribing the polyhedron, and that this property is still valid for an icosahedron whose faces have been recursively split (see Figs. 2(b)-2(d)). For N subdivision levels, the corresponding polyhedron has $P = \frac{1}{2}(20\times 4^N)+2$ vertices and 20×4^N faces.

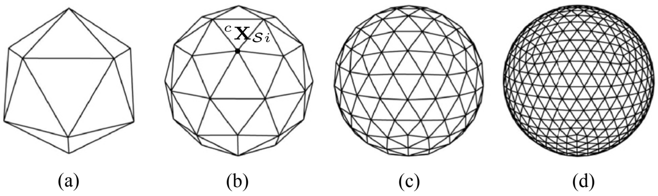


Fig. 2. The recursive subdivision of an icosahedron provides a sphere of uniform triangulation (image adapted from Kooima *et al.*, *IEEE Trans. Vis. Comput. Gr.*, Sep. 2009). (a) Icosahedron; (b), (c) and (d), icosahedron subdivided one, two, and three times, respectively. At each subdivision, each face is split into four equilateral triangles, followed by a re-projection of the vertices onto the unit sphere. In (b), the generic vertex ${}^c\mathbf{X}_{\mathcal{S}i}$ is shown.

III. Mixture of photometric potentials (MPP)

Our goal in this paper, is to design a visual gyroscope which estimates the orientation between a spherical image $I_{\mathcal{S}}$ acquired by the camera at the current time t, and a reference spherical image $I^*_{\mathcal{S}}$. The relative orientation between $I_{\mathcal{S}}$ and $I^*_{\mathcal{S}}$ will be represented by the axis-angle vector $\mathbf{r} = \theta\,\mathbf{w}$, where $\|\mathbf{w}\| = 1$. Since $\mathbf{r}$ minimizes the difference between $I_{\mathcal{S}}(\mathbf{r})$ and $I^*_{\mathcal{S}}$, we can write the classical *Sum-of-Squared-Differences* (SSD) cost function:

$$C_{\text{SSD}}(\mathbf{r}) = \|\boldsymbol{\mathcal{I}}_{\mathcal{S}}(\mathbf{r}) - \boldsymbol{\mathcal{I}}^*_{\mathcal{S}}\|, \qquad (1)$$

where the pixel intensities of the grayscale images $I_{\mathcal{S}}(\mathbf{r})$ and $I^*_{\mathcal{S}}$ at points ${}^c\mathbf{X}_{\mathcal{S}i}$, $i \in \{1, 2, \ldots, P\}$, have been collected as the components of the vectors $\boldsymbol{\mathcal{I}}_{\mathcal{S}}(\mathbf{r})$, $\boldsymbol{\mathcal{I}}^*_{\mathcal{S}} \in \mathbb{R}^P$, respectively. While intuitive and easy to relate to the rotational degrees of freedom (DOFs) of the camera (in the case of hemispherical images, cf. [26]), $C_{\text{SSD}}(\mathbf{r})$ is known to have, in practice, a narrow convergence domain associated with its minima. This is a major drawback which justifies the introduction of a new spherical-image representation. We propose here to adapt the notion of 2D photometric Gaussian mixture (originally introduced in [21] for visual servo control with a perspective camera), to enlarge the basin of convergence of our visual gyroscope. In this way, near-full coverage of SO(3) will be achieved (see Sect. V).

Recall that the *mixture density function* $G(\mathbf{x})$ of m multivariate heteroscedastic Gaussian probability density functions (pdfs) is defined as [27, Ch. 3]:

$$G(\mathbf{x}) = \sum_{i=1}^{m} w_i\,\phi(\mathbf{x};\,\boldsymbol{\mu}_i,\,\boldsymbol{\Sigma}_i), \qquad (2)$$

where

$$\phi(\mathbf{x};\,\boldsymbol{\mu}_i,\,\boldsymbol{\Sigma}_i) = \frac{1}{\sqrt{(2\pi)^n \det(\boldsymbol{\Sigma}_i)}}\,\exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\,\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)), \qquad (3)$$

denotes the Gaussian pdf of a random vector $\mathbf{x} \in \mathbb{R}^n$ with mean vector $\boldsymbol{\mu}_i \in \mathbb{R}^n$ and positive-definite covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{n\times n}$, $i \in \{1, \ldots, m\}$, and,

$$\sum_{i=1}^{m} w_i = 1, \quad w_i \geq 0, \quad i \in \{1, \ldots, m\}. \qquad (4)$$

Owing to this definition, $G(\mathbf{x})$ in (2) is the *convex combination* of m Gaussian pdfs with mixing weights $w_1, \ldots, w_m$. Inspired by (2) and extending the representation in [21] to spherical images ($n = 3$), we define the *Mixture of Photometric Potentials* (MPP) at reference point ${}^c\mathbf{X}_{\mathcal{S}g}$ as (see Fig. 3):

$$G_{\mathcal{S}}({}^c\mathbf{X}_{\mathcal{S}g},\,I_{\mathcal{S}}) = \sum_{i=1}^{P} \overline{\boldsymbol{\mathcal{I}}}_{\mathcal{S}}({}^c\mathbf{X}_{\mathcal{S}i})\,\frac{1}{\lambda_g^3\,(2\pi)^{3/2}}\,\exp(-\,\frac{(\mathrm{D}^c\mathbf{X}_{\mathcal{S}})^2}{2\lambda_g^2}), \qquad (5)$$

where $\mathrm{D}^c\mathbf{X}_{\mathcal{S}} = \arccos\big({}^c\mathbf{X}^T_{\mathcal{S}g}\,{}^c\mathbf{X}_{\mathcal{S}i}\big)$ is the geodesic distance between ${}^c\mathbf{X}_{\mathcal{S}g}$ and ${}^c\mathbf{X}_{\mathcal{S}i}$ on the unit sphere $\mathcal{S}^2$ and $\overline{\boldsymbol{\mathcal{I}}}_{\mathcal{S}}({}^c\mathbf{X}_{\mathcal{S}i}) \geq 0$ denotes the *normalized image intensity* at point ${}^c\mathbf{X}_{\mathcal{S}i}$ such that $\sum_{i=1}^{P} \overline{\boldsymbol{\mathcal{I}}}_{\mathcal{S}}({}^c\mathbf{X}_{\mathcal{S}i}) = 1$ (cf. equation (4)). The *expansion parameter* $\lambda_g > 0$ can be used to adjust the width of the Gaussians appearing in (5).

