Photometric visual servoing

Christophe Collewet, Eric Marchand

Photometric visual servoing

Christophe Collewet, Eric Marchand

***Abstract*—This paper proposes a new way to achieve robotic tasks by 2D visual servoing. Indeed, instead of using classical geometric features such as points, straight lines, pose or a homography, as is usually done, the luminance of all pixels in the image is here considered. The main advantage of this new approach is that it does not require any tracking or matching process. The key point of our approach relies on the analytic computation of the interaction matrix. This computation is based either on a temporal luminance constancy hypothesis or on a reflection model so that complex illumination changes can be considered. Experimental results on positioning and tracking tasks validate the proposed approach and show its robustness to approximated depths, low textured objects, partial occlusions and specular scenes. They also showed that luminance leads to lower positioning errors than a classical visual servoing based on 2D geometric visual features.**

I. Introduction

Visual servoing consists aims at controlling the motions of a robot by using data provided by a vision sensor [1]. More precisely, to achieve a visual servoing task, a set of visual features has to be selected from the image allowing to control the desired degrees of freedom (d.o.f). A control law is then designed so that these visual features $\mathbf{s}$ reach desired values $\mathbf{s}^*$. The control principle is thus to regulate the error vector $\mathbf{e}=\mathbf{s}-\mathbf{s}^*$ to zero. To build the control law, the knowledge of the interaction matrix $\mathbf{L_s}$ is usually required [1].

Visual features are always designed from visual measurements $\mathbf{m}(\mathbf{p}_k)$ (where $\mathbf{p}_k$ is the camera pose at time k) which requires a robust extraction, matching (between $\mathbf{m}(\mathbf{p}_0)$ and $\mathbf{m}(\mathbf{p}^*)$, where $\mathbf{p}^*$ is the desired camera pose) and real-time spatio-temporal tracking (between $\mathbf{m}(\mathbf{p}_{k-1})$ and $\mathbf{m}(\mathbf{p}_k)$). However, this process is a complex task, as testified by the abundant literature on the subject (see [2] for a recent survey), and is considered as one of the bottleneck of the expansion of visual servoing. Thus several works focus to alleviate this problem. An interesting way to avoid any tracking process is to use non geometric visual measurements as in [3], [4] instead of geometric measurements as it is usually done. Of course, directly using non geometric visual features also avoids any tracking process. In that case, parameters of a 2D motion model have been used in [5]–[8]. Nevertheless, such approaches require a complex image processing task.

In this paper we show that this tracking process can be totally removed and show that no other information than the image intensity (the pure luminance signal) needs to be considered to control the robot motion. Indeed, to achieve this goal we use as visual measurement and as visual feature the simplest that can be considered: the image intensity itself. We therefore call this new approach *photometric visual servoing*. In that case, the visual feature vector $\mathbf{s}$ is nothing but the image while $\mathbf{s}^*$ is the desired image.

Considering the image intensity as a feature has been considered previously [9], [10]. However, those works differ from our approach in two important points. First, they 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 not directly based on the image intensity. Second, the interaction matrix related to the eigenspace is not computed analytically but learned during an off-line step. This learning process has two drawbacks: it has to be done for each new object and requires the acquisition of many images of the scene at various camera positions. Considering an analytical interaction matrix, as we propose avoids these issues. An interesting approach, which also directly considers the pixels intensity, has been recently proposed in [11]. However, only the translations and the rotation around the optical axis have been considered (that is the 4 most simple d.o.f.) whereas, in our work, the 6 degrees of freedom are controlled. However, an image processing step is still required. Our approach does not require this step.

In this paper, we summarize several previous works. In [12], the analytic computation of the interaction matrix related to the luminance for a Lambertian scene is provided, only positioning tasks have been considered. In [13], this matrix has been computed considering a lighting source mounted on the camera and the use of the Blinn-Phong illumination model (a simplified model of the Phong model detailed in the next section), only tracking tasks have been considered. In [14], the Phong model has been used, only positioning tasks have been considered. In addition, these works refer to [15] where details concerning analytic computations are given. Note that in [16], although this is also a direct visual servoing approach, the considered features used in the control law are very different. In this paper, we specifically focus on the way the visual servoing problem has been turned into an optimization problem. More precisely, we analytically analyse the cost function to minimize in order to derive an efficient control law. Moreover, additional experimental results than those described in our previous works are presented, as for example a comparison between classical 2D geometric visual features and the use of the luminance. We will show that by using luminance much lower positioning errors can be obtained.

The remainder of this paper os organized as follows, we first compute the interaction matrix of the luminance in Section II. Then, we reformulate the visual servoing problem into an

C. Collewet and E. Marchand are with INRIA Rennes - Bretagne Atlantique, IRISA, Lagadic team, Rennes, France. E-mail: firstname.name@irisa.fr.

Part of this paper has been published in the IEEE Int. Conf. on Robotics and Automation, ICRA'08, ICRA'09 and in IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'08.

optimization problem in Section III, and propose a new control law dedicated to our cost function to minimize. Section IV shows experimental results on various scenes for several tasks.

II. Luminance as a visual feature

The visual features considered in this paper are the luminance of each point of the image. More precisely, we consider as visual features the luminance $I_{\mathbf{x}}$ at a constant pixel location $\mathbf{x} = (x, y)$ for all $\mathbf{x}$ belonging to the image domain and for a given pose $\mathbf{p}$. Thus, we have

$$\mathbf{s}(\mathbf{p}) = \mathbf{I}_{\mathbf{x}}(\mathbf{p}) = (\mathbf{I}_{1\bullet}, \mathbf{I}_{2\bullet}, \cdots, \mathbf{I}_{N\bullet}) \tag{1}$$

where $\mathbf{I}_{i\bullet}$ is the i-th line of the image. $\mathbf{I}_{\mathbf{x}}(\mathbf{p})$ is then a vector of size $k = N \times M$ where $N \times M$ is the size of the image.

As mentioned in the introduction, an estimation of the interaction matrix is required to control the robot motion. In our case, we are looking for the interaction matrix $\mathbf{L}_{I_{\mathbf{x}}}$ related to the luminance $I_{\mathbf{x}}(t)$ at time t, that is

$$\dot{I}_{\mathbf{x}} = \mathbf{L}_{I_{\mathbf{x}}} \mathbf{v} \tag{2}$$

with $\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega})$ where $\boldsymbol{v}$ is the linear camera velocity and $\boldsymbol{\omega}$ its angular velocity.

