

Photometric visual servoing for omnidirectional cameras

Guillaume Caron, Eric Marchand, El Mustapha Mouaddib

► **To cite this version:**

Guillaume Caron, Eric Marchand, El Mustapha Mouaddib. Photometric visual servoing for omnidirectional cameras. *Autonomous Robots*, Springer Verlag, 2013, 35 (2-3), pp.177-193. 10.1007/s10514-013-9342-3 . hal-00829822

HAL Id: hal-00829822

<https://hal.inria.fr/hal-00829822>

Submitted on 3 Jun 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Photometric visual servoing for omnidirectional cameras

Guillaume Caron* Eric Marchand† El Mustapha Mouaddib‡

Abstract

2D visual servoing consists in using data provided by a vision sensor for controlling the motions of a dynamic system. Most of visual servoing approaches has relied on the geometric features that have to be tracked and matched in the image acquired by the camera. Recent works have highlighted the interest of taking into account the photometric information of the entire image. This approach was tackled with images of perspective cameras. We propose, in this paper, to extend this technique to central cameras. This generalization allows to apply this kind of method to catadioptric cameras and wide field of view cameras. Several experiments have been successfully done with a fisheye camera in order to control a 6 degrees of freedom (dof) robot and with a catadioptric camera for a mobile robot navigation task.

Autonomous Robots, 2013

1 Introduction

Visual servoing uses the information provided by a vision sensor to control the movements of a dynamic system ([22, 9, 11]). Considering 2D visual servoing, geometric features are usually computed from image measurements and then, tracked and matched in the images acquired by the camera. The visual servoing control law allows to move the camera to a reference position minimizing the error between desired visual features and their correspondences detected in the current image.

Geometric features (points, lines, circles, moments) have been widely used for visual servoing. These features and their different representations can bring interesting properties for the servoing process, such as, when well chosen, a nice decoupling between the camera degrees of freedom ([8]). However, detecting, tracking and matching image measurements and the corresponding visual features effectively is still a difficult problem ([26]). To deal with this issue, it is possible to directly use the image as a whole rather than extract geometrical features. This leads to direct visual servoing scheme ([30, 16, 12, 15, 23]). It withdraws

*Université de Picardie Jules Verne, MIS laboratory, Amiens, France Guillaume.Caron@u-picardie.fr

†Université de Rennes 1, IRISA, INRIA Rennes - Bretagne Atlantique, Lagadic marchand@irisa.fr

‡Université de Picardie Jules Verne, MIS laboratory, Amiens, France mouaddib@u-picardie.fr

the detection and matching problems and, moreover, allows to consider more information leading to higher accuracy.

Such idea was initially proposed in [30] and in [16]. These works, although no features were extracted from the images, do not directly use the image intensity since an eigenspace decomposition is performed to reduce the dimensionality of image data. The control is then performed in the eigenspace and the related interaction matrix that links the variation of the eigenspace to the camera motion is learnt offline. This learning process has two drawbacks: it has to be done for each new object/scene and requires the acquisition of many images of the scene at various camera positions.

Photometric features, *i.e.* pixel intensities of the entire image, have been proposed to directly be used as input of the control scheme by [12]. In that case, the goal is to minimize the error between the current and desired image. The visual feature is nothing but the image itself. The analytic formulation of the interaction matrix that links the variation of the image intensity to the camera motion was exhibited in [12]. [6] used also a direct intensity based visual servoing approach. A direct tracking process based on the image intensity is considered to estimate the homography between current and desired image. A control law that uses the parameters of this homography as visual feature is then proposed. An interesting kernel-based approach, which also considers the pixels intensity, has been recently proposed in [23]. However, only the translations and the rotation around the optical axis have been considered.

This paper addresses the problem of photometric visual servoing for central omnidirectional cameras using image intensity as visual feature. It extends the work of [12] investigating different image representations and models directly induced by omnidirectional vision. Indeed, omnidirectional images can be elevated on the sphere of the unified projection model for central cameras ([3]).

Previous works were done on the omnidirectional visual servoing to control robotics arms ([33]) as well as mobile robots ([28, 34, 5]).

We propose to formulate the omnidirectional photometric visual servoing on the sphere of this model, which is better adapted to the omnidirectional image geometry and low level processing ([17]). We also considered the image plane representation for a fair comparison process.

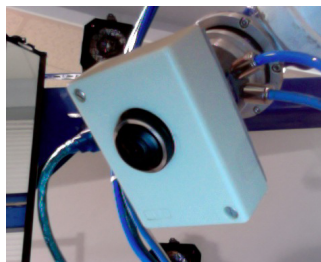
Many authors point out the interest of considering a large field of view (catadioptric camera, non-overlapping cameras...) for mobile robot navigation as ([1, 24, 29]), to cite a few papers about the topic:

- First, omnidirectional view provide a complete view of the camera environment which increases the chances of sensing non uniform zones of the environment, photometric visual servoing, as introduced in [12], being particularly efficient in these conditions.
- Second, translations have less impact on the acquired images and convergence. Rotation of the camera leads to a simple rotation in the image whereas with a perspective camera this may lead to a complete lack of shared information. We thus usually have a smaller disparity (distance between the projection of a physical point in the current

and desired image) in omnidirectional images, leading to a smaller error between the two images, which allows an easier convergence of the control law.

Omnidirectional photometric visual servoing methods developed in this paper are applied on a 6 d.o.f. Gantry robot (which is equipped with a camera with a fisheye lens, Fig. 1) and on a 2 d.o.f. mobile robot (with a catadioptric camera, Fig. 9(a)). In the later case, we also applied the visual servoing scheme proposed in the paper to an indoor navigation task.

This paper is organized as follows. First, the single viewpoint camera model is recalled and then omnidirectional visual servoing is recalled for point features. Then, the photometric visual servoing for an omnidirectional camera is developed. Visual servoing results obtained on a 6 d.o.f. Gantry robot are presented and behaviors between the various proposed control laws are compared. These studies are also made from results obtained from a mobile robot. The visual path following task is finally presented as well as results.



(a) The used fisheye camera for the visual servoing.



(b) An omnidirectional image from camera (a).

Figure 1: Vision sensor used for experiments on a Gantry robot.

2 Problem Statement

In this section, we will present some geometrical tools which will be useful for the next sections.

2.1 Coordinates representation

Central cameras can be modeled by the unified projection model ([3]). This model describes a family of cameras which includes perspective cameras and catadioptric ones with various shape mirrors ([2]), and fisheye cameras ([35, 13]).

According to this model (Fig. 2), a 3D point $\mathbf{X} = (X \ Y \ Z)^\top$ is first projected on a sphere of unit radius (the equivalent sphere) as \mathbf{X}_S :

$$\mathbf{X}_S = pr_S(\mathbf{X}) \quad \text{with} \quad \begin{cases} X_S = \frac{X}{\rho} \\ Y_S = \frac{Y}{\rho} \\ Z_S = \frac{Z}{\rho} \end{cases} \quad (1)$$

where $\mathbf{X}_S = (X_S \ Y_S \ Z_S)^\top$ and $\rho = \sqrt{X^2 + Y^2 + Z^2}$.

This parameterization is redundant on a sphere since a coordinate can be written as a combination of others. Hence, two angular coordinates (azimuth and elevation) are sufficient to represent a point on a sphere $\mathbf{S} = (\phi \ \theta)^\top$ with:

$$\mathbf{S} = \begin{cases} \phi = \arccos(Z_S) = \arccos(Z / \rho) \\ \theta = \arctan(Y_S / X_S) = \arctan(Y / X) \end{cases} \quad (2)$$

The spherical projection is followed by a perspective projection on the image plane (Fig. 2) as $\mathbf{x} = (x \ y \ 1)^\top$ by using parameter ξ which depends intrinsically on the omnidirectional camera type. The final relation between the 3D point and its normalized image point is:

$$\text{with : } \begin{cases} x = \frac{X_S}{Z_S + \xi} = \frac{X}{Z + \xi\rho} = \frac{\cos(\theta) \sin(\phi)}{\cos(\phi) + \xi} \\ y = \frac{Y_S}{Z_S + \xi} = \frac{Y}{Z + \xi\rho} = \frac{\sin(\theta) \sin(\phi)}{\cos(\phi) + \xi} \end{cases} \quad (3)$$

The image point is obtained by:

$$\mathbf{u} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} p_x & 0 & u_0 \\ 0 & p_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathbf{K}\mathbf{x}, \quad (4)$$

where (p_x, p_y) are the product of the focal length and horizontal (resp. vertical) size of a pixel and (u_0, v_0) are the principal point coordinates. They are part of the projection model parameters $\gamma = \{p_x, p_y, u_0, v_0, \xi\}$ that are supposed to be known after a calibration process.

The projection function is invertible as pr_ξ^{-1} and allows to retrieve the point on the sphere corresponding to \mathbf{x} :

$$\mathbf{X}_S = pr_\xi^{-1}(\mathbf{x}) = \begin{pmatrix} \frac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1} x \\ \frac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1} y \\ \frac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1} - \xi \end{pmatrix}. \quad (5)$$

This equation is used to obtain points coordinates on the unit sphere of the model in Cartesian representation which can be transformed to the angular representation using equation (2).