Before proceeding with the design of a visual gyroscope based on the MPP representation (see Sect. IV), it is worth pointing out here two important differences between equation (2) and (5):

- In (5), we set $\boldsymbol{\mu}_i = {}^c\mathbf{X}_{\mathcal{S}i}$ and $\boldsymbol{\Sigma}_i = \lambda_g^2\,\mathbf{I}_{3\times 3}$, $\forall\, i$, i.e. we restricted ourselves to a *homoscedastic isotropic mixture* [27, Sect. 3.3]. The squared Mahalanobis distance $(\mathbf{x}-\boldsymbol{\mu}_i)^T\,\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)$ appearing in (3) reduces to a squared weighted geodesic distance in our case.
- While intuition is borrowed from probability and statistics, there is nothing stochastic about the spherical-image representation in (5). In fact, we do not exploit the fact that $G_{\mathcal{S}}({}^c\mathbf{X}_{\mathcal{S}g},\,I_{\mathcal{S}})$ might represent a pdf.

***Remark** 1 (Generality of the MPP representation):* Note that (5) is an instance of a more general spherical-image representation. In fact, the Gaussians appearing in (5) are used to control the *power of attraction* (or radius of influence) of each pixel ${}^c\mathbf{X}_{\mathcal{S}g}$ in the image, and they can be replaced by other (infinitely-supported) kernel functions used in non-parametric statistics [28]. ◇

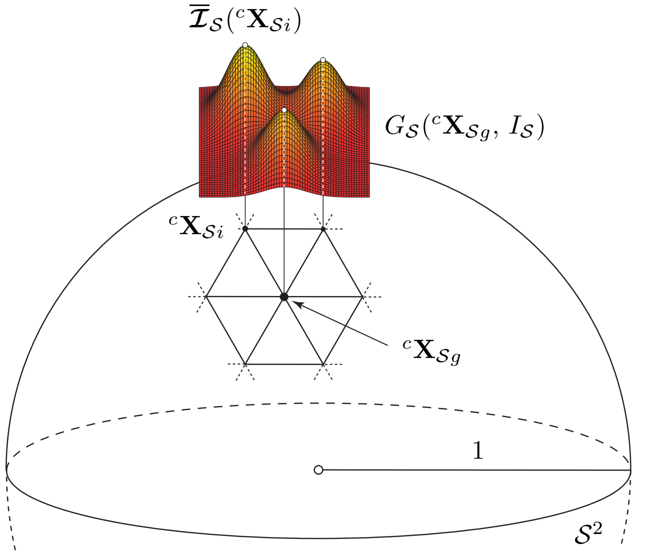

$\overline{\boldsymbol{\mathcal{I}}}_{\mathcal{S}}({}^c\mathbf{X}_{\mathcal{S}i})$
$G_{\mathcal{S}}({}^c\mathbf{X}_{\mathcal{S}g},\,I_{\mathcal{S}})$
${}^c\mathbf{X}_{\mathcal{S}i}$
${}^c\mathbf{X}_{\mathcal{S}g}$
1
$\mathcal{S}^2$

Fig. 3. Graphical illustration of the MPP (5) at point ${}^c\mathbf{X}_{\mathcal{S}g}$. Note that the figure only provides an intuitive description of the MPP representation, since the Gaussian functions in (5) live in a 4D space and they cannot be visualized.

IV. Visual gyroscope based on the MPP representation

Let ${}^{c}\mathbf{R}_{c*} \in \mathrm{SO}(3)$ represent the *unknown* rotation between the reference image $I^*_{\mathcal{S}}$ and the current image $I_{\mathcal{S}}$ that our visual gyroscope intends to estimate. By leveraging (5), we can introduce the following MPP at point ${}^{c}\mathbf{X}_{\mathcal{S}g}$:

$$G_{\mathcal{S}}({}^{c}\mathbf{X}_{\mathcal{S}g}, I_{\mathcal{S}}, {}^{c}\mathbf{R}_{c*}) \triangleq \sum_{i\,=\,1}^{P} H_{\mathcal{S}}({}^{c}\mathbf{X}_{\mathcal{S}g}, {}^{c}\mathbf{X}_{\mathcal{S}i}, I_{\mathcal{S}}, {}^{c}\mathbf{R}_{c*})$$

$$= \sum_{i\,=\,1}^{P} \overline{\mathcal{I}}_{\mathcal{S}}({}^{c}\mathbf{X}_{\mathcal{S}i})\, \frac{1}{\lambda_g^3\,(2\pi)^{3/2}}\, \exp(-\frac{(\mathrm{D}\,{}^{c}\mathbf{X}_{\mathcal{S}})^2}{2\lambda_g^2}),$$

where ${}^{c}\mathbf{X}_{\mathcal{S}g} = {}^{c}\mathbf{R}_{c*}{}^{c*}\mathbf{X}_{\mathcal{S}g}$, ${}^{c}\mathbf{X}_{\mathcal{S}i} = {}^{c}\mathbf{R}_{c*}{}^{c*}\mathbf{X}_{\mathcal{S}i}$ and $\|{}^{c*}\mathbf{X}_{\mathcal{S}g}\| = \|{}^{c*}\mathbf{X}_{\mathcal{S}i}\| = 1$. We can then replace the SSD cost function in (1) with,

$$C_{\text{MPP}}(\mathbf{r}) \triangleq \|\boldsymbol{\mathcal{C}}_{\text{MPP}}(\mathbf{r})\| = \|\mathbf{G}_{\mathcal{S}}(I_{\mathcal{S}}, {}^{c}\mathbf{R}_{c*}(\mathbf{r})) - \mathbf{G}_{\mathcal{S}}(I^*_{\mathcal{S}})\|, \tag{6}$$