Let us consider a particular point $P(t)$ belonging to the scene which projects into the camera plane at the point $p(t)$. $P(t)$ is time varying either because the camera is moving with respect to the scene or because the scene is moving itself with respect to the camera. Let us note that, unless explicitly stated otherwise, all the quantities are expressed in the camera frame. The computation of the interaction matrix (2) requires to write the total derivative of the luminance $I(p(t), t)$ in p at time t

$$\dot{I}(p(t), t) = \boldsymbol{\nabla} I(p(t), t)^\top \dot{p}(t) + \frac{\partial I(p(t), t)}{\partial t}. \tag{3}$$

However, considering that, at time t, the normalized coordinates of $p(t)$ coincide with $\mathbf{x}$, (3) becomes

$$\dot{I}(p(t), t) = \boldsymbol{\nabla} I_{\mathbf{x}}(t)^\top \dot{\mathbf{x}} + \dot{I}_{\mathbf{x}}(t) \tag{4}$$

with $\boldsymbol{\nabla} I_{\mathbf{x}}(t)$ the spatial gradient of $I_{\mathbf{x}}(t)$ and $\dot{\mathbf{x}}$ the 2D velocity of $p(t)$.

Therefore, to explicitly compute the interaction matrix $\mathbf{L}_{I_{\mathbf{x}}}$, an illumination model is required to estimate $\dot{I}(p(t), t)$.

The simplest one is, of course, the one that lied on the temporal luminance constancy hypothesis [17], as it is the case in most of computer vision applications. In that case, we simply have $\dot{I}(p(t), t) = 0$ and it becomes straightforward to derive the interaction matrix from (4) and (2) (see [12] for further details). In that case, we obtain

$$\mathbf{L}_{I_{\mathbf{x}}} = -\nabla I^\top \mathbf{L}_{\mathbf{x}} \tag{5}$$

where the interaction matrix $\mathbf{L}_{\mathbf{x}}$ related to $\mathbf{x}$ (i.e. such that $\dot{\mathbf{x}} = \mathbf{L}_{\mathbf{x}} \mathbf{v}$) has been introduced

$$\mathbf{L}_{\mathbf{x}} = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix}. \tag{6}$$

Of course, because of the temporal luminance constancy hypothesis, (5) is only valid for Lambertian scenes, that is for surfaces reflecting the light with the same intensity in each direction. Besides, (5) is only valid when the lighting source is not moving with respect to the scene. In fact, it is well known that the temporal luminance constancy hypothesis can be easily violated [18]. Therefore, to derive the interaction matrix, we have to consider a more realistic reflection model than the Lambert's one [1]. In this paper, we derive the interaction matrix from the Phong illumination model [20]. This model is not based on physical laws, but comes from the computer graphics community. Although empirical, it is widely used thanks to its simplicity, and also because it is appropriate for various types of materials, whether they are rough or smooth.

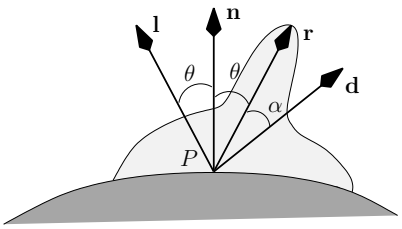

Fig. 1. Quantities involved in the Phong illumination model [20] (expressed here in the scene frame).

According to the Phong model (see Fig. 1), the intensity $I(p(t), t)$ at point p and at time t writes as follows

$$I(p(t), t) = K_s \cos^\eta \alpha + K_d \cos\theta + K_a. \tag{7}$$

This relation is composed of a diffuse, a specular and an ambient component and assumes a point light source. The scalar K_s describes the specular component of the lighting; K_d describes the weight of the diffuse term which depends on the *albedo* in $P(t)$; K_a is the intensity of ambient lighting in $P(t)$. Note that K_s, K_d and K_a depend on $P(t)$. θ is the angle between the normal to the surface $\mathbf{n}$ in $P(t)$ and the direction of the light source $\mathbf{l}$; α is the angle between $\mathbf{r}$ (which is $\mathbf{l}$ mirrored about $\mathbf{n}$) and the viewing direction $\mathbf{d}$; $\mathbf{r}$ can be seen as the direction due to a pure specular object. The parameter η allows to model the width of the specular lobe around $\mathbf{r}$, this scalar varies as the inverse of the roughness of the material.

Considering that $\mathbf{r}, \mathbf{d}$ and $\mathbf{l}$ are normalized, we can rewrite (7) as

$$I(p(t), t) = K_s {u_1}^\eta + K_d u_2 + K_a \tag{8}$$

where $u_1 = \mathbf{r}^\top \mathbf{d}$ while we have $u_2 = \mathbf{n}^\top \mathbf{l}$. Note that these vectors are easy to compute, since we have $\mathbf{d} = -\frac{\tilde{\mathbf{x}}}{\|\tilde{\mathbf{x}}\|}$ and $\mathbf{r} = 2u_2\mathbf{n} - \mathbf{l}$ with $\tilde{\mathbf{x}} = (x, y, 1)$.

Consequently, from (8), it becomes easy to compute $\dot{I}(p(t), t)$ involved in (4) (we assume here that the scene is only constituted by one material)

$$\dot{I}(p(t), t) = \eta K_s u_1^{\eta-1} \dot{u_1} + K_d \dot{u_2} \tag{9}$$

that leads to a general formulation of *the optical flow constraint equation* [17] considering the Phong illumination model

$$\boldsymbol{\nabla} I_{\mathbf{x}}^\top \mathbf{L}_{\mathbf{x}} \mathbf{v} + \dot{I}_{\mathbf{x}} = \eta K_s u_1^{\eta-1} \dot{u_1} + K_d \dot{u_2}. \tag{10}$$

[1] Indeed, the Lambert's model can only explain the behavior of non homogeneous opaque dielectric material [19]. It only describes a diffuse reflection component and does not take into account the viewing direction.