As we have seen, the unified model is based on an equivalence sphere as intermediate representation place. This place, will be used to do the visual servoing. Then, we have new possibilities to model the image and we will be interested in:

1. Cartesian representation \mathbf{x} or \mathbf{u} (two coordinates) on the 2D image plane
2. Cartesian representation \mathbf{X}_S (three coordinates) on the 3D equivalence sphere
3. Spherical representation \mathbf{S} (two coordinates) on the 3D equivalence sphere

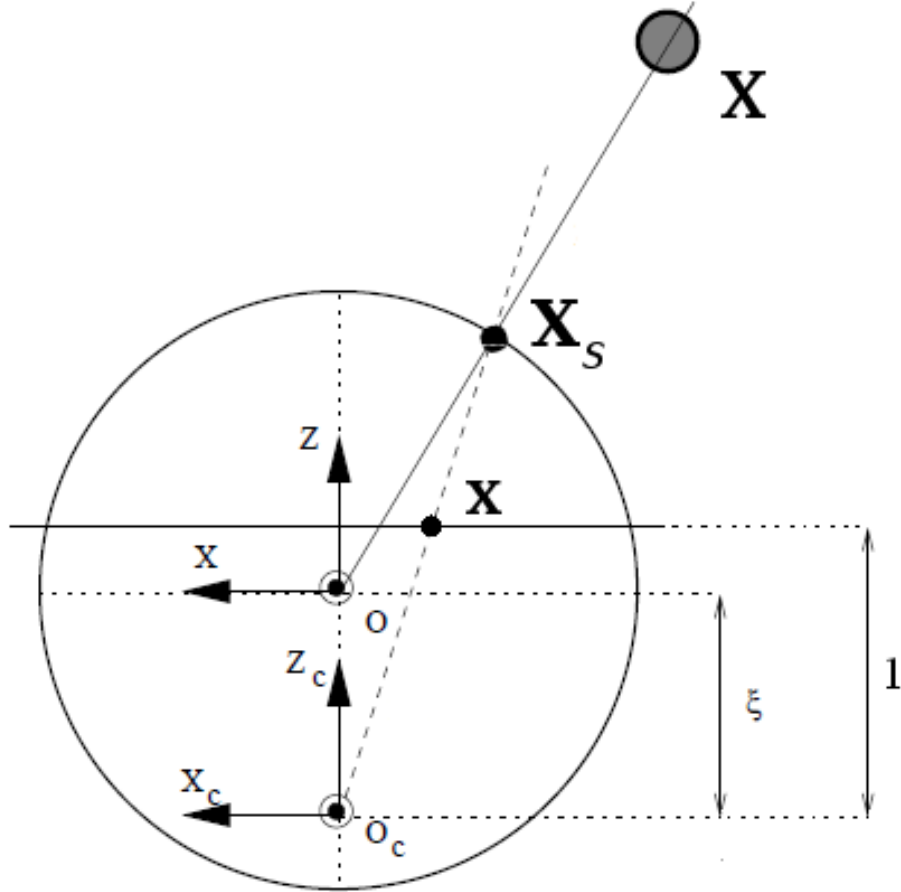


Figure 2: Unified projection model for central cameras ([3]).

2.2 Features for Omnidirectional Visual Servoing

Many features (points, lines, moments, spheres, ...) have been studied for visual servoing. As our work focuses on the photometric visual servoing, we will present only the point feature because it represents an introduction and because it is very close to the intensity feature in the visual servoing scheme.

2.2.1 Geometric Feature: the point

To control the degrees of freedom of a robot using visual servoing, image points can be used as features. Then the design of the control law aims to minimize the error \mathbf{e} between the current visual features $\mathbf{s}(\mathbf{r})$ observed from the pose $\mathbf{r} = [t_X, t_Y, t_Z, \theta_X, \theta_Y, \theta_Z]$ and their desired value in the image \mathbf{s}^* . The interaction matrix \mathbf{L}_s , linking the time variation of visual features $\dot{\mathbf{s}}$ to the camera velocity \mathbf{v} , has also to be defined. This link is formally expressed

as:

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v} \quad (6)$$

where $\mathbf{v} = (\mathbf{v} \ \boldsymbol{\omega})^\top$ is composed of the linear and angular camera velocities, $\mathbf{v} = [v_X, v_Y, v_Z]$ and

$\boldsymbol{\omega} = [\omega_X, \omega_Y, \omega_Z]$, respectively. A control law is designed to try to have an exponential decoupled decrease of the error $\mathbf{e} = \mathbf{s}(\mathbf{r}) - \mathbf{s}^*$:

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}}_s^+ \mathbf{e} \quad (7)$$

with λ a tunable gain to modify the convergence rate and $\widehat{\mathbf{L}}_s^+$, the pseudo-inverse of a model of \mathbf{L}_s . We mention ‘‘a model of \mathbf{L}_s ’’ because it cannot be exactly known due to never perfect knowledge of camera parameters or scene structure. Hence, if a representation (cartesian image plane, cartesian spherical or pure spherical) leads to a better behavior (convergence, residual decrease, camera trajectory), it will mean that this representation is a better modeling of the perfect interaction matrix than others.

2.2.2 Photometric Feature: the intensity

The visual features considered in photometric visual servoing are the luminance of each point of the image. This approach has already been developed in the case of perspective images. It needs to solve two problems: the interaction matrix and the image gradient.

We consider that the luminance I at a constant pixel location $\mathbf{x} = (x, y)$ for all \mathbf{x} belonging to the image domain and for a given pose \mathbf{r} . Thus, we have:

$$\mathbf{s}(\mathbf{r}) = \mathbf{I}(\mathbf{r}) = (\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_N)^\top \quad (8)$$

where \mathbf{I}_i is the row vector containing pixel intensities of the image line number i . i is varying from 1 to N , the image height. \mathbf{I}_i has a size equal to M , the image width. The size of the vector $\mathbf{I}(\mathbf{r})$ is equal to $N \times M$, *i.e.* the number of image pixels.

We have thus to minimize the difference between current and desired images:

$$\mathbf{e} = \mathbf{I}(\mathbf{r}) - \mathbf{I}^*(\mathbf{r}^*), \quad (9)$$

The control law is then, as in a classical visual servoing scheme, given by:

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}}_I^+ (\mathbf{I}(\mathbf{r}) - \mathbf{I}^*(\mathbf{r}^*)) \quad (10)$$

Nevertheless, considering visual servoing as an optimization problem ([25]), [12] formulate photometric servoing control law using a Levenberg-Marquardt like optimization technique. It has been shown that it ensures better convergence than other kind of control laws. So, instead of using a Gauss-Newton like control law (eq. (7)), this control law is used in the current work:

$$\mathbf{v} = -\lambda (\mathbf{H} + \mu \text{diag}(\mathbf{H}))^{-1} \mathbf{L}_I^\top (\mathbf{I}(\mathbf{r}) - \mathbf{I}^*(\mathbf{r}^*)) \quad (11)$$

with $\mathbf{H} = \mathbf{L}_I^\top \mathbf{L}_I$, considering \mathbf{L}_I is the interaction matrix related to luminance of image I . If μ is very high this control law behaves like a steepest descent whereas a very low value for μ leads eq. (11) to behave like eq. (10).

As for any pure image based visual servoing approach, only local stability can be obtained in the photometric visual servoing since redundant visual features are used ([9]). However, as pointed out in the latter reference, this domain is quite large in practice.

2.3 Gradient Computation

As mentioned in the previous section, we need to compute some gradients (geometrical, temporal and image gradient). These gradients depend on the coordinates representation and the used features. Since we will use the equivalence sphere as a work space, we will have other possibilities and schemes than traditional ones to estimate these gradients.

3 Omnidirectional Photometric Visual Servoing

This section describes three representations for omnidirectional photometric visual servoing: the image plane Cartesian representation, the spherical Cartesian representation and the spherical representation. Photometric visual servoing formulations for each representation are successively proposed and validated in simulation. They are faced in the discussion on simulation results that ends the section.

3.1 Image Plane Visual Servoing: Cartesian Representation

3.1.1 Interaction Matrix for Geometrical Servoing

When considering points as visual features, *i.e.* $\mathbf{x} = (x, y)$, the interaction matrix \mathbf{L}_x , that links the point velocity to the camera velocity \mathbf{v} is given by ([9]):

$$\mathbf{L}_x = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \frac{\partial \mathbf{X}}{\partial \mathbf{r}} \quad (12)$$

For the image plane, the partial derivatives $\frac{\partial \mathbf{x}}{\partial \mathbf{X}}$ are computed from eq. (3) and lead to the interaction matrix for a point in the omnidirectional image plane.

The second jacobian, $\frac{\partial \mathbf{X}}{\partial \mathbf{r}}$, is well known for points ([19]) and does not depend on the projection model which is encapsulated in the first Jacobian. Then, \mathbf{L}_x is ([4]):

$$\mathbf{L}_x = (\mathbf{L}_1 \quad | \quad \mathbf{L}_2) \quad (13)$$

with

$$\mathbf{L}_1 = \begin{pmatrix} -\frac{1+x^2(1-\xi(\alpha+\xi))+y^2}{\rho(\alpha+\xi)} & \frac{\xi xy}{\rho} & \frac{\alpha x}{\rho} \\ \frac{\xi xy}{\rho} & -\frac{1+y^2(1-\xi(\alpha+\xi))+x^2}{\rho(\alpha+\xi)} & \frac{\alpha y}{\rho} \end{pmatrix} \quad (14)$$

and

$$\mathbf{L}_2 = \begin{pmatrix} xy & -\frac{(1+x^2)\alpha-\xi y^2}{\alpha+\xi} & y \\ \frac{(1+y^2)\alpha-\xi x^2}{\alpha+\xi} & -xy & -x \end{pmatrix} \quad (15)$$

where $\alpha = \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}$ and ρ defined in eq. (3).