where $\mathbf{G}_{\mathcal{S}}(I_{\mathcal{S}}, {}^{c}\mathbf{R}_{c*}(\mathbf{r}))$, $\mathbf{G}_{\mathcal{S}}(I^*_{\mathcal{S}}) \in \mathbb{R}^P$ respectively stack the values of $G_{\mathcal{S}}({}^{c}\mathbf{X}_{\mathcal{S}g}, I_{\mathcal{S}}, {}^{c}\mathbf{R}_{c*}(\mathbf{r}))$ and $G_{\mathcal{S}}(I^*_{\mathcal{S}})$ at the P pixels of the spherical images, and ${}^{c}\mathbf{R}_{c*}(\mathbf{r})$ indicates that the rotation matrix has been computed from the axis-angle vector $\mathbf{r}$. An estimate $\widehat{\mathbf{r}}$ of $\mathbf{r}$ can be obtained by numerically minimizing (6) with the Gauss-Newton algorithm. If we initialize this iterative algorithm with $\mathbf{r}^{(0)}$ (step 0), to compute $\mathbf{r}^{(k+1)}$ from $\mathbf{r}^{(k)}$ we need to find ${}^{c^{(k+1)}}\mathbf{R}_{c*}$ by composing the rotation matrices determined from $\mathbf{r}^{(k)}$ and the increment $\Delta\mathbf{r}^{(k)}$, i.e.

$${}^{c^{(k+1)}}\mathbf{R}_{c*} = {}^{c^{(k+1)}}\mathbf{R}_{c^{(k)}}(\Delta\mathbf{r}^{(k)})\; {}^{c^{(k)}}\mathbf{R}_{c*}(\mathbf{r}^{(k)}), \tag{7}$$

with $k \in \{0, 1, \ldots\}$. On the other hand, $\Delta\mathbf{r}^{(k)}$ can be determined from the first-order Taylor expansion of $\boldsymbol{\mathcal{C}}_{\text{MPP}}(\mathbf{r})$ about $\widehat{\mathbf{r}}$,

$$\boldsymbol{\mathcal{C}}_{\text{MPP}}(\,\widehat{\mathbf{r}}\,) \simeq \boldsymbol{\mathcal{C}}_{\text{MPP}}(\mathbf{r}^{(k)}) + \frac{\partial\, \boldsymbol{\mathcal{C}}_{\text{MPP}}(\mathbf{r}^{(k)})}{\partial\, \mathbf{r}^{(k)}}\, \Delta\mathbf{r}^{(k)}. \tag{8}$$

In fact, by setting (8) to zero, we have,

$$\Delta\mathbf{r}^{(k)} = -\,\gamma \left[\frac{\partial\, \boldsymbol{\mathcal{C}}_{\text{MPP}}(\mathbf{r}^{(k)})}{\partial\, \mathbf{r}^{(k)}}\right]^{\dagger} \boldsymbol{\mathcal{C}}_{\text{MPP}}(\mathbf{r}^{(k)}), \tag{9}$$

where $[\,\cdot\,]^{\dagger}$ denotes the (Moore-Penrose) left pseudo-inverse and γ is a positive gain. Thus, equation (7) allows one to update the estimated orientation of the camera at each step k until convergence. Note that since $\mathbf{G}_{\mathcal{S}}(I^*_{\mathcal{S}})$ is constant, the $P \times 3$ Jacobian matrix in (9) is,

$$\frac{\partial\, \boldsymbol{\mathcal{C}}_{\text{MPP}}(\mathbf{r}^{(k)})}{\partial\, \mathbf{r}^{(k)}} = \frac{\partial\, \mathbf{G}_{\mathcal{S}}(I_{\mathcal{S}}, {}^{c^{(k)}}\mathbf{R}_{c*})}{\partial\, \mathbf{r}^{(k)}}. \tag{10}$$

If we apply the chain rule to (10) and combine like factors, we obtain:

$$\frac{\partial\, \boldsymbol{\mathcal{C}}_{\text{MPP}}(\mathbf{r}^{(k)})}{\partial\, \mathbf{r}^{(k)}} = \begin{bmatrix} \vdots \\ \left(\displaystyle\sum_{i=1}^{P} \frac{\partial H_{\mathcal{S}}}{\partial\, {}^{c^{(k)}}\mathbf{X}_{\mathcal{S}g}}\right) \dfrac{\partial\, {}^{c^{(k)}}\mathbf{X}_{\mathcal{S}g}}{\partial\, \mathbf{r}^{(k)}} \\ \vdots \end{bmatrix}. \tag{11}$$

It now remains to compute the two partial derivatives appearing on the right-hand side of (11). The partial derivative of $H_{\mathcal{S}}$ with respect to ${}^{c^{(k)}}\mathbf{X}_{\mathcal{S}g}$, a 1×3 vector, is given by:

$$\frac{\partial\, H_{\mathcal{S}}}{\partial\, {}^{c^{(k)}}\mathbf{X}_{\mathcal{S}g}} = \frac{\overline{\mathcal{I}}_{\mathcal{S}}({}^{c^{(k)}}\mathbf{X}_{\mathcal{S}i})}{\lambda_g^5\,(2\pi)^{3/2}}\, \mathrm{D}\,{}^{c^{(k)}}\mathbf{X}_{\mathcal{S}}\, \exp(-\frac{(\mathrm{D}\,{}^{c^{(k)}}\mathbf{X}_{\mathcal{S}})^2}{2\,\lambda_g^2})$$

$$\cdot \frac{{}^{c^{(k)}}\mathbf{X}^T_{\mathcal{S}i}}{\sqrt{1 - ({}^{c^{(k)}}\mathbf{X}^T_{\mathcal{S}g}\,{}^{c^{(k)}}\mathbf{X}_{\mathcal{S}i})^2}}\,,$$

where ${}^{c^{(k)}}\mathbf{X}_{\mathcal{S}g} = {}^{c^{(k)}}\mathbf{R}_{c*}{}^{c*}\mathbf{X}_{\mathcal{S}g}$, ${}^{c^{(k)}}\mathbf{X}_{\mathcal{S}i} = {}^{c^{(k)}}\mathbf{R}_{c*}{}^{c*}\mathbf{X}_{\mathcal{S}i}$. The partial derivative of ${}^{c^{(k)}}\mathbf{X}_{\mathcal{S}g}$ with respect to $\mathbf{r}^{(k)}$, a 3×3 matrix, can be obtained from the well-known formula relating the velocity of a 3D point to the spatial velocity of a camera, considering *zero* translational speed,

$${}^{c^{(k)}}\dot{\mathbf{X}}_{\mathcal{S}g} = -\,\boldsymbol{\omega}_c \times {}^{c^{(k)}}\mathbf{X}_{\mathcal{S}g} = \left[{}^{c^{(k)}}\mathbf{X}_{\mathcal{S}g}\right]_{\times} \boldsymbol{\omega}_c\,,$$

where $\boldsymbol{\omega}_c \in \mathbb{R}^3$ denotes the angular velocity of the camera frame $\mathcal{F}_c$, which corresponds to the increment $\Delta\mathbf{r}^{(k)}$ in the notation of this section. In conclusion, we have that:

$$\frac{\partial\, {}^{c^{(k)}}\mathbf{X}_{\mathcal{S}g}}{\partial\, \mathbf{r}^{(k)}} = \left[{}^{c^{(k)}}\mathbf{X}_{\mathcal{S}g}\right]_{\times}.$$

