



HAL
open science

Modeling complex luminance variations for target tracking

Christophe Collewet, Eric Marchand

► **To cite this version:**

Christophe Collewet, Eric Marchand. Modeling complex luminance variations for target tracking. IEEE Int. Conf. on Computer Vision and Pattern Recognition, Jun 2008, Anchorage, United States. inria-00304345

HAL Id: inria-00304345

<https://inria.hal.science/inria-00304345>

Submitted on 23 Jul 2008

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.

Modeling complex luminance variations for target tracking

Christophe Collewet and Eric Marchand
INRIA Rennes - Bretagne Atlantique, IRISA
Campus Universitaire de Beaulieu, 35 042 Rennes Cedex, France

firstname.name@irisa.fr

Abstract

Lambert's model is widely used in low level computer vision algorithms such as matching, tracking or optical flow computation for example. However, it is well known that these algorithms often fail when they face complex luminance variations. Therefore, we revise in this paper the underlying hypothesis of its temporal constancy and propose a new optical flow constraint. To do that, we use the Blinn-Phong reflection model to take into account that the scene may move with respect to the lighting and/or to the observer, and that specular highlights may occur. To validate in practice these analytical results, we consider the case where a camera is mounted on a robot end-effector with a lighting mounted on this camera and show experimental results of target tracking by visual servoing. Such an approach requires to analytically compute the luminance variations due to the observer motion which can be easily derived from our revised optical flow constraint. In addition, while the visual servoing classical approaches rely on geometric features, we present here a new method that directly relies on the luminance of all pixels in the image which does not require any tracking or matching process.

1. Introduction

Most of computer vision-based applications rely on low level algorithms such as matching, features tracking, or optical flow computations for example. Such approaches are based on a relation that links the luminance of a physical point \mathbf{P} at time t to its luminance at a later time $t + dt$ due to a relative motion of the observer with respect to that scene or to other events like illumination changes. Nevertheless, due to the complexity of this relation, most often, the algorithms aforementioned are based on the temporal luminance constancy hypothesis:

$$I(\mathbf{x} + \mathbf{dx}, t + dt) = I(\mathbf{x}, t) \quad (1)$$

where $\mathbf{x} = (x, y, 1)$ is the perspective projection of \mathbf{P} assuming that \mathbf{x} has a displacement $\mathbf{dx} = (dx, dy, 0)$ in the

time interval dt . A first order Taylor series expansion of (1) around \mathbf{x} yields the so-called *optical flow constraint equation* (OFCE) [7]

$$\dot{I} = \nabla I^\top \dot{\mathbf{x}} + I_t = 0 \quad (2)$$

with $I_t = \partial I / \partial t$ and ∇I the spatial gradient of I .

However, it is well known that this constraint can be easily violated [14], e.g. if the orientation of a Lambertian surface is changing with respect to the lighting. Consequently, many authors have addressed this issue. The basic assumption of the luminance constancy, can be extended by the spatial gradient constancy assumption as proposed in [13]. Dealing with target tracking, under the assumption of a Lambertian scene, illumination variations can be taken into account [5]. This Lambertian hypothesis may be removed when modeling the illumination changes as a surface that evolves over time [12]. This latter work can be seen as related to [1] where the intensity variation between dt is expressed as a mixture of causes.

A general framework has been proposed in [10] leading to the following relation:

$$I(\mathbf{x} + \mathbf{dx}, t + dt) = (1 + m(\mathbf{x}, t))I(\mathbf{x}, t) + c(\mathbf{x}, t). \quad (3)$$

However, the coefficients $m(\mathbf{x}, t)$ and $c(\mathbf{x}, t)$ are considered as locally constant, and then are estimated numerically.

Nevertheless, all these works do not rely on a time-varying physical reflection model.

A very interesting approach, close to our paper, can be found in [6]. Indeed, a generalization of (2) is presented leading to:

$$\nabla I^\top \dot{\mathbf{x}} + I_t = \frac{dI}{dt} \quad (4)$$

where dI/dt is analytically computed according to several physical models of brightness variation. In particular, they study the case of a non planar Lambertian surface undergoing to a rotation motion.

In this paper, we will analytically compute the term dI/dt involved in (4) in the more general case where the

scene is not Lambertian. Indeed, we will use the Blinn-Phong reflection model [2] to tackle more complex illumination changes such as the occurrence of specularities. This will consequently yield a revised optical flow constraint. To validate this new constraint, we will consider a 3D target tracking task by visual servoing. Briefly, it consists in using the data provided by a camera in order to control the motion of a robotic system to achieve positioning or tracking tasks [8]. Note that, since our goal is to maintain a rigid link between the camera and the target to track, such task requires to control the 6 degrees of freedom of the camera. In addition, this paper proposes a new way to achieve target tracking by visual servoing. Indeed, instead of using geometric features as it is usually done, we use directly the luminance of all pixels in the image. Since our approach relies directly on image intensity, it does not require any more any 2D tracking nor matching.