3.1.2 Interaction Matrix for Photometric Servoing

The interaction matrix $\mathbf{L}_I(\mathbf{x})$ formulation is defined under temporal luminance consistency ([12]):

$$\mathbf{I}(\mathbf{x} + d\mathbf{x}, t + dt) = \mathbf{I}(\mathbf{x}, t) \quad (16)$$

assuming $d\mathbf{x}$ is small. If it is small enough, and considering a Lambertian scene, the optical flow constraint equation (OFCE) is valid ([21]):

$$\nabla \mathbf{I}^T \dot{\mathbf{x}} + \mathbf{I}_t = 0 \quad (17)$$

with ∇I the spatial gradient of $I(\mathbf{x}, t)$ and $I_t = \frac{\partial I(\mathbf{x}, t)}{\partial t}$, is the temporal gradient. For the omnidirectional image plane photometric visual servoing (IP-VS), the knowledge of \mathbf{L}_x

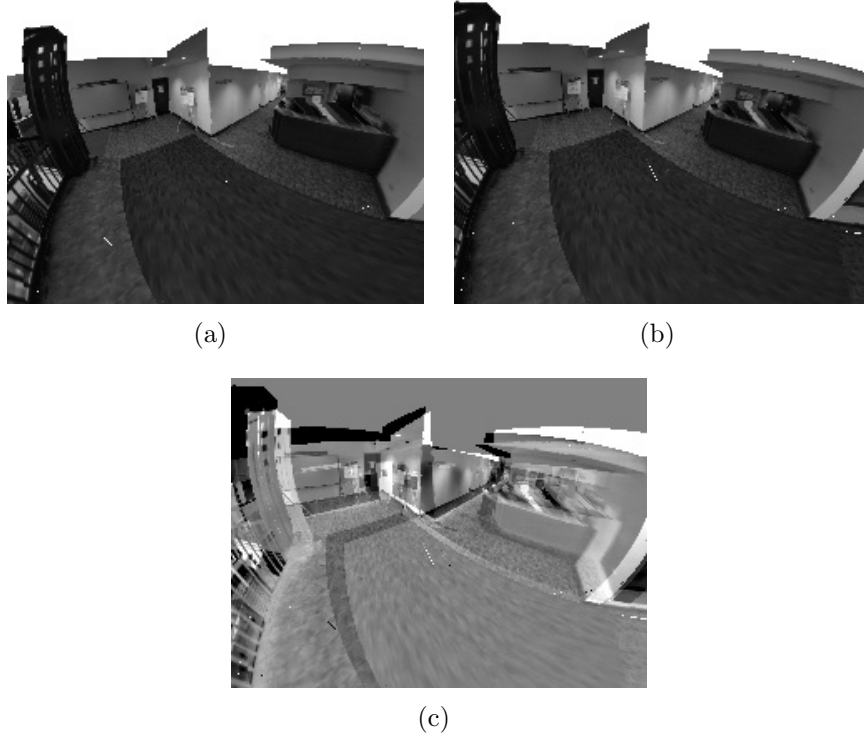


Figure 3: Omnidirectional virtual images. (a) desired image, (b) an initial image, (c) difference between (a) and (b).

(eq. (13)) leads to:

$$I_t = -\nabla I^\top \mathbf{L}_x \mathbf{v}. \quad (18)$$

And similarly to [12], we get the interaction matrix $\mathbf{L}_I(\mathbf{x})$ related to I at pixel \mathbf{x} :

$$\mathbf{L}_I(\mathbf{x}) = -\nabla I^\top \mathbf{L}_x, \quad (19)$$

where \mathbf{L}_x is given by eq. (13).

3.1.3 Photometric Gradient Computation

On the image plane and in the case of perspective image, the image gradients ∇I are computed using the same neighborhood for all image point: a square regularly sampled. For instance, in [12], it is approximated by a convolution of \mathbf{I} with two Gaussian derivatives.

In omnidirectional images, we can use the same approach to compute the gradients. But, the image geometry is different, *i.e.* resolution and orientation are not constant. For this reason, many works have been done to adapt image processing tools for these images, and for gradient computation. One of these approaches, consists in adapting the neighborhood ([17]). The reader can find more details and a discussion about these technics in ([7]). In the latter work, no particular impact of adapted image gradients computation was noticeable for the image plane representation, with convolution window size 7×7 as we use. Hence, for the IP-VS, image gradients ∇I were classically computed.

3.1.4 Validation

Validation is done on synthetic views of a real environment from one of the Trakmark (<http://trakmark.net>) datasets. We made a real-time omnidirectional OpenGL camera using GPU shaders (GLSL language). The projection model of this camera is the unified projection model ([3]). An obtained image is presented in figure 3(a).

Figure 4 shows the cost function shape of the control law, which is actually the SSD (Sum of Squared Differences) of pixel intensities, when varying two degrees of freedom. The minimum is clearly identified in a range distance of $[-15 \text{ cm}, 15 \text{ cm}]$ and a range orientation of $[-10^\circ, 10^\circ]$, even if a valley appears when a translation along the X axis is combined to a rotation around the Y axis.

Using this synthetic environment, 256 initial camera poses (position and orientation) are randomly chosen around the desired pose (fig. 3(b)) leading to initial errors as shown in figure 3(c). Image plane photometric visual servoing is then applied from each of these 256 initial conditions, which are, in a mean, distant from 7.4 cm from the desired pose and 2.5 degrees on all rotation axes (standard deviations of 2.8 cm in position and 2.0 degrees in orientation around all axes). These small errors are considered in a first time in order to verify the initial hypothesis needed to apply the OFCE (Section 3.1.2) but will be challenged later in wide motion real experiments.

All tested initial poses allowed to converge, computing the interaction matrix for each image (with the knowledge of current ρ for each pixel, obtained from

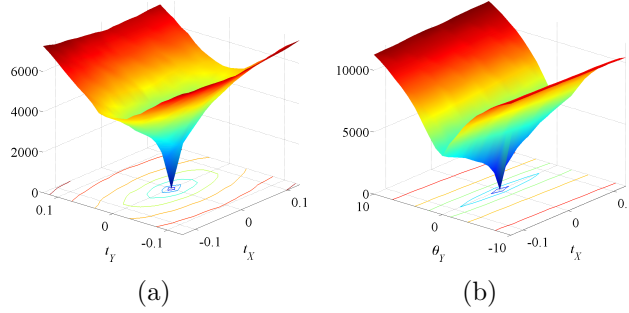


Figure 4: Cost function shape when varying two degrees of freedom: (a) combined translation along X and Y axes (in meters) and (b) combined translation along the X axis and rotation around the Y axis of the camera (in meters and degrees).

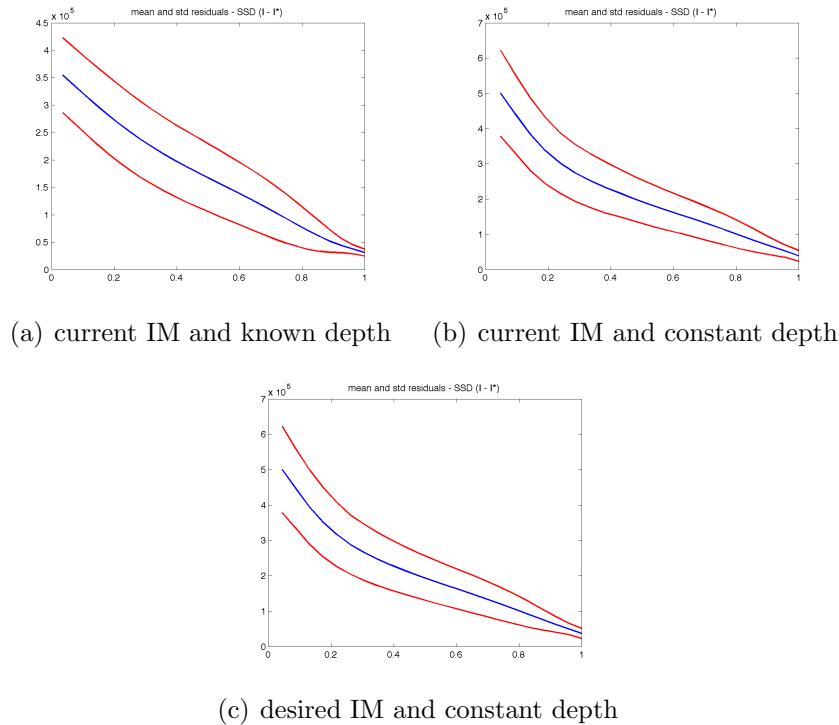


Figure 5: Residual evolutions of IP-VS with respect to a normalized time scale considering (a) current interaction matrix (IM) is computed with current depth of each pixel point, (b) current IM with constant depth and (c) desired IM and constant depth.

Z-Buffer), which has a mean condition number of 15.7 (the nearest from 1 is the best). A mean of 12 iterations are needed to converge. These two quantities will be used in comparisons with other image representations.

But since, in real conditions, the depth is unknown, we also ran the visual servoing computing the interaction matrix for each image but with a constant depth for all pixels and

all images. The ρ value corresponds to the mean of ρ values at the desired pose. 100 % of these simulations converged in a mean of 10.3 iterations for a mean conditioning of 21.2.

Finally, the computation (and inversion) of the interaction matrix for each image is a bit too long to keep a standard 25 Hz frame rate and, as it is often chosen as a correct approximation, assuming the camera is rather close to its optimal pose, the desired interaction matrix with a constant depth is finally used, as it will be the case in real experiments. With this consideration, still 100 % of tested cases converged in a mean of 10.3 iterations for an interaction matrix conditioning of 18.8. All these results are summed up in table 1.

Table 1: Synthetic IP-VS results. The knowledge of depth values and current interaction matrix (IM) is varying.

current IM	yes	yes	no
desired IM	no	no	yes
depth image	yes	no	no
constant depth	no	yes	yes
mean iterations	12.0	10.3	10.3
mean conditioning	15.7	21.2	18.8

Of course, approximations made, from the use of current interaction with known depth image to the use of the desired interaction matrix with a constant depth, tend to decrease the error evolution quality (the “most exponential” error decrease implies the best quality). This can be verified on the mean residual evolutions with respect to a normalized time scale on figure 5. However, despite a bit worth exponential decrease of residuals, the number of iterations to convergence is slightly better. One may note that the knowledge of real depth impacts only the translation part of the interaction matrix (eq. (13)), allowing to think that changing the depth knowledge do not have a direct impact on the rotational part. It only generates a faster or slower correction of the orientation, due to the translation speed influenced by the use of a constant depth or not.