***Remark** 2 (Spherical-image processing):* The proposed MPP representation is computationally inexpensive, since it does not involve any image-processing step. For example, differently from the existing direct methods, one need not to compute the spatial gradient of the spherical image (which is known to be a challenging task [6]). ◇

***Remark** 3 (Tuning of the visual gyroscope):* The choice of the expansion parameter λ_g, the *only* tuning parameter of the gyroscope (for N fixed), depends on the appearance of the 3D environment where the camera-robot moves. While there do not exist general guidelines, it has been experimentally observed that for $N \geq 3$, the bigger λ_g, the larger the convergence domain of the Gauss-Newton algorithm (see Sect. V-B for more details). ◇

***Remark** 4 (Levenberg-Marquardt algorithm):* To improve the convergence performance, one might replace the Gauss-Newton algorithm with the Levenberg-Marquardt optimization scheme. In this case, the update law (9) becomes:

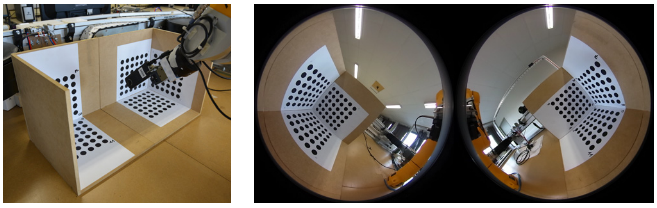
$$\Delta\mathbf{r}^{(k)} = -\,\gamma\,\left(\boldsymbol{J}^T\boldsymbol{J} + \nu\,\mathrm{diag}(\boldsymbol{J}^T\boldsymbol{J})\right)^{-1}\boldsymbol{J}^T\boldsymbol{\mathcal{C}}_{\text{MPP}}(\mathbf{r}^{(k)}),$$

where $\boldsymbol{J} \triangleq \partial\boldsymbol{\mathcal{C}}_{\text{MPP}}(\mathbf{r}^{(k)})/\partial\mathbf{r}^{(k)}$, $\mathrm{diag}(\boldsymbol{J}^T\boldsymbol{J})$ is the diagonal matrix given by the diagonal entries of $\boldsymbol{J}^T\boldsymbol{J}$ and $\nu > 0$ is a damping factor. A large value of ν (e.g. $\nu = 1$) yields the gradient-descent method, while a small value (e.g. $\nu = 10^{-3}$) results essentially in the Gauss-Newton update. ◇

V. Experimental validation

A. Calibration of the spherical camera

The proposed visual gyroscope has been experimentally validated using the *Ricoh Theta S* camera. Spherical images are created by two fisheye lenses mounted back to back, collecting light at a view angle that far exceeds 180° (cf. Fig. 1). The light is allocated to the two image sensors using two 90° prisms. In our experiments, we considered



(a) (b)

Fig. 4. Calibration of the Ricoh Theta camera: (a) The calibration rig consists of six checkerboard patterns glued inside two half cubes; (b) Dual-fisheye image of the calibration rig. The camera is held by a robot arm.

1280 pixels × 720 pixels images in the dual-fisheye format[1] (i.e. two hemispheric images, see Fig. 4(b)). We applied a mask to these images in order to select those areas that change over time and that are thus informative for our gyroscope. For the calibration, we modeled the Ricoh Theta as a stereo system with two fisheye cameras having the same center of projection. As shown in Fig. 4(a), our calibration rig consists of six checkerboard patterns glued inside two half cubes. A variant of the calibration algorithm proposed in [29] was considered: we set the translation parameters between the two fisheye cameras to zero and estimated the rotation parameters only. This yielded $\mathscr{P}_{c_1} = \{577.7741, 576.1130, 958.6632, 316.8989, 1.9878\}$, $\mathscr{P}_{c_2} = \{567.8953, 565.1663, 321.5507, 319.4833, 1.9392\}$, and $\mathbf{r}_{1,2} = [-0.0082, 3.1319, -0.0108]^T$ rad.

B. Industrial robot

In order to have pure rotations about the optical center, the camera was mounted on the end-effector of a 6 DOF Stäubli TX60 robot located in a 10.05 m × 7.03 m × 2.70 m room with neon lighting (see Fig. 5). We used the Tsai & Lenz's algorithm in the ViSP-library[2] implementation, to extrinsically calibrate the camera with respect to the robot end-effector, i.e. to compute ${}^{e}\mathbf{M}_c \in \mathrm{SE}(3)$, the rigid transformation between the camera frame $\mathcal{F}_c$ and the end-effector frame $\mathcal{F}_e$. The calibration rig was observed by the camera from six different poses (see Fig. 4(b)), leading to:

$$ {}^{e}\mathbf{M}_c = \begin{bmatrix} -0.0136 & 0.9997 & 0.0222 & -0.0401 \\ 0.0142 & -0.0220 & 0.9996 & 0.0000 \\ 0.9998 & 0.0139 & -0.0139 & 0.2372 \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

where the translation is expressed in meters. In our tests, we considered 5 Collection Points (CPs) located on the same plane parallel to the ground (the camera center being 60 cm above the base of the Stäubli robot). At each CP, the camera was rotated of 360° about the vertical axis, with a step size of 2.5°, yielding 144 images. The distance between CP0 and CP1, CP2, CP3 and CP4 is 40, 120, 240 and 400 mm, respectively. The encoders of the Stäubli robot provided us with an accurate ground truth for the validation of our visual gyroscope.