Thereafter, by explicitly computing the total time derivative of u_1 and u_2 and writing $\dot{u_1}$ as $\dot{u_1} = \mathbf{L}_1^\top \mathbf{v}$ and $\dot{u_2}$ as $\dot{u_2} = \mathbf{L}_2^\top \mathbf{v}$ where $\mathbf{L}_1$ and $\mathbf{L}_2$ are 6-dimensional vectors, we obtain the interaction matrix related to the intensity at pixel $\mathbf{x}$ in the general case [2]

$$\mathbf{L}_{I_\mathbf{x}} = -\boldsymbol{\nabla} I^\top \mathbf{L}_\mathbf{x} + \eta K_s {u_1}^{\eta-1} \mathbf{L}_1^\top + K_d \mathbf{L}_2^\top. \quad (11)$$

To compute the vectors $\mathbf{L}_1$ and $\mathbf{L}_2$ involved in (11) we have to explicitly express $\dot{u_1}$ and $\dot{u_2}$. However, to do that, we have to assume some hypothesis about how $\mathbf{n}$ and $\mathbf{l}$ move with respect to the camera. Various cases have been studied in [15]. Due to lack of place, we report here only the case of an eye-in-hand robot system where the light source is mounted on the camera and where the interaction matrix is computed at the desired position. It is a very classical way to proceed [1]. Indeed, it avoids to compute on-line 3D information like the depths for example. We consider here this case. More precisely, we consider that, at the desired position the depth of all the points where the luminance is measured are equal to a constant value Z^*. That means that we consider that the object is planar and that the camera and the object planes are parallel at this position. Moreover, we consider a directional lighting source. In these conditions, all computations done (see [15]), the interaction matrix related to the luminance writes simply as

$$\widehat{\mathbf{L}}_{I_\mathbf{x}} = \frac{\eta K_s {u_1}^{\eta-1}}{\|\tilde{\mathbf{x}}\|} \left[\frac{x}{\bar{Z}} \;\; \frac{y}{\bar{Z}} \;\; -\frac{x^2+y^2}{\bar{Z}} \;\; y \;\; -x \;\; 0 \right] - \boldsymbol{\nabla} I^\top \mathbf{L}_\mathbf{x}. \quad (12)$$

where $\bar{Z} = Z^* \| \tilde{\mathbf{x}} \|^2$.

Note that this matrix requires the computation of $\boldsymbol{\nabla} I$, let us point out that it is the *only* image processing step necessary to implement our method.

On the other hand, even if the analytical computation of the vectors $\mathbf{L}_1$ and $\mathbf{L}_2$ is not straightforward in the general case, their final expression is very simple and easy to compute in this particular case.

III. Visual servoing control law

The interaction matrix associated to the luminance being known, the control law can now be derived. For that, we have turned the visual servoing problem into an optimization problem where the goal was to minimize the following cost function [12]

$$\mathcal{C}(\mathbf{p}) = \frac{1}{2} \| \mathbf{e} \|^2 \quad (13)$$

where $\mathbf{e} = \mathbf{I}_\mathbf{x}(\mathbf{p}) - \mathbf{I}_\mathbf{x}(\mathbf{p}^*)$.

However, it is well known that in visual servoing, some image motions due to particular camera motions are not observable. Of course, to derive an efficient control law such camera motions must be avoided.

In the next section we will first prove that such motions exist, compute them, and then propose a control law that will ensure a high decrease of the cost function.

[2]Note that we recover the interaction matrix $-\boldsymbol{\nabla} I^\top \mathbf{L}_\mathbf{x}$ associated to the intensity under temporal constancy (see (5)), i.e. in the Lambertian case ($K_s = 0$) and when $\dot{u_2} = 0$ (the lighting direction is motionless with respect to the point P).

A. Analysis of the cost function

At the desired position, thanks to a first order Taylor series expansion of the visual features $\mathbf{I}_\mathbf{x}(\mathbf{p})$ around $\mathbf{p}^*$, an approximation of the cost function in a neighborhood of $\mathbf{p}^*$ can be obtained (see [12], [15] for more details):

$$\widehat{\mathcal{C}(\mathbf{p})} = \frac{1}{2} (\mathbf{v}\Delta t)^\top \mathbf{H}^* (\mathbf{v}\Delta t). \quad (14)$$

where

$$\mathbf{H}^* = \mathbf{L}_{\mathbf{I}_\mathbf{x}^*}^\top \mathbf{L}_{\mathbf{I}_\mathbf{x}^*}. \quad (15)$$

is the Hessian matrix at $\mathbf{p}^*$.

Since $\mathbf{L}_{\mathbf{I}_\mathbf{x}^*}$ is analytically known, $\mathbf{H}^*$ is also known and (14) can be easily evaluated. However, we will consider here only the case where the temporal luminance constancy hypothesis is valid and, first, when the camera and the object planes are parallel. In this case, we will denotes $\mathbf{H}^*$ by $\mathbf{H}_\parallel^*$. We will consider the more complex case when these planes are not parallel afterwards.

By considering the relation between the normalized coordinates $\mathbf{x}$ and their pixel value $\mathbf{u} = (u, v)$, a line of $\mathbf{L}_{\mathbf{I}_\mathbf{x}^*}$ writes simply at first order in h

$$\begin{aligned} \mathbf{L}_{I_x^*} = \big(& \nabla I_x/Z^*,\ \nabla I_y/Z^*,\ -h(m\nabla I_x + n\nabla I_y)/Z^*, \\ & -\nabla I_y,\ \nabla I_x,\ -(n\nabla I_x + m\nabla I_y)h \big) \end{aligned} \quad (16)$$

where $x = m\,h$ and $y = n\,h$ have been substituted in (5) with $m = u - u_0$, $n = v - v_0$ where (u_0, v_0) is the principal point of the camera and $h = 1/F$ with $F = f/\mu$, f being the focal length and μ the size of a pixel (supposed to be square).

Note that since F is a high value, the first order Taylor series expansion (16) is valid. From (16), $\mathbf{H}_\parallel^*$ is easily obtained

$$\mathbf{H}_\parallel^* = \begin{bmatrix} \frac{h_{11}}{{Z^*}^2} & \frac{h_{12}}{{Z^*}^2} & h\frac{h_{13}}{{Z^*}^2} & -\frac{h_{12}}{Z^*} & \frac{h_{11}}{Z^*} & h\frac{h_{16}}{Z^*} \\ \frac{h_{12}}{{Z^*}^2} & \frac{h_{22}}{{Z^*}^2} & h\frac{h_{23}}{{Z^*}^2} & -\frac{h_{22}}{Z^*} & \frac{h_{12}}{Z^*} & h\frac{h_{26}}{Z^*} \\ h\frac{h_{13}}{{Z^*}^2} & h\frac{h_{23}}{{Z^*}^2} & 0 & -h\frac{h_{23}}{Z^*} & h\frac{h_{13}}{Z^*} & 0 \\ -\frac{h_{12}}{Z^*} & -\frac{h_{22}}{Z^*} & -h\frac{h_{23}}{Z^*} & h_{22} & -h_{12} & -h\,h_{26} \\ \frac{h_{11}}{Z^*} & \frac{h_{12}}{Z^*} & h\frac{h_{13}}{Z^*} & -h_{12} & h_{11} & h\,h_{16} \\ h\frac{h_{16}}{Z^*} & h\frac{h_{26}}{Z^*} & 0 & -h\,h_{26} & h\,h_{16} & 0 \end{bmatrix} \quad (17)$$

where the h_{ij} are functions of the image gradients computed at each pixel that is not useful to detail.