3.2 Sphere Based Visual Servoing: Cartesian representation

This section will present the cartesian formulation of the visual servoing on the equivalence sphere. Thus, the feature vector is, for this representation (eq. 8):

$$\mathbf{s}(\mathbf{r}) = [\dots \mathbf{I}_S(\mathbf{X}_S) \dots]^T \quad (20)$$

3.2.1 Interaction Matrix for Photometric Servoing

Working on the equivalent sphere implies working with a coordinates system on the sphere. The spherical coordinates are the inverse projection from image plane coordinates (eq. (5)). We will call the image indexed by these coordinates: spherical image I_S .

Still under constant illumination assumption (eq. (16)), the OFCE (eq. (17)) is reformulated to fit the cartesian spherical representation:

$$\nabla I_S^\top \dot{\mathbf{X}}_S + I_{S_t} = 0, \quad (21)$$

detailed as:

$$\frac{\partial I_S}{\partial X_S} \dot{X}_S + \frac{\partial I_S}{\partial Y_S} \dot{Y}_S + \frac{\partial I_S}{\partial Z_S} \dot{Z}_S + I_{S_t} = 0. \quad (22)$$

The interaction matrix $\mathbf{L}_{I_S(\mathbf{x}_S)}$ related to I_S at spherical point \mathbf{X}_S , for cartesian spherical photometric visual servoing (CS-VS), is defined by:

$$\mathbf{L}_{I_S(\mathbf{x}_S)} = -\nabla I_S^\top \mathbf{L}_{\mathbf{x}_S}. \quad (23)$$

$\mathbf{L}_{\mathbf{x}_S}$ is given by equation (24) but the image gradient on the sphere have yet to be expressed.

3.2.2 Interaction Matrix for Geometric Servoing

This parameterization uses 3-coordinates points \mathbf{X}_S onto the equivalent sphere as defined in equation (1). The interaction matrix $\mathbf{L}_{\mathbf{x}_S}$ has been defined by ([20]), in a different purpose:

$$\mathbf{L}_{\mathbf{x}_S} = \frac{\partial \mathbf{X}_S}{\partial \mathbf{X}} \frac{\partial \mathbf{X}}{\partial \mathbf{r}} = \left(\frac{1}{\rho} (\mathbf{X}_S \mathbf{X}_S^\top - \mathbf{I}_3) \quad [\mathbf{X}_S]_\times \right)_{3 \times 6}. \quad (24)$$

3.2.3 Photometric Gradient Computation

To approximate the image gradient on the sphere, we estimate $\frac{\partial I_S}{\partial X_S}$, $\frac{\partial I_S}{\partial Y_S}$ and $\frac{\partial I_S}{\partial Z_S}$ by using a finite difference. We need to chose values for Δ_{X_S} , Δ_{Y_S} and Δ_{Z_S} and to interpolate the values of the intensity at this point. To compute gradients $(\frac{\partial I_S}{\partial X_S}, \frac{\partial I_S}{\partial Y_S}, \frac{\partial I_S}{\partial Z_S})$ directly on the sphere, we propose to define the following sampling step:

$$\Delta_{X_S} = \Delta_{Y_S} = \Delta_{Z_S} = \left\| pr_\xi^{-1} \left(\begin{bmatrix} u_0 + 1 \\ v_0 \\ 1 \end{bmatrix} \right) - \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\|. \quad (25)$$

From a point $(X_S \ Y_S \ Z_S)^\top$, three N -neighborhoods are defined, one for each axis. N defines the neighborhood size around a point. The spherical neighborhood for the first coordinate, X_{S_N} , is defined by:

$$X_{S_N} = \left\{ \begin{array}{c} (X_S + k\Delta_{X_S} \ Y_S \ Z_S)^\top \\ \left\| (X_S + k\Delta_{X_S} \ Y_S \ Z_S)^\top \right\| \\ -\frac{N}{2} \leq k \leq \frac{N}{2}, k \neq 0 \end{array} \right\}. \quad (26)$$

The procedure is similar for Y_{S_N} and Z_{S_N} . The neighborhood computation leads to N Cartesian spherical points which are projected in the omnidirectional image plane to retrieve intensities which are used to compute the gradients.

3.2.4 Validation

As for the IP-VS, Cartesian spherical photometric visual servoing has been applied in the synthetic environment presented in Section 3.1.4. Results for this representation are summed up in Table 2 and still present a 100 % convergence for tested initial poses.

Comparison with respect to other representation and interpretation are done in Section 3.4.

Table 2: Synthetic CS-VS results. The knowledge of depth values and current interaction matrix (IM) is varying.

current IM	yes	yes	no
desired IM	no	no	yes
depth image	yes	no	no
constant depth	no	yes	yes
mean iterations	9.4	7.0	7.0
mean conditioning	21.4	18.0	17.7

3.3 Sphere Based Visual Servoing: Spherical representation

This section will present the spherical formulation of the visual servoing on the equivalence sphere. Thus, the feature vector is, for this representation (eq. 8):

$$\mathbf{s}(\mathbf{r}) = [\dots \mathbf{I}_S(\mathbf{S}) \dots]^\top \quad (27)$$

3.3.1 Interaction Matrix for Photometric Servoing

Considering the azimuth and elevation parameterization of a spherical point, and still under constant illumination assumption (eq. (16)), the OFCE (eq. (17)) is reformulated to fit this representation:

$$\nabla I_S^\top \dot{\mathbf{S}} + I_{S_t} = 0. \quad (28)$$

This constraint equation is used in order to develop the interaction matrix $\mathbf{L}_{I_S}(\mathbf{S})$ of the pure spherical photometric visual servoing (PS-VS), knowing \mathbf{L}_S (eq. (30)):

$$\mathbf{L}_{I_S}(\mathbf{S}) = -\nabla I_S^\top \mathbf{L}_S. \quad (29)$$

3.3.2 Interaction Matrix for Geometrical Servoing

As explained in Section 2.1, two coordinates $\mathbf{S} = (\phi \ \theta)^\top$ are enough to express a point on the unit sphere. The interaction matrix associated to feature \mathbf{S} is then:

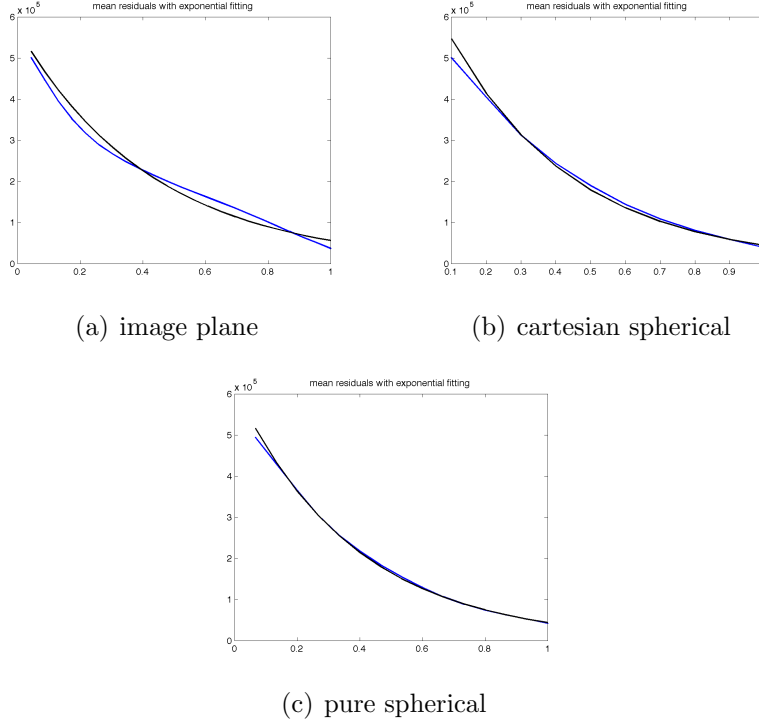


Figure 6: Exponential curve fitting on residual decrease. Blue curves are mean residuals over (normalized) time and black curves, the exponential fitting.

$$\mathbf{L}_S = \frac{\partial \mathbf{S}}{\partial \mathbf{X}} \frac{\partial \mathbf{X}}{\partial \mathbf{r}} = \begin{pmatrix} -\frac{c\theta c\phi}{\rho} & -\frac{s\theta c\phi}{\rho} & \frac{s\phi}{\rho} & s\theta & -c\theta & 0 \\ \frac{s\theta}{\rho s\phi} & -\frac{c\theta}{\rho s\phi} & 0 & \frac{c\theta c\phi}{s\phi} & \frac{s\theta c\phi}{s\phi} & -1 \end{pmatrix} \quad (30)$$

with $c\theta = \cos \theta$, $c\phi = \cos \phi$, $s\theta = \sin \theta$, $s\phi = \sin \phi$. This pure spherical formulation of visual servoing is minimal but has a singularity when $\sin \phi = 0$, *i.e.* when $x = y = 0$ in the normalized image plane (principal point). Furthermore, this representation leads to an interaction matrix more non linear than in the Cartesian spherical case, due to trigonometric functions.

3.3.3 Gradient Computation

To approximate the image gradient ∇I_S on the sphere, we estimate $\frac{\partial I_S}{\partial \theta}$ and $\frac{\partial I_S}{\partial \phi}$ by using a finite difference. We need to chose values for $\Delta\theta$, $\Delta\phi$ and to interpolate the value of the intensity at this point. The reader can see [17] and our previous paper [7] to see the details of the direct pure spherical image gradient computation.

3.3.4 Validation

As for the two other representations, PS-VS has been applied in the synthetic environment presented in Section 3.1.4. Results for this representation are summed up in table 4 and still present a 100 % convergence for tested initial poses.