1) Yaw-angle estimation (1 DOF): To study the shape of the cost function C_{MPP} and to easily evaluate the impact of the expansion parameter λ_g and of the subdivision level N, on the magnitude of the angular estimation error, we rotated the camera about a *single axis*, the (vertical) x-axis of $\mathcal{F}_c$ (see Fig. 5). In our first series of tests, we focused on CP2 and ran the Levenberg-Marquardt algorithm (with $\gamma = 1$) in conjunction with a redescending M-estimator with Cauchy's score function [30, Sect. 4.8] to minimize $C_{\text{MPP}}(r)$ (as stopping criterion, we set a 10^{-6} threshold on the stability of residuals). For the sake of uniformity, $C_{\text{MPP}}(r)$ was normalized between 0 and 1 (henceforth denoted by $\overline{C}_{\text{MPP}}(r)$): in fact, $\max(C_{\text{MPP}}(r))$ varied over a large range (up to three orders of magnitude). Fig. 6 shows the width of the convergence domain of $\overline{C}_{\text{MPP}}(r)$ for different values of N and λ_g. In particular, Fig. 6(a) reports the MPP cost function for $N = 2$ and $\lambda_g = 0.01$ without the M-estimator, Fig. 6(b) for $N = 3$ and $\lambda_g = 0.4$ and Fig. 6(c) for $N = 5$ and $\lambda_g = 0.3$ with M-estimator. The width of the convergence domain is 115°, 312.5° and 360°, respectively (see the vertical dashed lines). Note that for $\lambda_g = 0.01$, the MPP cost function essentially reduces to the SSD cost function in (1). More insight into the shape of $\overline{C}_{\text{MPP}}(r)$ is provided by Figs. 6(d)-6(e) which report the width of the convergence domain against λ_g for $N \in \{2, 3, \ldots, 6\}$ (the values of λ_g considered are marked with a cross). The values of $\overline{C}_{\text{MPP}}(r)$ for $\lambda_g > 1$ are not shown in the figures, since we observed no changes with respect to the case of $\lambda_g = 1$. Some conclusions can be drawn from Fig. 6:

- The bigger N, the smoother $C_{\text{MPP}}(r)$ and the larger the convergence domain,
- The bigger λ_g, the larger the convergence domain: however, the effect vanishes for $\lambda_g > 0.5$ (and $N > 2$),
- The M-estimator has a "linearizing effect" on $C_{\text{MPP}}(r)$ (the effect is more pronounced for large λ_g's). As a consequence, the convergence radius might increase (especially for $\lambda_g \in [0.2, 0.4]$). However, for small N,

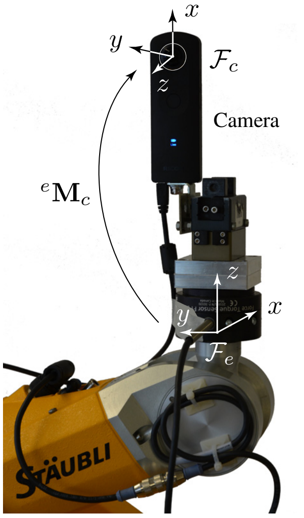
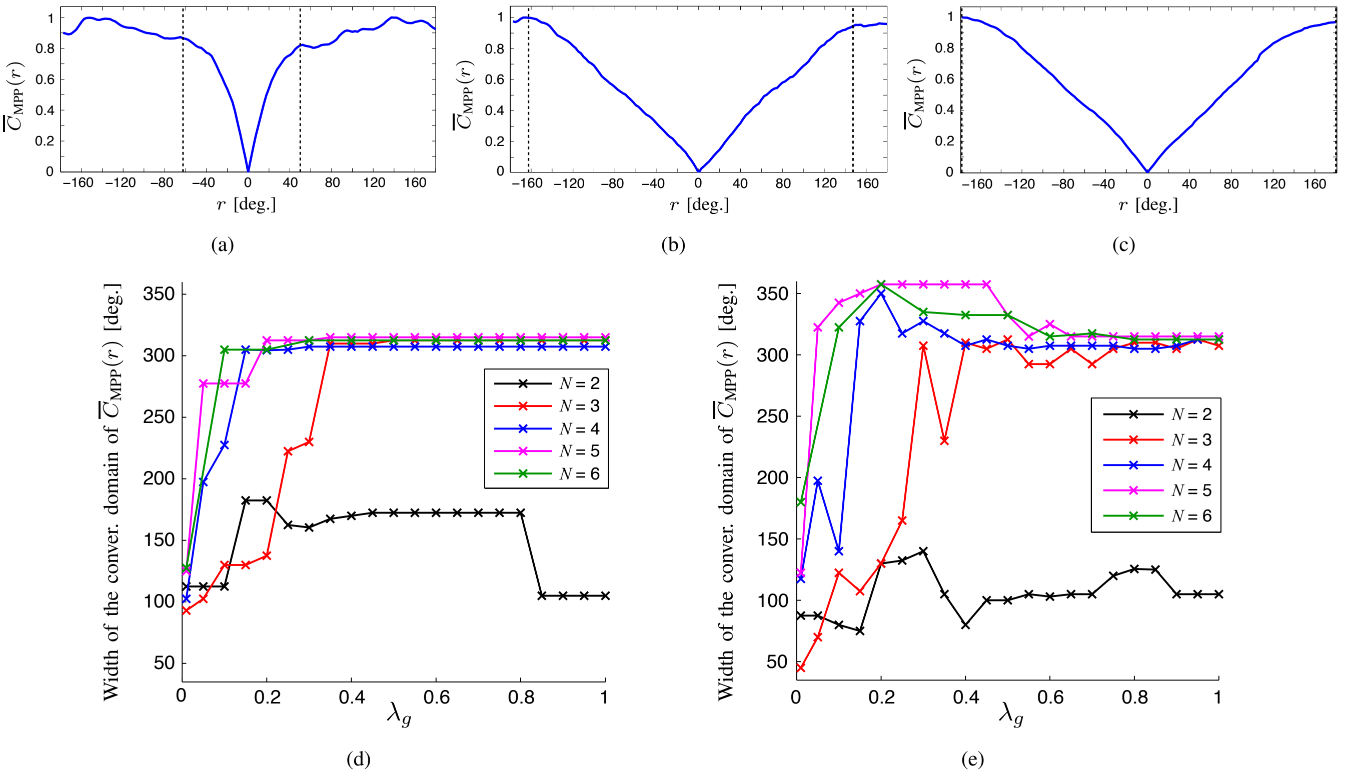


Fig. 5. The Ricoh Theta mounted on the end-effector of the Stäubli TX60 robot. ${}^{e}\mathbf{M}_c$ is the rigid transformation between the camera frame $\mathcal{F}_c$ and the end-effector frame $\mathcal{F}_e$.