From (17), it is easy to show that the rank of $\mathbf{H}_\parallel^*$ is 4 since the first line is obtained from the fifth line divided by Z^* and the second line from the fourth divided by $-Z^*$. Therefore, whatever the image content is, 0 is a double eigenvalue, and its associated eigenvectors denoted $\mathbf{e}_1$ and $\mathbf{e}_2$ are simply generated by the kernel of $\mathbf{H}_\parallel^*$

$$\text{Ker}\ \mathbf{H}_\parallel^* = \left\{ \begin{pmatrix} -Z^* & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & Z^* & 0 & 1 & 0 & 0 \end{pmatrix} \right\}.$$

That means that along any direction $\mathbf{d}_\parallel = \gamma_1\,\mathbf{e}_1 + \gamma_2\,\mathbf{e}_2$ (with γ_1 and γ_2 non null scalars), the approximated cost function (14) does not vary and therefore that the true cost function (13)

will slowly vary. Note that these motions coincide with what it is observed in practice, it is always possible to compensate a x (respectively y) axis translational motion with a y (respectively x) axis rotational motion to keep an image almost constant from a different point of view. In addition, note that $\mathbf{d}_{\|}$ does not depend at all on the image content, moreover $\mathbf{d}_{\|}$ is a constant value.

The problem is now to find a direction that highly decreases the cost function. Since $\mathbf{d}_{\|}$ is a constant value, we search a direction orthogonal to $\mathbf{d}_{\|}$. Such a direction can be simply given by $\nabla\mathcal{C}(\mathbf{p})$ since near the desired pose $\mathbf{p}^*$ we have

$$\widehat{\nabla\mathcal{C}}(\mathbf{p})^\top \mathbf{d}_{\|} = (\mathbf{v}\Delta t)^\top {\mathbf{H}^*_{\|}}^\top \mathbf{d}_{\|} = 0 \tag{18}$$

(we recall that $\mathbf{d}_{\|} \in \text{Ker } \mathbf{H}^*_{\|}$).

Note that, in practice, this direction is also valid even quite far from $\mathbf{p}^*$ as will be proved by the experimental results.

Now we investigate the more complex case where the camera desired position is no more parallel to the object plane. In that case, the depths are given by $1/Z = ax + by + c$ and the matrix $\mathbf{H}^*$ writes now as $\mathbf{H}^* = \mathbf{H}^*_{\|} + \mathbf{H}^*_{\angle}$ where

$$\mathbf{H}^*_{\angle} = \begin{bmatrix} -2b_{11}c\,h & -2b_{12}c\,h & 0 & b_{12}\,h & -b_{11}\,h & 0 \\ -2b_{12}c\,h & -2b_{22}c\,h & 0 & b_{22}\,h & -b_{12}\,h & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ b_{12}\,h & b_{22}\,h & 0 & 0 & 0 & 0 \\ -b_{11}\,h & -b_{12}\,h & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{19}$$

with $b_{ij} = a\alpha_{ij} + b\beta_{ij}$ where α_{ij} and β_{ij} are functions of all the image gradients computed at each pixels. As in the previous case where the camera desired position was parallel to the object plane, the rank of $\mathbf{H}^*$ is still 4 and the same conclusions can be led: whatever the image content is, 0 is a double eigenvalue the associated eigenvectors of which are still generated by the kernel of $\mathbf{H}^*$

$$\text{Ker } \mathbf{H}^* = \left\{ \begin{pmatrix} -1/c & 0 & \mu_1 & 0 & 1 & \mu_2 \end{pmatrix}, \begin{pmatrix} \mu_3 & 1/c & \mu_4 & 1 & \mu_5 & 0 \end{pmatrix} \right\}$$

where the μ_k ($k \in \{1, \cdots, 5\}$) are functions of the h_{ij} and the b_{ij}. Those are not useful to detail.

Here again, that means that there are some directions $\mathbf{d}$ generated by Ker $\mathbf{H}^*$ where the cost function (13) slowly varies. Moreover, as previously, $\nabla\mathcal{C}(\mathbf{p})$ is an optimal direction since near $\mathbf{p}^*$, $\nabla\mathcal{C}(\mathbf{p})$ and $\mathbf{d}$ are orthogonal

$$\widehat{\nabla\mathcal{C}}(\mathbf{p})^\top \mathbf{d} = (\mathbf{v}\Delta t)^\top {\mathbf{H}^*}^\top \mathbf{d} = 0. \tag{20}$$

B. Design of the control law

We propose the following algorithm to reach the minimum of the cost function. The camera is first moved in the direction of $\nabla\mathcal{C}$ to highly decrease the cost function and next, to a direction according to $\mathbf{d}$ to explore the remainder 2-dimensional subspace to reach its minimum. The first step can be easily done by using a steepest descent approach. However, if the direction of $\nabla\mathcal{C}(\mathbf{p})$ is almost constant, its amplitude is not constant and may even vary very slowly in practice. To cope with this problem we propose to use the following control law

$$\mathbf{v} = -v_c \frac{\nabla\mathcal{C}(\mathbf{p}_{init})}{\| \nabla\mathcal{C}(\mathbf{p}_{init}) \|}. \tag{21}$$

That is, a constant velocity with norm v_c is applied in the steepest descent computed at the initial camera pose. Consequently, this first step behaves as an open-loop system. To turn into a closed-loop system, we first detect roughly the minimum along the direction of $\nabla\mathcal{C}$ from a 3rd order polynomial filtering of $\mathcal{C}(\mathbf{p})$ and then apply a control law formally equal to the one used in the Levenberg-Marquardt approach. We denote MLM this method in the remainder of the paper. The resulting control law is then given by (see [15])

$$\mathbf{v} = -\lambda \left(\mathbf{H} + \mu \,\text{diag}(\mathbf{H})\right)^{-1} \widehat{\mathbf{L}_{\mathbf{I}_\mathbf{x}}}^\top (\mathbf{I}_\mathbf{x}(\mathbf{p}) - \mathbf{I}_\mathbf{x}(\mathbf{p}^*)) \tag{22}$$

where λ and μ are positive scalars, $\mathbf{H}$ is the Hessian matrix at $\mathbf{p}$.