Table 3: Synthetic PS-VS results with a varying knowledge about depth values and current interaction matrix (IM) or not.

current IM	yes	yes	no
desired IM	no	no	yes
depth image	yes	no	no
constant depth	no	yes	yes
mean iterations	10.0	7.2	7.3
mean conditioning	21.3	18.0	19.1

3.4 Simulation results discussion

Simulation results presented in Sections 3.1.4, 3.2.4 and 3.3.4 can be crossed to understand if a representation is better than another between image plane, cartesian spherical and pure spherical for photometric visual servoing. Even if some differences appear in terms of iterations number to converge or interaction matrix conditioning, none of the representations clearly outperforms the others. We could however note that spherical representations lead to a slightly better behavior with respect to the image plane representation: iterations number is reduced of around 23 %. Another criterion to analyze is the “exponentiality” of the residual decrease. Fitting an exponential curve on the mean residuals evolution allow to qualitatively understand a slightly better decrease of spherical representations over the image plane one (Fig. 6). The fitting is clearly better on mean residuals evolution obtained with spherical representations. This means less approximations are made with the spherical representations to ensure the exponential decrease of the error imposed to derive the control law.

One can note that the previous comparison of the three proposed representations were led using good knowledge of camera parameters. To be more complete, a final set of simulations is presented in table 4 after application of noise on intrinsic camera parameters. IP-VS, CS-VS and PS-VS are executed using the desired interaction matrix computed with the same depth for every pixels. A Gaussian noise is added to the camera parameters used for virtual image generation in terms of percentages of their value: 1% to 5%. As the table 4 denotes, there is no significant impact of the camera parameters bad knowledge for spherical representations on the chosen behavior indicators (mean iteration number to converge, interaction matrix conditioning and mean position and orientation error at convergence).

The only impact we can note is concerning IP-VS with a progressive increase of the interaction matrix condition number as well as more important position and orientation errors at convergence. Even if these errors are still small, they represent a relative increase of around 20% for the position error and 400% for the orientation error, which is not the case in both spherical representations for which the position and orientation error at convergence is rather constant when increasing camera noise parameters, up to 5%. Hence, these last simulation results highlight that the spherical representations for the omnidirectional photometric visual servoing are more robust to noised camera parameters than the image plane representation.

Table 4: Synthetic IP-VS, CS-VS and PS-VS results with a varying knowledge about camera parameters: perfect to a Gaussian noise of 5% of exact camera parameter values. The interaction matrix computed at desired pose is used with unknown depth (set constant for each pixel).

noise %		0%	5%
IP-VS	μ_i	10.3	10
	μ_c	18.8	22.6
	μ_t	0.05 mm	0.19 mm
	μ_o	0.14°	0.56°
CS-VS	μ_i	7	7
	μ_c	17.7	17
	μ_t	0.07 mm	0.05 mm
	μ_o	0.09°	0.11°
PS-VS	μ_i	7.3	7.2
	μ_c	19.1	18.7
	μ_t	0.05 mm	0.04 mm
	μ_o	0.12°	0.11°

4 Experimental results: 6 dof positioning task

In this first set of experiments, we shall consider a 6 dof positioning task. Experiments are carried out on a Gantry robot (Fig. 8(b)). A fisheye camera (Fig. 1) is mounted on the robot end-effector. Implementation was done using the ViSP library ([27]). In this section, three kind of experiments are presented. When converging, positioning error is less than one tenth of a millimeter in translation and 0.02° in rotation.

The first experiment shows photometric visual servoing using IP-VS, CS-VS and PS-VS with a unique desired position and three different initial positions and orientations. The second experiment presented in this section is a positioning task using CS-VS starting from an initial position that is far from the desired one.

For all experiments, the interaction matrix is computed only at the desired position. Furthermore, the depth Z is an unknown parameter and is supposed constant for all pixels, along the motion of the camera. The achievement of experiments shows the method is robust to a coarse estimation of Z .

Experiments were led using a Sony Vaio laptop with an Intel Centrino 1.2 GHz microprocessor, 2 GB RAM and running Windows XP, for image acquisition and computation of control vectors. Without any particular code optimization, the loop can be closed at 20 Hz for images of 640x480 pixels, which is close to standard camera acquisition framerate. Better implementation and the use of the GPU could easily make the processing faster. Since there are no geometrical features to extract, track and match, the image processing step is quite straightforward (gradient computation) and is timeless.

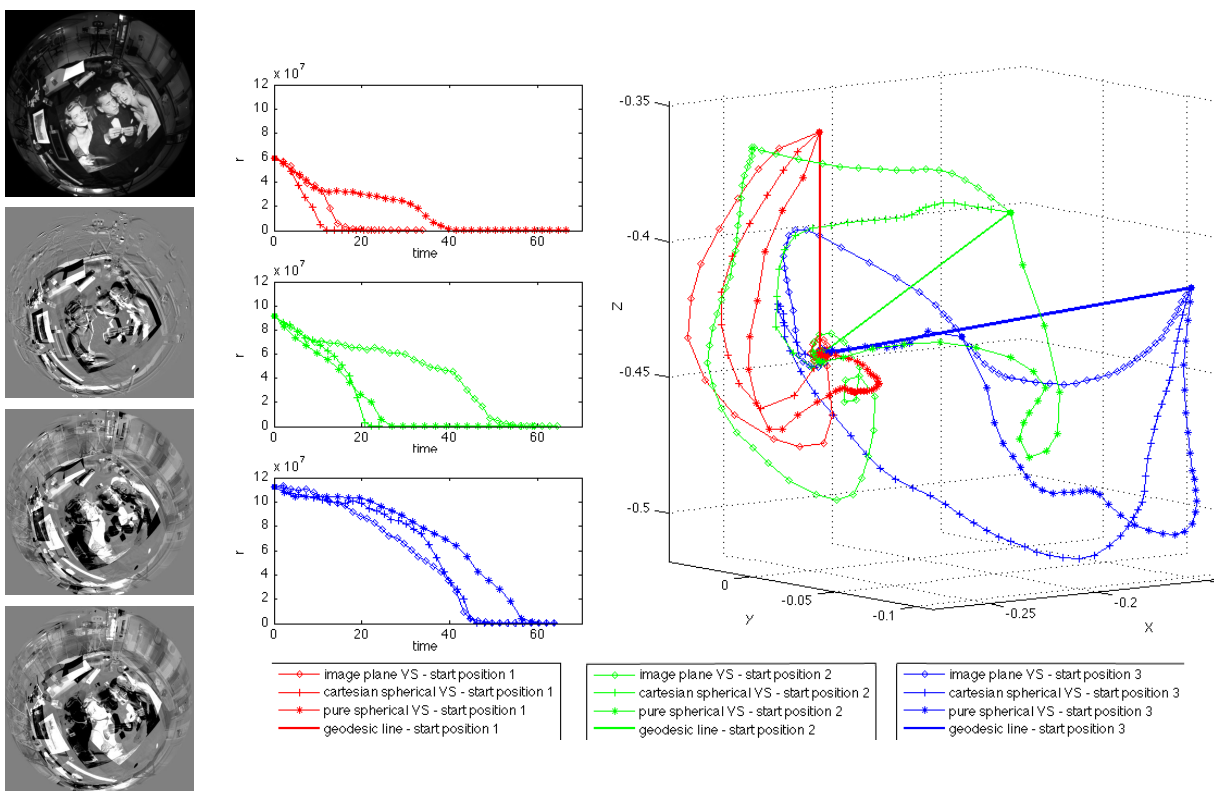


Figure 7: Different starting positions to reach the same desired one. Fisheye difference images shows several initial conditions. On the right side, camera trajectories to reach the desired position are obtained, for each starting position, using IP-VS, CS-VS and PS-VS with adapted gradients computation (spherical for the two latter). On the left side, residual evolutions over time are always in advantage of CS-VS.

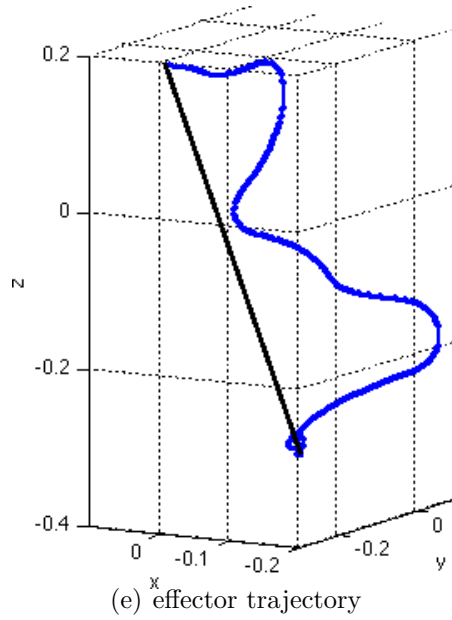
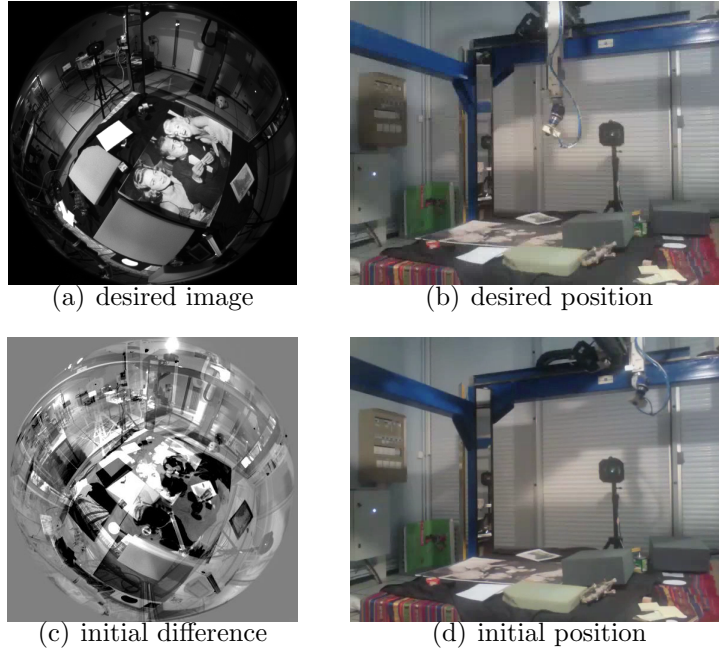


Figure 8: Experimentation starting from a position distant of $\{-41.10cm, 39.40cm, -40.00cm, 2.44^\circ, -9.31^\circ, -10.05^\circ\}$ from the desired position, inducing important differences between initial and desired images (see (a)(c)). (e) shows in blue the effector trajectory (black: geodesic).