[1] The full image dataset, called SVMIS, is available at: http://mis.u-picardie.fr/~g-caron/pub/data/SVMIS_dataset.zip

[2] https://visp.inria.fr



(a) (b) (c)

(d) (e)

Fig. 6. [*1 DOF Stäubli*] Normalized cost function $\overline{C}_{\text{MPP}}(r)$ for: (a) $N = 2$, $\lambda_g = 0.01$, without the M-estimator; (b) $N = 3$, $\lambda_g = 0.4$ with the M-estimator; (c) $N = 5$, $\lambda_g = 0.3$ with the M-estimator (the width of the convergence domain is indicated by the vertical dashed lines). (d), (e) Width of the convergence domain of $\overline{C}_{\text{MPP}}(r)$ against λ_g for $N \in \{2, \dots, 6\}$ without and with M-estimator, respectively.

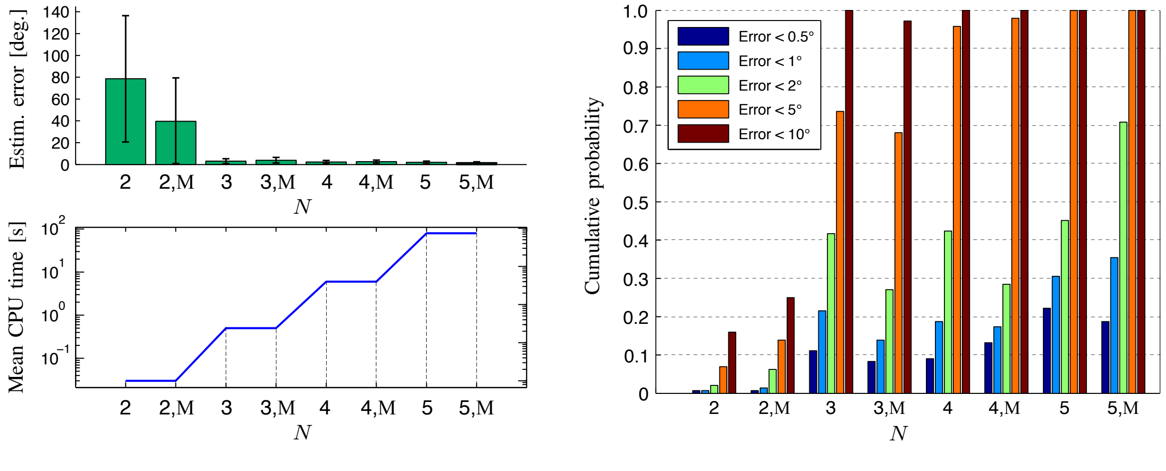
the M-estimator perturbs $C_{\text{MPP}}(r)$ as well, which might result in a loss of estimation accuracy (see the "bumps" in Fig. 6(a)),

- Under ideal conditions (i.e. no image noise), $C_{\text{MPP}}(r)$ is perfectly symmetrical about the origin.

The estimation accuracy and computational complexity of our gyroscope are evaluated in Fig. 7. In the experiments, we set $\lambda_g = 0.325$, $\gamma = 1$, and considered two candidate initial conditions, $r^{(0)} \in \{0, \pi\}$, retaining the one that minimizes $C_{\text{MPP}}(r)$. This strategy is simple but effective to reject outliers, and it noticeably reduces the estimation error. Fig. 7(a, *top*) reports the mean and standard deviation of the magnitude of the angular estimation error, and Fig. 7(a, *bottom*) the mean CPU time in seconds for $N \in \{2, 3, 4, 5\}$ with ("M") or without M-estimator (we ran the gyroscope on a MacBook Pro with 2.4 GHz Intel Core i7 processor and 8 GB RAM). Fig. 7(b) complements Fig. 7(a, *top*) by displaying the empirical cumulative distribution function obtained from the magnitude of the angular estimation error for $N \in \{2, 3, 4, 5\}$. The vertical bars in Fig. 7(b) show the ratio of the elements of the error vector whose value is smaller than 0.5°, 1°, 2°, 5° and 10°. For instance, if a 5°-error is allowed, about 75% of the estimates are acceptable for $N = 3$ and about 95% for $N = 4$. The latter result is all the more remarkable, since for $N = 4$ the gyroscope uses two 37 pixels × 37 pixels grayscale images (one per fisheye lens). Obviously, the bigger N, the more accurate the gyroscope, to the detriment of the computation time. In fact, since $P = \frac{1}{2}(20 \times 4^N) + 2$ (cf. Sect. II), the runtime grows exponentially with N. The gyroscope turned out to be also robust to occasional image occlusions caused by the motion of the robot arm.

A final study was performed to evaluate the impact of the *translational motion* on the estimation accuracy of the gyroscope. We tuned the Levenberg-Marquardt algorithm (with M-estimator) as in Fig. 7, and we estimated the angle between image 71 at CP0 and all the other images in CP0, CP1, ..., CP4. Fig. 8 reports the mean and standard deviation of the magnitude of the angular estimation error for a growing distance from CP0 and for $N \in \{3, 4, 5\}$. The translational motion has a negligible effect on the estimation error: however, for $N = 5$ (and higher) the gyroscope appears to be more sensitive to translations. For $N \in \{3, 4, 5\}$, the mean estimation error (over the 5 CPs) is 2.95°, 2.49°, 1.69°, respectively.

2) Attitude estimation (3 DOFs): The gyroscope was also used to estimate the three rotational DOFs of the Ricoh Theta. To this end, we considered a single collection point, CP2, where we obtained the maximum number of distinct 3D orientations of the camera (94 overall), which did not violate the mechanical constraints of the robot. We set $\lambda_g = 0.275$, $\gamma = 1$, and initialized the Gauss-Newton algorithm (without M-estimator) with $\mathbf{r}^{(0)} = \mathbf{0}$. Table I reports the statistics of the estimation error $\|\mathbf{r} - \widehat{\mathbf{r}}\|$ over the 94 images for $N \in \{3, 4, 5\}$, and the corresponding mean CPU time and mean number of iterations of the optimization algorithm.