More precisely, first, a high value for μ is used in (22) (typically $\mu = 1$) to turn the control law into a steepest descent like approach[3] and to reach the minimum along the direction of $\nabla\mathcal{C}$. Once this minimum has been reached, a lower value is used (typically $\mu = 10^{-2}$) to switch continuously to a Gauss-Newton (GN) control law (commonly used in visual servoing, see [1]) in order to explore the remainder 2-dimensional subspace generated by $\mathbf{d}$ and to reach the minimum of the cost function. This way to proceed ensure both a high convergence rate and a correct robot path.

Remarks about the stability: since redundant visual features (that is $k > 6$ considering a 6 d.o.f. robot) have been used, as it is also the case in classical visual servoing, only the local stability can be obtained (see [1] for a proof). However, as pointed out in [1], this domain can be quite large in practice. In addition, since we use redundant visual features, it is clear that the potential dimension of the null space can be high. However, that does not mean at all that all the motions that belongs to this null space are feasible, see for example [21]. They use redundant visual features but prove that there are no local minima.

IV. Experimental results

In all the experiments reported here, the camera is mounted on a 6 degrees of freedom gantry robot. Control law is computed in real-time on a Core 2 Duo 3Gz PC running Linux. Images are acquired at 66Hz using an IEEE 1394 camera with a resolution of 320×240[4]. The size of the vector $\mathbf{I}_\mathbf{x}$ is then 76800. Despite this size, the matrix $\widehat{\mathbf{L}}_{\mathbf{I}_\mathbf{x}}$ can be computed at each iteration if needed.

[3]More precisely, each component of the gradient is scaled according to the diagonal of the Hessian, which leads to larger displacements along the direction where the gradient is low.

[4]Note that if a higher resolution is used, the computations time of the error vector and of the interaction matrix will be highly increased, it is thus better to decrease their size by decreasing the image resolution. There is no real advantage to use high resolution images for control issues.

A. Positioning tasks using the basic temporal luminance constancy model

We assume in this section that the temporal luminance constancy hypothesis is valid, i.e. we use the interaction matrix given in (5). In order to make this assumption as valid as possible, a diffuse lighting has been used so that the luminance can be considered as constant wrt to the viewing direction. Moreover, the lighting is also motionless wrt the scene being observed. In this section, we will first compare the GN and MLM methods, then compare the photometric visual servoing with respect to a classical approach based on a matching and a tracking process and, finally, we will show that the photometric visual servoing is robust.

Comparison between the GN and the MLM method. The goal of the first experiment is to compare the control laws based on GN (the usual visual servoing control law) and MLM approaches when a planar object (a photo) is considered. The initial error pose was $\mathbf{\Delta p}_{init} = (5$ cm, -23 cm, 5 cm, -12.5°, -8.4°, -15.5°$)$ [5]. The desired pose was so that the object and CCD planes are parallel at $Z = Z^* = 80$ cm. The interaction matrix has been computed at each iteration but assuming that all the depths are constant and equal to Z^* (i.e. Eq. (5) with $Z = Z^*$), which is of course a coarse approximation.

Fig. 2a depicts the behavior of the cost functions using the GN method or the MLM method while Fig. 2b depicts the trajectories (expressed in the desired frame) when using either the GN or the MLM method. Fig. 2c and Fig. 2d depict respectively the camera velocity. The initial and final images are shown respectively, in Fig. 2e and Fig. 2f. First, as can be seen in Fig. 2a, both control laws converge since the cost functions vanish[6]. However, the time-to-convergence with the GN method is much higher than that of the MLM method. The trajectory when using the GN method is also shaky compared to the one of the MLM method (Fig. 2b). The velocity of the camera when using the MLM method is smoother than when using the GN method (Fig. 2d and Fig. 2c). This experiment clearly shows that the MLM method outperforms the GN one. Note that in both cases the positioning errors is very low, for the MLM method we obtained $\mathbf{\Delta p} = (0.26$ mm, 0.30 mm, 0.03 mm, 0.02°, -0.02°, 0.03°$)$. It is very difficult to reach so low positioning errors when using geometric visual features as it is usually done. Indeed, this nice result is obtained because $\mathbf{e}$ is very sensitive to the pose $\mathbf{p}$.

Comparison with a feature matching process. Considering such images, it is also possible to extract geometric features like SIFT or SURF and match them between current and desired images. In this experiment we use SURF features (with the OpenCV implementation) and use these points coordinates within a classical 2D visual servoing control law and within a robust one [22]. The later allows to reject wrong matched features directly at the control level. 200 points have been extracted in the desired image and between 50 and 100 points are matched at each iteration and considered in the control law. The goal is here to compare the precision of the positioning between these two classical approaches and the new proposed one. Table I shows the obtained results. As expected the positioning error is far lower using the new proposed approach. Translation error (that is $\|\mathbf{t}\|$) is only 0.07 mm (our robot precision is 0.1 mm) which has to be compared to the 0.87 mm and 1.28 mm using the two other methods. Indeed, in a classical approach, an extraction process obviously introduces errors in the features coordinates which implies imprecisions in the positioning task. Similar results have been obtained from other initial positions and other scenes.

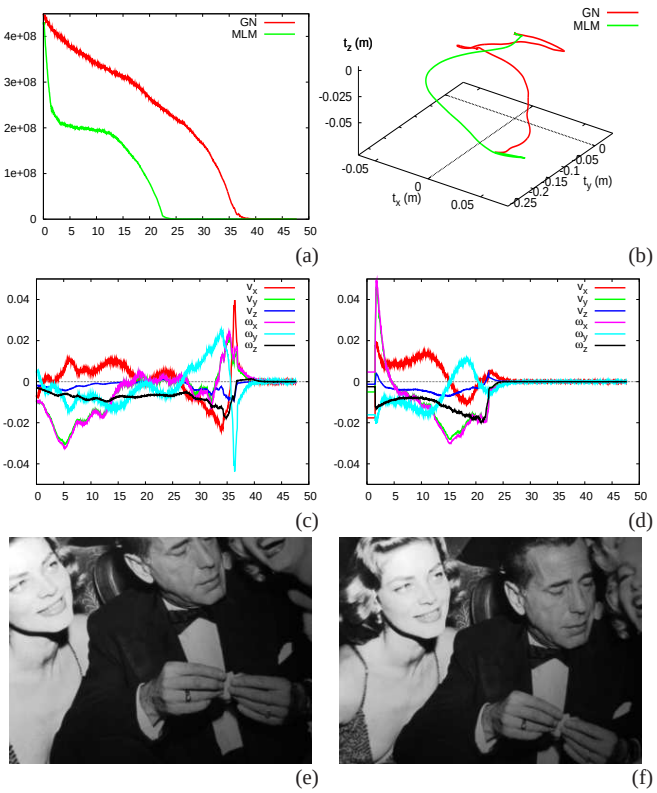

Fig. 2. First experiment. MLM vs. GN method (x axis in second). (a) Comparison of cost functions, (b) Comparison of camera trajectories, (c) Camera velocities (m/s or rad/s) for the GN method, (d) Camera velocities (m/s or rad/s) for the MLM method, (e) Initial image, (f) Final image.