4.1 Behavior evaluation for different initial poses

IP-VS, CS-VS and PS-VS are compared using standard image gradients computation for IP-VS and Cartesian spherical, resp. pure spherical gradients for CS-VS, resp. PS-VS. These gradients computations are theoretically and practically more valid than other computations ([7]). Results are presented in Figure 7.

Omnidirectional photometric visual servoing is done from three different initial positions to a unique desired position. Errors $\{\Delta_X, \Delta_Y, \Delta_Z, \Delta_{R_X}, \Delta_{R_Y}, \Delta_{R_Z}\}$ between desired and initial positions are respectively $\{0cm, 0cm, -8.2cm, 0^\circ, 0^\circ, 0^\circ\}$, $\{-8.7cm, -0.1cm, -4.5cm, -2.4^\circ, 11.4^\circ, 8.0^\circ\}$ and $\{-15.9cm, -1.6cm, -1.3cm, 2.5^\circ, 14.6^\circ, 7.4^\circ\}$. The latter experiment is the hardest situation, with a translation along the X axis and a rotation around the Y axis, leading to projective ambiguities. Hence, the shape of the cost function leads to less straight motions. Figure 7 shows initial images of differences and the third initial position produces the most important difference. Indeed, trajectories of IP-VS, CS-VS and PS-CS are the most different for the latter experiment. Residuals evolution over time (Fig. 7, left), always show that CS-VS converges faster.

4.2 Large motion

This experiment aims to show this technique can be used for relatively large motions. CS-VS was the only formulation allowing to converge. A possible interpretation of this result is that the translation in camera frame is mostly along optical axis and this representation has a Z component in equations, particularly in the spherical image gradient. The initial camera position is such that the initial positioning error is $\{-41.1cm, 39.4cm, -40.0cm, 2.4^\circ, -9.3^\circ, -10.1^\circ\}$. Despite important position and coupled orientations differences, which is highlighted by figure 8(c), photometric visual servoing succeeds. Figure 8(e) shows resulting camera trajectory in robot frame.

4.3 Six dof experimental results discussion

The precision at convergence is very similar to the one reached in the perspective case for which a comparison with a basic visual feature control law has been done. As reported in ([12]), SURF keypoints along with a classical or statistically robust control law have been considered. As expected, in that case, if detection and matching succeeds with no lost features, the convergence area is larger than in the photometric case. It remains that keypoint matching is still a difficult issue (especially in the case of catadioptric sensor) and that an incorrect matching could jeopardize the achievement of the task.

Furthermore, higher precision was obtained with the photometric feature rather than with the points, mainly for two reasons. Firstly, even when successful, geometric feature matching always features some uncertainty leading to imprecision in the repositioning task. Secondly, the photometric servoing control law is computed with dense information rather than sparse, leading to a higher precision thanks to feature redundancy. Let us note that, in

the perspective case, the final positioning precision is, in translation, ten times more precise with the photometric feature than with SURF points.

Since similar repositioning precisions, using a photometric scheme, have been obtained with perspective and catadioptric cameras (on the same robot), similar conclusions in both cases can then be extrapolated but have not been verified due to the lack of an efficient keypoint matching algorithm on omnidirectional images.

5 Application to mobile robot navigation

This section aims to define and evaluate a path following task with a mobile robot using a set of images as waypoints. Photometric visual servoing is applied on each image, one by one, in order to follow the path. We will first tackle the positioning of a mobile robot using one desired image and modeling the robot motion in a simple scheme.

5.1 Formulation

We want to control a non-holonomic unicycle mobile robot with two degrees of freedom (Fig. 9(a)) using the proposed photometric approach. We consider the omnidirectional camera axis is perfectly aligned with the rotation axis of the robot, which is commonly assumed, if it is well placed on the robot. The camera is furthermore placed so that its Y axis is on the main translation axis of the mobile robot. With these considerations, only velocities v_Y and ω_Z , from the velocity vector defined in section 2.2.1, have to be used and computed (Figure 9(b)). Indeed, even if the mobile robot has four wheels, wheels on the same side are actuated together and the unicycle model is valid.

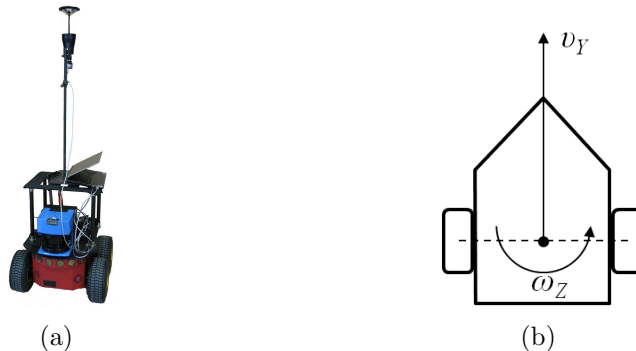


Figure 9: (a) The mobile robot Pioneer 3-AT with a catadioptric camera placed vertically. (b) top view sketch of the robot, showing the two controlled degrees of freedom.

Section 4 has experimentally shown that using the spherical image with a Cartesian parametrization allows the system to converge, contrary to others, in difficult situations. We hence start to express the photometric visual servoing of a unicycle mobile robot with this

formulation. The interaction matrix linked to the intensity of a spherical point w.r.t. the mobile robot camera motion is $\mathbf{L}_{I_S(\mathbf{x}_S)} = -\nabla I_S^T \mathbf{L}_{\mathbf{x}_S}$, with (eq. (24)):

$$\mathbf{L}_{\mathbf{x}_S} = \begin{pmatrix} \frac{X_S Y_S}{\rho} & Y_S \\ \frac{Y_S^2 - 1}{\rho} & -X_S \\ \frac{Y_S Z_S}{\rho} & 0 \end{pmatrix}. \quad (31)$$

However, we saw, despite the large motion, PS-VS and CS-VS are rather close in terms of performance. So we define the interaction matrix between the motion of a point and the camera velocity for the pure spherical representation:

$$\mathbf{L}_S = \begin{pmatrix} -\frac{\sin \theta \cos \phi}{\rho} & 0 \\ -\frac{\cos \theta}{\rho \sin \phi} & -1 \end{pmatrix}. \quad (32)$$

This interaction matrix, deduced from equation (30), shows a decoupling property between the control of the translational and rotational velocities ([34]). This matrix substitutes the geometrical interaction matrix in equation (29) to get the interaction matrix linked to the intensity of a spherical point w.r.t. the camera motion.

5.2 Experiments

Pure spherical and Cartesian representations are compared. The former clearly shows theoretical advantages and the latter is practically more efficient in a six d.o.f. control, when a large motion is considered.

As in the six d.o.f. visual servoing, the desired omnidirectional image as a whole is the target to be reached, wherever are a particular object within the scene or within the image. Considering the mobile robot, the robot seems to move toward a target but this is inherent to the non-holonomic constraints of the considered system. Indeed, if a holonomic system was considered any kind of motion in any direction could have been considered as it was the case with the six d.o.f gantry robot.

5.2.1 Validation and evaluation experiments

In order to evaluate the behavior of CS-VS and PS-VS, a unique desired position (Figure 10) is to be reached from four different initial positions (Tab. 5). The positions and orientations of the robot, used in this section for the positioning error measurement, are obtained from an external localization tool, using infrared makers on the robot and a camera placed at the ceiling, looking down.

CS-VS and PS-VS succeed to position the robot at the desired place for four initial positions (one is given in figure 11). The difference of orientation between desired and initial pose of the robot has a more important impact than the difference in position. The fourth experimentation (Tab. 5) shows here a weak difference between initial and desired orientations. The servoing converges in this case but if we introduce a more important



Figure 10: Desired position. (a) The omnidirectional image acquired at this position and (b) corresponding external view of the mobile robot.

orientation difference, as 10° or 15° , the method did not allow to converge to the desired position.

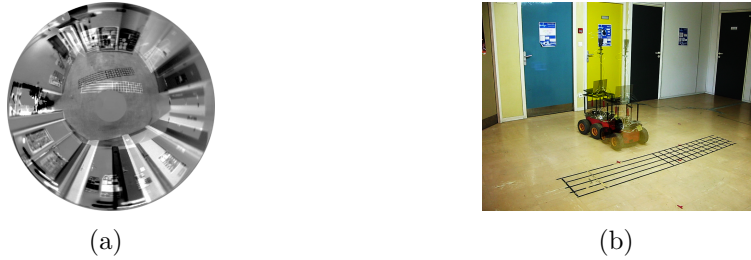


Figure 11: First initial position. (a) Difference between the desired and initial omnidirectional images. (b) Initial position of the robot with the desired one as a ghost (Figure 10(b)).

CS-VS and PS-VS converge to the desired robot position for initial errors presented in table 5. Contrary to the six d.o.f. Gantry robot, the final positioning is not so perfect w.r.t. the actual desired position (Tab. 6). These errors are due to the simple control law.

Even if some approximations are used on the control law or camera/robot axes alignment, the positioning is rather precise, particularly in orientation. Each representation has pros since the final position error is smaller in a mean for the CS-VS whereas the final orientation error is smaller for the PS-VS, in our experiments.