(a)

(b)

Fig. 7. [*1 DOF Stäubli*] (a, *top*) Mean and standard deviation of the magnitude of the estimation error for $N \in \{2, 3, 4, 5\}$ with ("M") or without M-estimator; (a, *bottom*) mean CPU time for $N \in \{2, 3, 4, 5\}$ (note the logarithmic scale on the vertical axis); (b) Empirical cumulative distribution function obtained from the magnitude of the angular estimation error for $N \in \{2, 3, 4, 5\}$.

N	3	4	5
Mean error [deg.]	7.55	4.15	3.69
Stand. dev. of the error [deg.]	3.18	1.77	1.72
Mean CPU time [s]	0.38	3.92	57.16
Mean number of iterations	11.10	7.83	7.66

TABLE I
[*3 DOFs Stäubli*] Statistics of $\|\mathbf{r} - \widehat{\mathbf{r}}\|$, mean CPU time and mean number of iterations of the gyroscope over the 94 images.

C. *Fixed-wing UAV*

In a final battery of tests, we mounted the Ricoh Theta on a Parrot Disco FPV drone and recorded an MP4 video, included in the SVMIS dataset, at 30 fps (see Fig. 9(b)). The Disco is a single-propeller, battery-powered UAV with a wingspan of 1.15 m, a weight of 0.75 kg and a flight autonomy of 45 minutes. We chose a front elevated position for the camera on the drone, in order to have a view of the surrounding environment as unobstructed as possible, and to not overly perturbe its center of mass. The Disco was remotely controlled over an open field near Amiens, France. It reached a maximum altitude of 106 m and a maximum speed over ground of 85.9 km/h. The total flight time was 6’41” and the total distance traveled 4.2 km (see Fig. 9(a)). The gyroscope was initialized and parametrized as in Sect. V-B.2. In the absence of a reliable ground truth (the output of the Ricoh Theta’s embedded IMU was too inaccurate for a quantitative study), we decided to perform a qualitative evaluation of our method. To this end, we chose the reference image reported in Fig. 9(c) (time 4’41” of the video sequence): this equirectangular image was generated from the corresponding dual fisheye and it is displayed at full resolution for ease of visualization. Fig. 9(d) reports the current image (time 5’11”) without correction, and Fig. 9(e) after correction using the attitude of the drone estimated by the gyroscope for $N = 3$. The reference and current images were taken at an altitude of 50.1 m and 58.3 m, respectively, and they are 108 m apart (as the crow flies). In spite of this, the gyroscope yielded a satisfactory compensation: in fact, Fig. 9(e) and Fig. 9(c) have a very similar appearance in terms of skyline and light-source direction (sun’s position). A comparable compensation level was observed in 88% of the frames of the 6’41” sequence, as it is evident in the video available in the attachment accompanying this paper and at the address below[3]. The poor results (12% of the frames) are attributable to local minima of the MPP cost function.

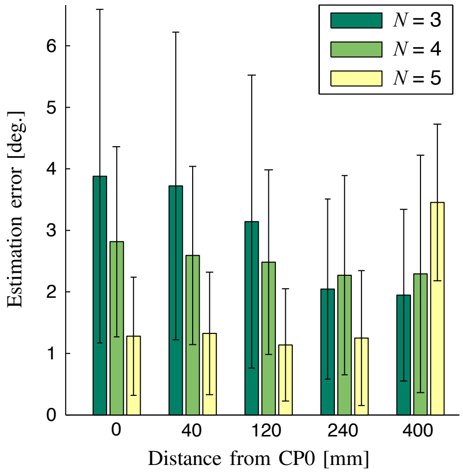
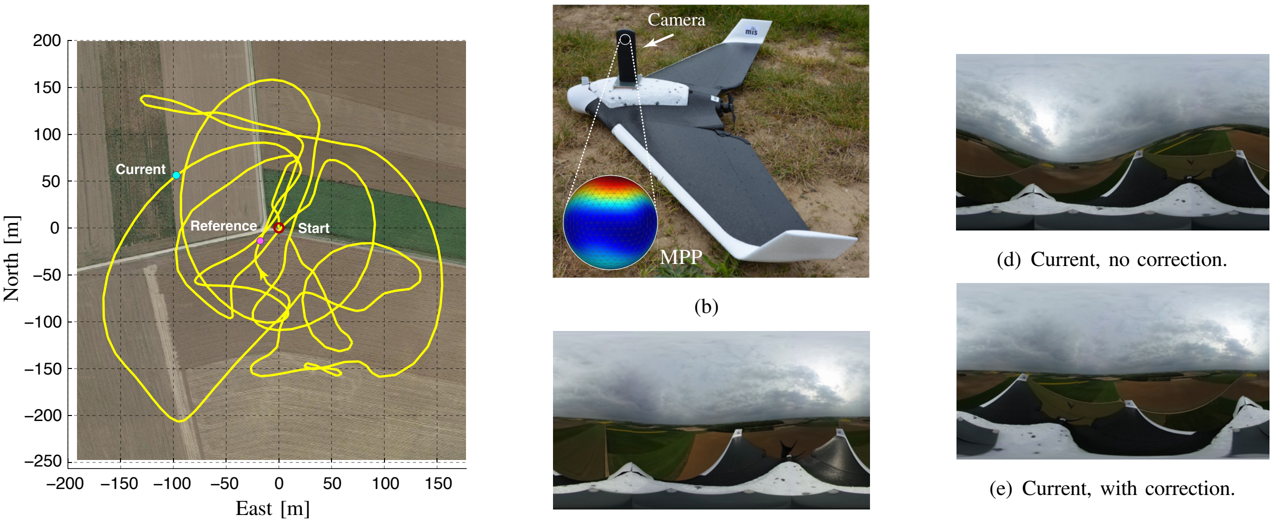


Fig. 8. [*1 DOF Stäubli*] Mean and standard deviation of the magnitude of the estimation error for a growing distance from CP0 and for $N \in \{3, 4, 5\}$.