Behavior with respect to partial occlusions. This experiment deals with partial occlusions. The desired object pose as well as the initial pose are unchanged. After having moved the camera to its initial position, an object has been added to the scene, so that the initial image is now the one shown in Fig. 3a and the desired image is still the one shown in Fig. 2f. Moreover, as seen in Fig. 3b and Fig. 3c, the object introduced in the scene is also moved by hand during the camera motion which highly increases the occluded surface. Despite that, the control law still converges (see Fig. 3f). Of course, since the desired image is not the true one, the error cannot vanish at the end of the motion (see Fig. 3f). Nevertheless, the final positioning error is not affected by the occlusions since we

[5]The following notation has been used: $\mathbf{\Delta p} = (\mathbf{t}, \mathbf{u}\theta)$ where $\mathbf{t}$ describes the translation part of the homogeneous matrix related to the transformation from the current to the desired frame, while its rotation part is expressed under the form $\mathbf{u}\theta$ where $\mathbf{u}$ represents the unit rotation axis vector and θ the rotation angle around this axis.

[6]In fact, the cost functions do not exactly vanish, the mean error of the intensity levels is 2.2 (with a standard deviation of 0.4) at the end of the motion for the MLM method, this error is due to the acquisition noise and not due to the positioning error which, as we shall see, is very low.

TABLE I
FINAL POSITIONING ERROR: WE COMPARED THE PROPOSED APPROACH WITH APPROACHES BASED ON GEOMETRIC FEATURE EXTRACTION.

	$\mathbf{t}$ (in mm)			$\mathbf{u}\,\theta$ (in degrees)		
Initial error	143.042	-177.517	12.496	-16.083	-10.139	-1.517
Photometric VS	-0.027	0.042	-0.049	0.001	-0.001	-0.006
SURF (2D VS)	0.486	0.558	-0.467	0.041	-0.046	-0.018
SURF (2D robust VS)	1.072	-0.560	-0.418	-0.056	-0.027	-0.074

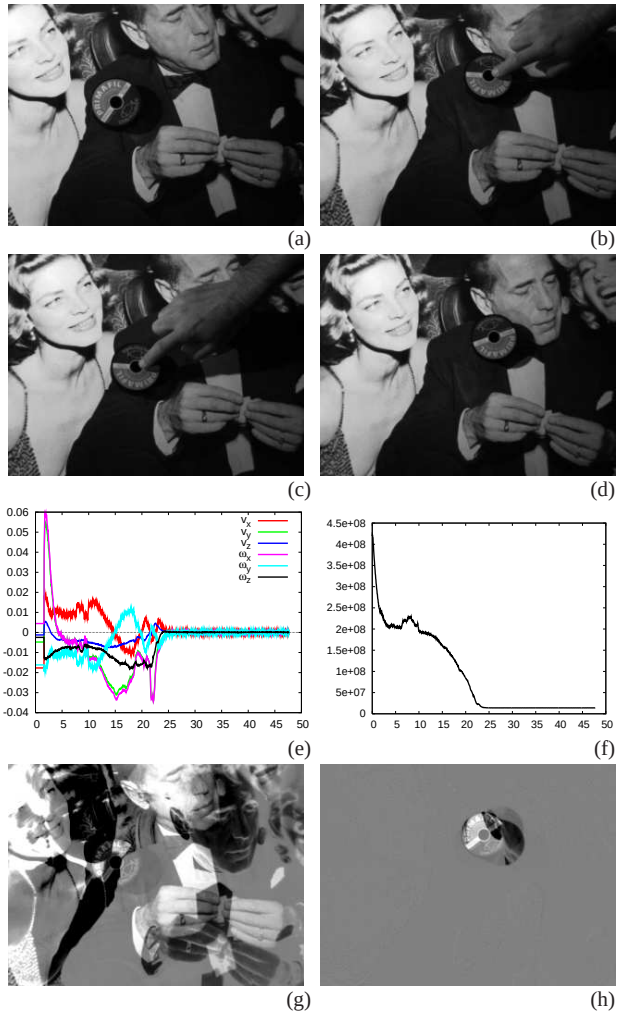

Fig. 3. Second experiment. Partial occlusions (x axis in second). (a) Initial image, (b) Image at t $\approx$ 11 s, (c) Image at t $\approx$ 13 s (d) Final image, (e) Camera velocities (m/s or rad/s), (f) Cost function, (g) $\mathbf{I_x} - \mathbf{I_x^*}$ at the initial position, (h) $\mathbf{I_x} - \mathbf{I_x^*}$ at the end of the motion.

have $\mathbf{\Delta p}$ = (-0.1 mm, 2 mm, 0.3 mm, 0.13°, 0.04°, 0.07°). It is very similar with the previous experiments. This very nice behavior is due to the high redundancy of the visual features we use.

Robustness with respect to non planar scenes. The goal of this third experiment is to show the robustness of the control law wrt non planar scenes (see Fig. 4). This figure shows that large errors in the depth are introduced (the height of the castle tower is around 30 cm). The initial and desired poses are still unchanged. Fig. 5 depicts this experiment. Here again, the control law still converges (despite the fact that $\widehat{\mathbf{L}_{\mathbf{I_x}}}$ has been computed with a constant depth Z^* = 80 cm) and the positioning error is still low since we have $\mathbf{\Delta p}$ = (0.2 mm, -0.0 mm, 0.1 mm, -0.01°, 0.00°, 0.06°).

Fig. 4. The non planar scene used in Fig. 5 experiment.