Table 5: Initial errors in the motion plane of the mobile robot. Δ_X and Δ_Y are in centimeters and Δ_{R_Z} in degrees.

exp	Δ_X	Δ_Y	Δ_{R_Z}
1	-48.64	-0.38	-17.56
2	-47.91	-1.43	9.44
3	48.24	12.99	16.70
4	46.17	-3.82	-4.37

Table 6: Final positioning errors at convergence of CS-VS and PS-VS in the motion plane of the mobile robot. Δ_X and Δ_Y are in centimeters and angular errors Δ_{Rz} in degrees.

method	exp	Δ_X	Δ_Y	Δ_{Rz}
CS-VS	1	1.11	1.98	0.76
	2	3.74	-2.22	0.64
	3	0.52	3.10	1.16
	4	2.32	0.49	0.96
PS-VS	1	1.44	3.35	0.79
	2	-0.60	-3.77	-0.15
	3	1.46	3.59	0.76
	4	1.21	-2.13	0.29

5.2.2 Large motion trajectory

In order to achieve the photometric visual servoing, the difference between the initial and desired images has to be rather low to stay under the hypothesis of the Optical Flow Constraint Equation (equations (17), (21) and (28)). That is why initial mobile robot positions tested until now are rather close to the desired position: about 50 cm and 18° maximum. When increasing the distance between initial and desired positions, the difference between the initial image and the desired one becomes more and more important as the initial position is far from the desired one.

To show the robustness of the method w.r.t. the hypotheses, this experiment propose an initial error of 1.20 m. The difference between the initial and the desired images is hence important (Fig. 12(b)).

This study allows us to define the maximum distance between images of the path to follow to make it successful (see section 5.3.2). Despite this important initial error, the CS-VS allows to position precisely the robot to the desired position with a final error of 0.5 cm and 0.3° . In the same situation, the PS-VS is defeated. At the beginning, the robot motion with PS-VS and CS-VS is similar but after starting to shift, the PS-VS is lost in a local minimum.

5.2.3 Two dof experimental results discussion

Table 5 shows the maximum orientation error, between initial and desired poses, leading to convergence. For the considered scene, it is around 18 degrees (around camera z axis). So if the robot starts from a larger orientation difference with respect to its desired one, the servoing may diverge (although, again, the convergence area is difficult to be quantitatively assessed). This is due to the fact that current and desired images are directly compared, pixel-to-pixel.

However, one can note that a rotation of the mobile robot is only producing a rotation of the omnidirectional image, keeping the same image up to a rotation angle, whereas it may lead to no overlapping at all for perspective images. To have more quantitative insight, one

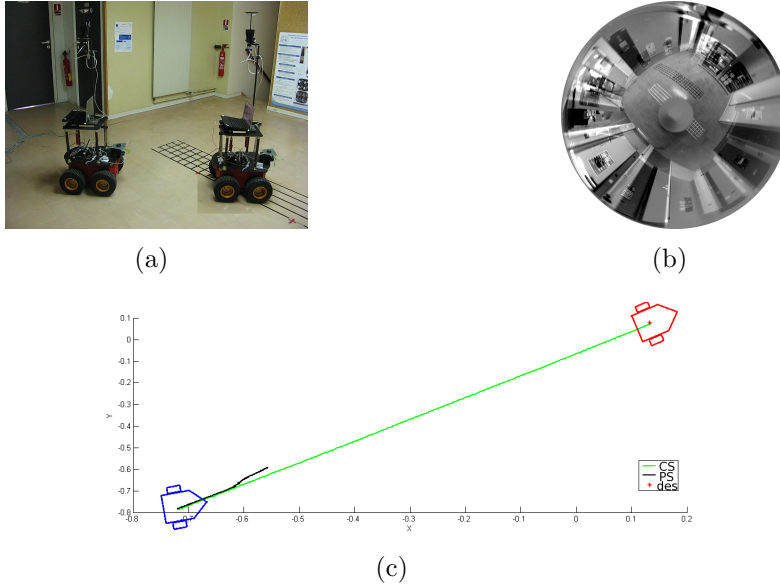


Figure 12: Photometric visual servoing from a far initial position. (a) shows the initial position (on the left side) and the desired one (on the right side) of the mobile robot. There is 1.20 m between these two positions, leading to an important difference between images acquired at these positions (b). (c) shows the robot trajectory obtained with CS-VS, converging to the good position, and PS-VS, lost in a local minimum. With the CS-VS, the final error is around 0.5 cm and 0.3°.

may consider the Gantry robot photometric visual servoing for which the use of a fisheye camera, with the modeling proposed in our paper, leads to convergence for a large initial error as visible in Figure 8: around 70 cm error in translation and 10 around two axes. The latter initial translation error is much larger than in the use of a perspective camera as in ([12]), where the maximum initial translation error shown in the paper is around 25 cm, one third of the maximum initial translation error in our paper for a 6 d.o.f. motion.

As a last complementary observation, Figure 12 shows a mobile robot two d.o.f. positioning experiment with 120 cm between initial and desired poses, almost twice larger than in the six d.o.f. case.

All these observations experimentally show the interest of considering a large field of view vision system for mobile robot navigation.

5.3 Application to navigation: visual path following

The idea in this experiment is to track a path defined by a set of images (a visual path). Our goal is not to localize the robot within its environment (visual odometry) but only to ensure that it is able to reproduce a visual path defined as a set of images previously acquired by the camera.

With respect to previous approaches that rely on 3D reconstruction (eg, [31]) or even

on appearance-based approaches ([32]), the learning step of the approach is simple. It does not require any feature extraction nor scene reconstruction: no image processing is done, only raw images are stored. The robot is driven manually along a desired path. While the robot is moving, the images acquired by the camera are stored chronologically thus defining a trajectory in the image space. Let us call I_0^*, \dots, I_N^* the key images that define this visual path.

The vehicle is initially positioned close to the initial position of the learned visual path (defined by the image I_0^*). The navigation is performed using a visual servoing task. In [31, 14, 32, 10] the considered control schemes are either pose-based control law or consider classical visual servoing process based on the use of visual features extracted from the current and key images (I and I_k^*).

In this work the approach proposed in the previous section is used to navigate from a waypoint (key-image) to another.

Considering reasonable mean distances and orientation differences between waypoints, we propose to apply the photometric visual servoing for the visual path following. A difficulty is to decide if a waypoint is reached to try to reach the next one. It is then necessary to know when the current and desired images are the most similar. The ZNCC: Zero-mean Normalized Cross Correlation ([18]) is a good criterion for this issue. The SSD, *i.e.* the cost function of the control law, has the drawback, in the navigation application, to be shifted when an illumination changing appears. This is not a problem for photometric visual servoing since even if the minimal value is not always the same, the corresponding optimal position is the same. So, a fixed threshold on the SSD measure to know if the desired image is reached would become wrong since the scene illumination can change. The ZNCC has, on the contrary, the same value even if the global illumination changes. So, if the ZNCC between the current and the desired image is greater than an experimentally chosen threshold (0.7 in experimentations) and corresponding to a quasi perfect alignment between two images, the servoing is considered ended for the desired image associated to the waypoint. The robot looks then for the next waypoint.

5.3.1 Visual path following

In this context, we considered that the translation velocity is a constant, leading to a one dof control problem, *i.e.* the rotation velocity. Thus, from the interaction matrix of a Cartesian point of a spherical image for a unicycle mobile robot (eq. (31)), we directly obtain the one concerning the rotation dof only:

$$\mathbf{L}_{\mathbf{x}_S} = \begin{pmatrix} Y_S \\ -X_S \\ 0 \end{pmatrix}. \quad (33)$$

The latter interaction matrix does not depend anymore on the point deepness ρ which has to be noted (it was approximated before). The composition of this Jacobian with the Cartesian spherical gradients leads to the interaction matrix linked to the intensity of a spherical point w.r.t. the mobile camera motion.

Dealing with the pure spherical formulation, the interaction matrix for a point for the visual path following is:

$$\mathbf{L}_s = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \quad (34)$$

which does not depend on the 3D point deepness too. The latter interaction matrix is furthermore constant, anywhere on the sphere. The photometric interaction matrix for the pure spherical representation is then obtained from the composition between the pure spherical gradients and the Jacobian of equation (34).

The advantage to consider the translation velocity as a constant is to lose the deepness parameter ρ of the interaction matrix. It only depends now on information directly obtained from the image: the backprojection of a pixel coordinates on the sphere and the spherical gradient.

5.3.2 Experimental results

Validation on a short curved path The visual memory is composed by an image sequence acquired in a learning stage in which the robot is manually driven. Omnidirectional images are acquired with 30 cm or 10° between them, using odometers of the robot. These 30 cm and 10° distances are empirically deduced from table 5. The experimentation is led over a curved trajectory of 3.5 m with 20 reference images. The interaction matrices computation is then done offline before starting the path following.

The path following, done by successive photometric visual servoing on each reference image, leads to slightly different trajectories if the spherical representation is used or the Cartesian one (Fig. 13). Positions and orientations of the robot are obtained with the external localization tool previously mentioned.

Application to a round trip After the validation of the visual path following in a small area but for a highly curved path, we changed the environment of the robot and also used another omnidirectional camera which is more compact. In this experiment, key images are still acquired every 30 cm or 10° but along a path of 20 m. The goal of the robot is to go to the end of its visual path and then to come back at its original position following the inverse visual path. So a total path length of 40 m has to be traveled. In addition to this more challenging path length, the environment is now rather textureless and natural light through windows at the end of the path is varying. People are also walking in the environment causing its partial occlusion. Despite these much more challenging conditions than in the previous experiment, the omnidirectional photometric visual path following succeeds with a correct precision in execution of the path as figure 14 qualitatively shows. It also demonstrates that a precise localization of the camera/robot along the path is not necessary to precisely achieve a navigation task.