VI. Conclusions and Future Work

In this paper, we have introduced a direct omnidirectional visual gyroscope which leverages a new and compact representation of spherical images based on

[3] `http://mis.u-picardie.fr/~g-caron/pub/data/SVMIS_video.zip`



(a)

(b)

(c) Reference.

(d) Current, no correction.

(e) Current, with correction.

Fig. 9. (a) GPS trajectory of the Disco drone (local geodetic frame); (b) The Ricoh Theta onboard the drone; (c) Reference equirectangular image (cf. the magenta disk in (a)); (d) Current equirectangular image without correction (cf. the cyan disk in (a)); (e) Current equirectangular image after correction using the attitude of the drone estimated by the visual gyroscope.

the *Mixture of Photometric Potentials* (MPP). Experimental evidence obtained with a twin-fisheye camera mounted on two robotic platforms, indicates that the proposed gyroscope is accurate, easy to tune and robust to illumination changes.

While a computation time of less than 1 second per image for $N = 3$ is promising, our gyroscope is not, at present, in the realm of real-time applications: however, we are confident that a machine-specific optimized implementation will overcome this limitation. We are also looking at adaptative tuning schemes of the parameters λ_g and N (possibly in conjunction with inertial measurements), and at suitable monotonic functions which might reshape the MPP cost function to provide global convergence.

References

[1] D. Scaramuzza and F. Fraundorfer. Visual Odometry - Part I : The First 30 Years and Fundamentals. *IEEE Rob. Autom. Mag.*, 18(4):80–92, 2011.

[2] K. Yamazawa, Y. Yagi, and M. Yachida. Omnidirectional Imaging with Hyperboloidal Projection. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, volume 2, pages 1029–1034, 1993.

[3] K. Miyamoto. Fish Eye Lens. *J. Opt. Soc. Am.*, 54(8):1060–1061, 1964.

[4] S. Li. Full-View Spherical Image Camera. In *Proc. IEEE Int. Conf. Pattern Recogn.*, volume 4, pages 386–390, 2006.

[5] T. Luhmann. A Historical Review on Panorama Photogrammetry. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, 34(5/W16), 2004.

[6] J.D. Adarve and R. Mahony. Spherepix: a Data Structure for Spherical Image Processing. *IEEE Robot. Autonom. Lett.*, 2(2):483–490, 2017.

[7] H.-E. Benseddik, H. Hadj-Abdelkader, B. Cherki, and S. Bouchafa. Direct method for rotation estimation from spherical images using 3D mesh surfaces with SPHARM representation. *J. Vis. Commun. Image R.*, 40:708–720, 2016.

[8] J. Kopf. 360° Video Stabilization. *ACM Trans. Graphic*, 35(6), 2016.

[9] J. Jung, B. Kim, J. Lee, B. Kim, and S. Lee. Robust upright adjustment of 360 spherical panoramas. *Vis. Comput.*, 33(6):737–747, 2017.

[10] G. Caron, E. Mouaddib, and E. Marchand. 3D model based tracking for omnidirectional vision: A new spherical approach. *Robot. Autonom. Syst.*, 60(8):1056–1068, 2012.

[11] G.L. Mariottini, S. Scheggi, F. Morbidi, and D. Prattichizzo. An accurate and robust visual-compass algorithm for robot-mounted omnidirectional cameras. *Robot. Autonom. Syst.*, 60(9):1179–1190, 2012.

[12] J.-C. Bazin, C. Demonceaux, P. Vasseur, and I. Kweon. Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment. *Int. J. Robot. Res.*, 31(1):63–81, 2012.

[13] D. Churchill and A. Vardy. An Orientation Invariant Visual Homing Algorithm. *J. Intell. Robot. Syst.*, 71(1):3–29, 2013.

[14] D. Scaramuzza and R. Siegwart. Appearance-Guided Monocular Omnidirectional Visual Odometry for Outdoor Ground Vehicles. *IEEE Trans. Robot.*, 24(5):1015–1026, 2008.

[15] W. Stürzl and H. Mallot. Efficient visual homing based on Fourier transformed panoramic images. *Robot. Autonom. Syst.*, 54(4):300–313, 2006.

[16] A. Kyatkin and G. Chirikjian. Pattern Matching as a Correlation on the Discrete Motion Group. *Comput. Vis. Image Und.*, 74(1):22–35, 1999.

[17] A. Makadia, C. Geyer, and K. Daniilidis. Correspondence-free structure from motion. *Int. J. Comput. Vision*, 75(3):311–327, 2007.

[18] F. Morbidi and G. Caron. Phase Correlation for Dense Visual Compass from Omnidirectional Camera-Robot Images. *IEEE Robot. Autonom. Lett.*, 2(2):688–695, 2017.

[19] L. Heng and B. Choi. Semi-direct visual odometry for a fisheye-stereo camera. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 4077–4084, 2016.

[20] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems. *IEEE Trans. Robot.*, 33(2):249–265, 2017.

[21] N. Crombez, G. Caron, and E. Mouaddib. Photometric Gaussian Mixtures based Visual Servoing. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 5486–5491, 2015.

[22] X. Ying and Z. Hu. Can We Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model? In *Proc. Eur. Conf. Comp. Vis.*, pages 442–455, 2004.

[23] C. Geyer and K. Daniilidis. A Unifying Theory for Central Panoramic Systems and Practical Implications. In *Proc. Eur. Conf. Comp. Vis.*, pages 445–461, 2000.

[24] W. Gao and S. Shen. Dual-Fisheye Omnidirectional Stereo. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 6715–6722, 2017.

[25] S. Li and Y. Hai. A Full-View Spherical Image Format. In *Proc. IEEE Int. Conf. Pattern Recogn.*, pages 2337–2340, 2010.

[26] G. Caron, E. Marchand, and E. Mouaddib. Photometric visual servoing for omnidirectional cameras. *Auton. Robot.*, 35(2):177–193, 2013.

[27] G. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, 2000.

[28] D.M. Titterington, A.F. Smith, and U.E. Makov. *Statistical Analysis of Finite Mixture Distributions*. John Wiley & Sons, 1985.

[29] G. Caron and D. Eynard. Multiple Camera Types Simultaneous Stereo Calibration. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 2933–2938, 2011.

[30] P.J. Huber. *Robust Statistics*. John Wiley & Sons, 1981.