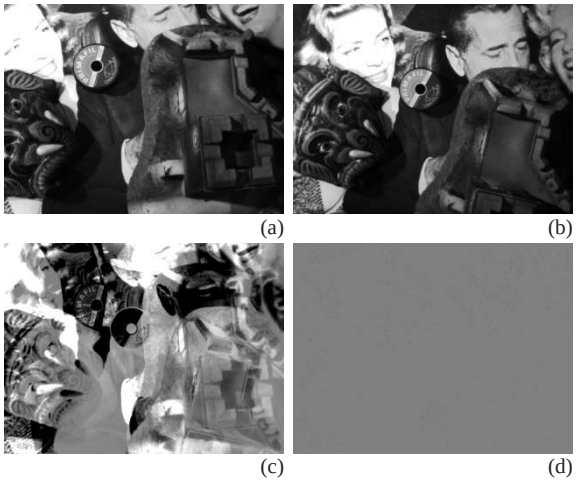

Fig. 5. Third experiment. Robustness wrt depths. (a) Initial image, (b) Final image, (c) $\mathbf{I_x} - \mathbf{I_x^*}$ at the initial position, (d) $\mathbf{I_x} - \mathbf{I_x^*}$ at the end of the motion.

Influence of the image content. The goal of these last set of experiments is to show that, even if the luminance is used as a visual feature, our approach does not depend too much on the texture of the scene being observed. Fig. 6 depicts the behavior of our algorithm for different planar objects (the initial as well as the desired pose is unchanged). As can be seen, the control law converges in each case, and even in the case of a low textured scene. Let us point out that similar positioning errors than for the first experiment have been obtained.

B. Tracking tasks

Our goal is now to perform a tracking task with respect to a moving object. That is, we have to maintain a rigid link between the target to track and the camera. Considering that the scene is moving, a specific illumination model has to be considered as explained in section II. A directional light-ring is located around the camera lens[7]. The scene is then no more illuminated by a diffuse lighting. The object is unchanged but it is attached to a motorized rail that allows to control its

[7]The nature of the light is directional because LEDs with a small emission angle have been used.

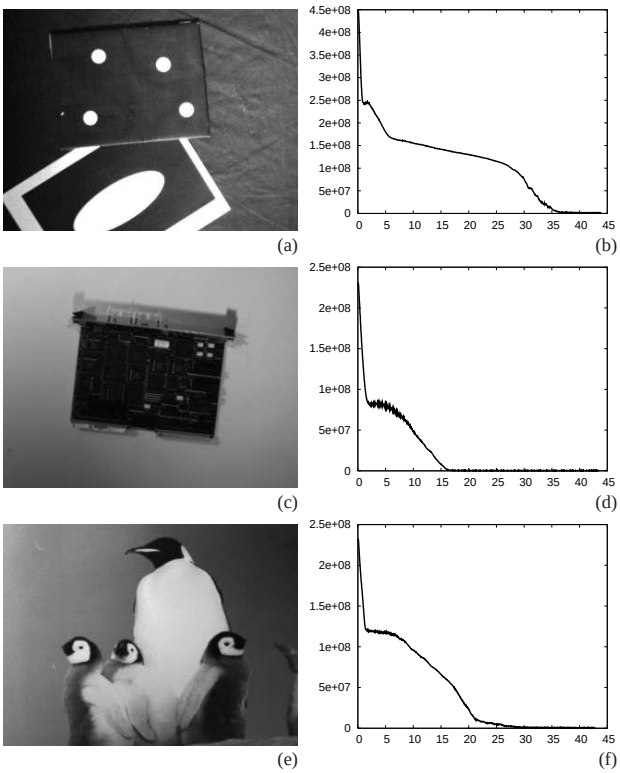

Fig. 6. Fourth experiment. Same positioning task wrt to various objects. Objects considered (left column); Cost functions (right column) (x axis in second).

motion (see [15] for an object moved by hand). Although only one d.o.f of the object is controlled (with a motion that is completely unknown from the tracking process), the 6 d.o.f of the robot are controlled (the object velocity is 1 cm/s). Since we have a constant target velocity, a simple integrator, as in [23], has been introduced in the control law to eliminate the steady state tracking error.

In this experiment, we use equation (12) to compute $\widehat{\mathbf{L}_{\mathbf{I}_\mathbf{x}}}$. In this relation, two parameters depending on the object surface are required: K_s and η. However, in practice, a large domain of values for these parameters has led to good results. The same values ($K_s = 200$ and $\eta = 200$) have been used for all our experiments and never been changed despite the fact that various material has been considered (glass, various plastics, metal, glossy and matt paper), see [13] and [14] for other experiments using a complex illumination model. When the velocity is constant, the object is perfectly tracked, as can be seen on Figure 7a where $\| \mathbf{e} \|$ is depicted, despite the occurrence of a large specularity which shows the importance of using a complex illumination model. Let us note that without using this new model, experiments fail [15]. Error in the image remains small except when the object stops or accelerates (corresponding to the peaks in Figure 7a). For each pixel, except during accelerations and decelerations, $|I_\mathbf{x} - I^*_\mathbf{x}| < 5$. The camera velocity (see Fig. 7b) shows a pure motion along the x ($\pm$ 1cm/s) axis that corresponds to the ground truth.