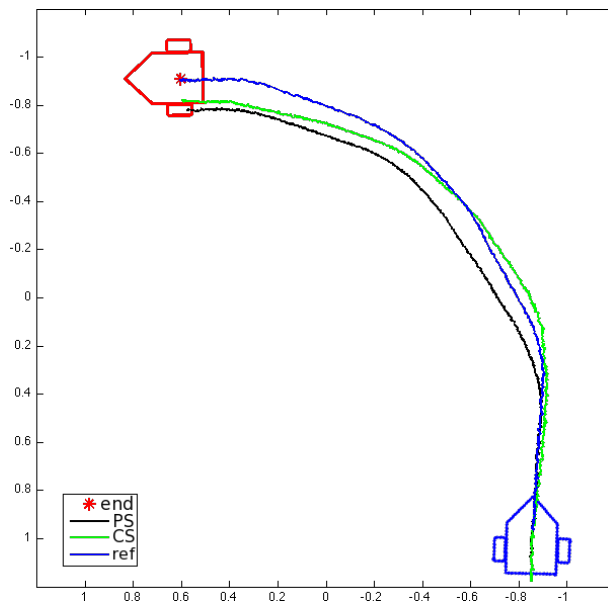


Figure 13: Robot trajectories for visual path following (CS-VS and PS-VS). The robot initial orientation corresponds to the vertical of this graphical representation. The learnt trajectory is blue, the visual path following with PS representation is black and green for the CS. The unit is the meter.

6 Video

A video demonstrating the omnidirectional photometric visual servoing of a Gantry robot and visual path following of a mobile robot is available in supplementary material as well as at the following URL: <http://mis.u-picardie.fr/~g-caron/videos/omniPhotoVSandPF.mp4>.

7 Conclusion

This paper has developed the first photometric omnidirectional visual servoing work, using entire image luminance. Different formulations were expressed and compared. Results on a gantry robot and on a mobile robot show the achievement of omnidirectional visual servoing using luminance as a feature. Formally valid, spherical visual servoing, particularly the Cartesian spherical, has shown better behavior than other studied formulations. The Cartesian spherical representation allows robots to reach a desired position that is far away from the initial one. Finally, we applied the photometric visual servoing to the visual path following problem showing the proposed method can be adapted to various applications and robots.

Of course, this kind of intensity based method to control a robot avoid feature detection and matching but the convergence volume is smaller. However, our simulation results and experiments show that the omnidirectional photometric visual control is able to converge from initial distances of around 75 cm. Furthermore, when converging, since the entire

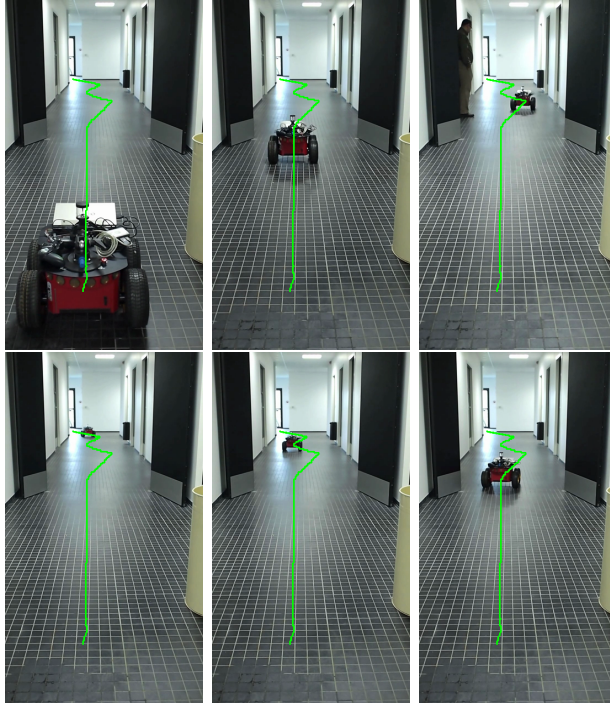


Figure 14: Visual path following. The first line shows robot positions from the beginning of the visual memory to the end and the second the inverse direction so that the robot comes back to its initial position after the round trip. In green is plotted the trajectory of the robot red part center during the learning process.

omnidirectional image is used, the robot positioning is very precise with positioning errors below one tenth of a millimeter in translation and 0.02° in rotation, for a Gantry robot.

References

- [1] M. Aranda, G. López-Nicolás, and C. Sagüés. Angle-based homing from a reference image set using the 1D trifocal tensor. *Autonomous Robots*, 34(1-2):73–91, 2013.
- [2] S. Baker and S. K. Nayar. A theory of single-viewpoint catadioptric image formation. *Int. Journal on Computer Vision*, 35(2):175–196, 1999.
- [3] J. P. Barreto and H. Araujo. Issues on the geometry of central catadioptric imaging. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 422–427, Hawaii, USA, December 2001.
- [4] J.P. Barreto, F. Martin, and R. Horaud. Visual servoing/tracking using central catadioptric images. *Int. Symp. on Experimental Robotics VIII.*, 2003.

- [5] H. Becerra, G. López-Nicolás, and C. Sagüés. Omnidirectional visual control of mobile robots based on the 1D trifocal tensor. *Robotics and Autonomous Systems*, 58(6):796–808, 2010.
- [6] S. Benhimane and E. Malis. Homography-based 2d visual tracking and servoing. *Int. Journal of Robotics Research*, 26(7):661–676, 2007.
- [7] G. Caron, E. Marchand, and E. Mouaddib. Omnidirectional photometric visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 6202–6207, Taipei, Taiwan, October 2010.
- [8] F. Chaumette. Image moments: a general and useful set of features for visual servoing. *IEEE Trans. on Robotics*, 20(4):713–723, August 2004.
- [9] F. Chaumette and S. Hutchinson. Visual servo control, Part I: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82–90, December 2006.
- [10] Z. Chen and S.T. Birchfield. Qualitative vision-based path following. *IEEE Transactions on Robotics*, 25(3):749–754, 2009.
- [11] G. Chesi and K Hashimoto, editors. *Visual servoing via advanced numerical methods*. LNCIS 401. Springer, 2010.
- [12] C. Collewet and E. Marchand. Photometric visual servoing. *IEEE Trans. on Robotics*, 27(4):828–834, 2011.
- [13] J. Courbon, Y. Mezouar, L. Eck, and P. Martinet. A generic fisheye camera model for robotic applications. In *IEEE/RSJ Int. Conf. on Intelligent Robots and System*, pages 1683–1688, San Diego, USA, October 2007.
- [14] J. Courbon, Y. Mezouar, and P. Martinet. Autonomous navigation of vehicles from a visual memory using a generic camera model. *IEEE Trans. on Intelligent Transportation Systems*, 10(3):392–402, 2009.
- [15] A. Dame and E. Marchand. Mutual information-based visual servoing. *IEEE Trans. on Robotics*, 27(5):958–969, 2011.
- [16] K. Deguchi. A direct interpretation of dynamic images with camera and object motions for vision guided robot control. *Int. Journal on Computer Vision*, 37(1):7–20, 2000.
- [17] C. Demonceaux and P. Vasseur. Omnidirectional image processing using geodesic metric. In *Int. Conf. on Image Processing*, pages 221–224, Cairo, Egypt, November 2009.
- [18] L. Di Stephano, S. Mattoccia, and F. Tombari. ZNCC-based template matching using bounded partial correlation. *Pattern Recognition Letters*, 26:2129–2134, 2005.
- [19] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.

- [20] T. Hamel and R. Mahony. Visual servoing of an under-actuated dynamic rigid-body system: An image-based approach. *IEEE Trans. on Robotics and Automation*, 18(2):187–198, 2002.
- [21] B. Horn and Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
- [22] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12:651–670, 1996.
- [23] V. Kallem, M. Dewan, J.P. Swensen, G.D. Hager, and N.J. Cowan. Kernel-based visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and System*, pages 1975–1980, San Diego, USA, October 2007.
- [24] P. Lébraly, C. Deymier, O. Ait-Aider, E. Royer, and M. Dhome. Flexible extrinsic calibration of non-overlapping cameras using a planar mirror: Application to vision-based robotics. In *IEEE/RSJ Int. Conf. on Intelligent Robots and System*, pages 5640–5647, Taipei, Taiwan, October 2010.
- [25] E. Malis. Improving vision-based control using efficient second-order minimization techniques. In *IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 1843–1848, New Orleans, April 2004.
- [26] E. Marchand and F. Chaumette. Feature tracking for visual servoing purposes. *Robotics and Autonomous Systems*, 52(1):53–70, July 2005.
- [27] E. Marchand, F. Spindler, and F. Chaumette. Visp for visual servoing: a generic software platform with a wide class of robot control skills. *IEEE Robotics and Automation Magazine*, 12(4):40–52, December 2005.
- [28] G. Mariottini and D. Prattichizzo. Image-based visual servoing with central catadioptric cameras. *Int. J. of Robotics Research*, 27(1):41–56, 2008.
- [29] M. Meilland, A. I. Comport, and P. Rives. Dense visual mapping of large scale environments for real-time localization. In *IEEE/RSJ Int. Conf. on Intelligent Robots and System*, pages 4242–4248, San Francisco, CA, sept. 2011.
- [30] S.K. Nayar, S.A. Nene, and H. Murase. Subspace methods for robot vision. *IEEE Trans. on Robotics and Automation*, 12(5):750–758, October 1996.
- [31] E. Royer, M. Lhuillier, M. Dhome, and J.M. Lavest. Monocular vision for mobile robot localization and autonomous navigation. *Int. Journal of Computer Vision*, 74(3):237–260, 2007.
- [32] S. Segvic, A. Remazeilles, A. Diosi, and F. Chaumette. A mapping and localization framework for scalable appearance-based navigation. *Computer Vision and Image Understanding*, 113(2):172–187, February 2009.

- [33] O. Tahri, Y. Mezouar, F. Chaumette, and P. Corke. Decoupled image-based visual servoing for cameras obeying the unified projection model. *IEEE Trans. on Robotics*, 26(4):684–697, 2010.
- [34] R. Tatsambon Fomena, H. Yoon, A. Cherubini, F. Chaumette, and S. Hutchinson. Coarsely calibrated visual servoing of a mobile robot using a catadioptric vision system. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 5432–5437, St Louis, USA, October 2009.
- [35] X. Ying and Z. Hu. Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model. In *Eur. Conf. on Comp. Vision*, pages 442–455, Prague, Czech Republic, May 2004.