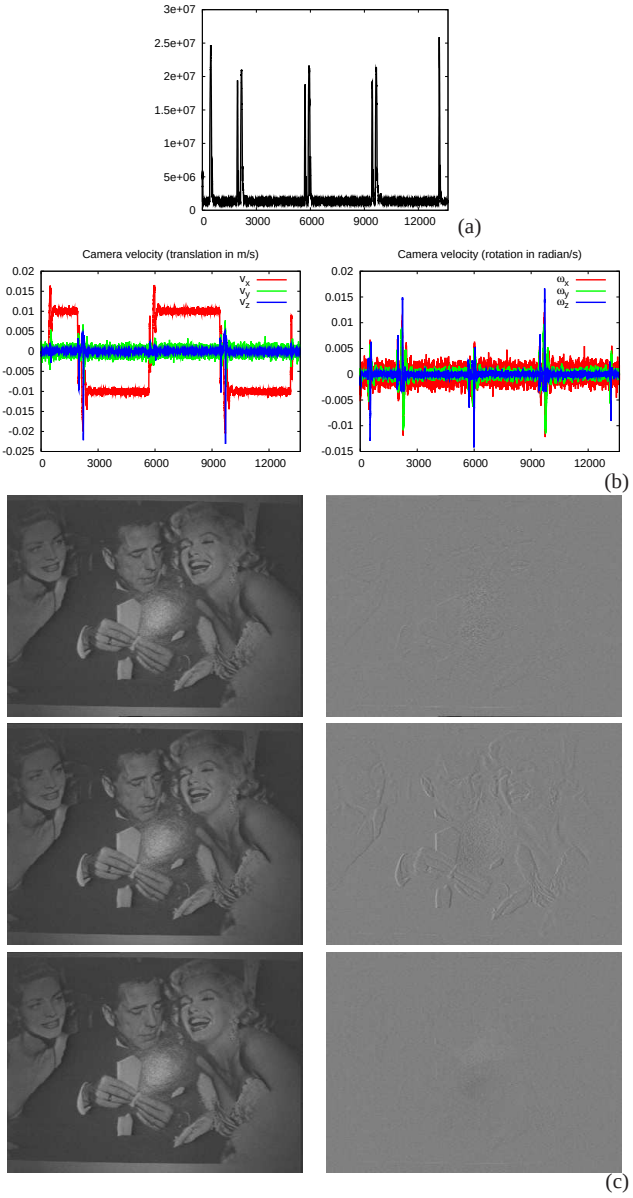

Fig. 7. Target tracking considering the complete interaction matrix that integrates specularity, diffuse and ambient terms (x axis in frame number). (a) Error $\| \mathbf{I_x} - \mathbf{I^*_x} \|$, (b) Camera velocity (m/s and radian/s), (c) Images at different time (left) and corresponding errors $\mathbf{I_x} - \mathbf{I^*_x}$ (right).

V. Conclusion

We have shown in this paper that it is possible to use directly the luminance of all the pixels in an image as visual features in visual servoing. To the best of our knowledge this is the first time that visual servoing has been handled without any complex image processing task (except the image spatial gradient required for the computation of the interaction matrix), nor learning step. Indeed, unlike classical visual servoing where geometrical features are used, using photometric visual servoing does not need any matching between the initial and desired features, nor between the current and the previous features. It is a very important issue when complex scenes have to be considered. Our approach has been validated on various scenes and various lightings for positioning or tracking tasks. Concerning positioning tasks, the positioning error is always

very low and much lower than classical visual servoing based on 2D geometric visual features. Supplementary advantages are that our approach is not sensitive to partial occlusions and to coarse approximations of the depths required to compute the interaction matrix.

ACKNOWLEDGEMENTS

The authors wish to thank François Chaumette and Seth Hutchinson for their constructive comments.

REFERENCES

[1] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, December 2006.

[2] E. Marchand and F. Chaumette, "Feature tracking for visual servoing purposes," *Robotics and Autonomous Systems*, vol. 52, no. 1, pp. 53–70, June 2005, special issue on "Advances in Robot Vision", D. Kragic, H. Christensen (Eds.).

[3] E. Malis and S. Benhimane, "A unified approach to visual tracking and servoing," *Robotics and Autonomous Systems*, vol. 1, no. 52, pp. 39–52, 2005.

[4] S. Benhimane and E. Malis, "Homography-based 2d visual tracking and servoing," *Int. Journal of Robotics Research*, vol. 26, no. 7, pp. 661–676, July 2007.

[5] P. Questa, E. Grossmann, and G. Sandini, "Camera self orientation and docking maneuver using normal flow," in *SPIE AeroSense'95*, vol. 2488, Orlando, Florida, USA, April 1995, pp. 274–283.

[6] V. Sundareswaran, P. Bouthemy, and F. Chaumette, "Exploiting image motion for active vision in a visual servoing framework," *Int. Journal of Robotics Research*, vol. 15, no. 6, pp. 629–645, June 1996.

[7] J. Santos-Victor and G. Sandini, "Visual behaviors for docking," *Computer Vision and Image Understanding*, vol. 67, no. 3, pp. 223–238, September 1997.

[8] A. Crétual and F. Chaumette, "Visual servoing based on image motion," *Int. Journal of Robotics Research*, vol. 20, no. 11, pp. 857–877, November 2001.

[9] S. Nayar, S. Nene, and H. Murase, "Subspace methods for robot vision," *IEEE Trans. on Robotics*, vol. 12, no. 5, pp. 750–758, October 1996.

[10] K. Deguchi, "A direct interpretation of dynamic images with camera and object motions for vision guided robot control," *Int. Journal of Computer Vision*, vol. 37, no. 1, pp. 7–20, June 2000.

[11] V. Kallem, M. Dewan, J. Swensen, G. Hager, and N. Cowan, "Kernel-based visual servoing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and System, IROS'07*, San Diego, USA, October 2007.

[12] C. Collewet, E. Marchand, and F. Chaumette, "Visual servoing set free from image processing," in *IEEE Int. Conf. on Robotics and Automation, ICRA'08*, Pasadena, California, May 2008, pp. 81–86.

[13] C. Collewet and E. Marchand, "Modeling complex luminance variations for target tracking," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'08*, Anchorage, Alaska, June 2008, pp. 1–7.

[14] ——, "Photometry-based visual servoing using light reflexion models," in *IEEE Int. Conf. on Robotics and Automation, ICRA'09*, Kobe, Japan, May 2009.

[15] ——, "Photometric visual servoing," INRIA, Tech. Rep. No. 6631, September 2008.

[16] A. Dame and E. Marchand, "Entropy-based visual servoing," in *IEEE Int. Conf. on Robotics and Automation, ICRA'09*, Kobe, Japan, May 2009, pp. 707–713.

[17] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, August 1981.

[18] A. Verri and T. Poggio, "Motion field and optical flow: qualitative properties," *IEEE Trans. on PAMI*, vol. 11, no. 5, pp. 490–498, May 1989.

[19] J. Reichmann, "Determination of absorption and scattering coefficients for non homogeneous media," *Applied Optics*, vol. 12, pp. 1811–1815, 1973.

[20] B. Phong, "Illumination for computer generated pictures," *Communication of the ACM*, vol. 18, no. 6, pp. 311–317, June 1975.

[21] F. Schramm, G. Morel, A. Micaelli, and A. Lottin, "Extended 2d visual servoing," in *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, vol. 1, New Orleans, USA, 2004, pp. 267–273.

[22] A. Comport, E. Marchand, and F. Chaumette, "Statistically robust 2D visual servoing," *IEEE Trans. on Robotics*, vol. 22, no. 2, pp. 415–421, apr 2006.

[23] F. Chaumette, P. Rives, and B. Espiau, "Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing," in *IEEE Int. Conf. on Robotics and Automation*, vol. 3, Sacramento, California, USA, April 1991, pp. 2248–2253.