In this case, since the surface is moving with respect to the lighting, it is clear that the assumption of temporal luminance constancy is violated and that, consequently, a more complex modeling is necessary. The term dI/dt we propose in this paper is also necessary to compute the so-called interaction matrix which is in the heart of the matter. Indeed, it is the relation that links the variation of the image intensity to the camera motion.

The remainder of this paper is organized as follows. First, we revise the optical flow constraint in the case of the Blinn-Phong reflection model in Section 2. We then recall some fundamentals of visual servoing and derive the control law to achieve the tracking task in Section 3. Section 4 shows experimental results related to this task.

2. Revised optical flow constraint

To derive the term dI/dt involved in (4), we have to consider a more realistic reflection model than the Lambert's one. We first consider the well known Phong model [11] (see Fig. 1)

$$I(\mathbf{x}) = K_s \cos^k \alpha + K_d \cos \theta + K_a. \quad (5)$$

It is composed of a diffuse, a specular and an ambient component and assumes a point light source. The scalar k is inversely proportional to the roughness of the object surface; K_s describes the specular component of the lighting; K_d describes the part of the diffuse term which depends on the *albedo* in \mathbf{P} ; K_a is the intensity of ambient lighting in \mathbf{P} . Note that K_s, K_d and K_a only depend on \mathbf{P} . θ is the angle between the normal to the surface \mathbf{n} in \mathbf{P} 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{V} . Although empirical, this model is widely used thanks to its simplicity, and because it is appropriate for various types of materials, whether they are rough or smooth.

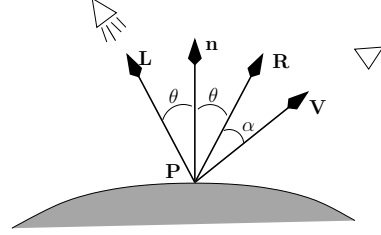


Figure 1. Phong model in the general case [11].

Considering that \mathbf{R}, \mathbf{V} and \mathbf{L} are normalized, we thus have

$$I(\mathbf{x}) = K_s(\mathbf{R}^\top \mathbf{V})^k + K_d \mathbf{n}^\top \mathbf{L} + K_a. \quad (6)$$

Since \mathbf{R} may be hard to compute, following Blinn work [2], this equation can be simplified (see Fig. 2) by considering for α the angle between \mathbf{n} and \mathbf{H} where \mathbf{H} is the halfway vector between \mathbf{L} and \mathbf{V} . This leads to

$$I(\mathbf{x}) = K_s(\mathbf{n}^\top \mathbf{H})^k + K_d \mathbf{n}^\top \mathbf{L} + K_a \quad (7)$$

with

$$\mathbf{V} = -\frac{\mathbf{x}}{\|\mathbf{x}\|} \quad \text{and} \quad \mathbf{H} = \frac{\mathbf{L} + \mathbf{V}}{\|\mathbf{L} + \mathbf{V}\|}. \quad (8)$$

Denoting $u_1 = \mathbf{n}^\top \mathbf{H}$ and $u_2 = \mathbf{n}^\top \mathbf{L}$, (7) becomes

$$I(\mathbf{x}) = K_s(u_1)^k + K_d u_2 + K_a \quad (9)$$

which gives

$$\dot{I} = k K_s u_1^{k-1} \dot{u}_1 + K_d \dot{u}_2. \quad (10)$$

Consequently, (4) becomes

$$\nabla I^\top \dot{\mathbf{x}} + I_t = k K_s u_1^{k-1} \dot{u}_1 + K_d \dot{u}_2. \quad (11)$$

Although, it is possible to explicitly compute \dot{u}_l ($l = 1$ or 2) in the general case, we will restrict ourselves to more specific cases that will prove to be useful during the experiments.

2.1. Light source on the camera

We focus in this section on the case where the light source is mounted on the camera with the same direction as the optical axis (see Fig. 3). We thus have $\mathbf{L} = -\mathbf{k}$ (where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are the unit vectors corresponding to the axis of the camera frame).

To explicitly compute (10), we have to compute \dot{u}_l :

$$\dot{u}_l = \frac{\partial u_l}{\partial x} \dot{x} + \frac{\partial u_l}{\partial y} \dot{y} + \frac{\partial u_l}{\partial t}. \quad (12)$$

Considering the specular term u_1 , we have

$$\begin{cases} \frac{\partial u_1}{\partial x} &= \mathbf{n}^\top \frac{\partial \mathbf{H}}{\partial x} + \mathbf{H}^\top \frac{\partial \mathbf{n}}{\partial x} \\ \frac{\partial u_1}{\partial y} &= \mathbf{n}^\top \frac{\partial \mathbf{H}}{\partial y} + \mathbf{H}^\top \frac{\partial \mathbf{n}}{\partial y} \\ \frac{\partial u_1}{\partial t} &= \mathbf{H}^\top \frac{\partial \mathbf{n}}{\partial t} \end{cases} \quad (13)$$

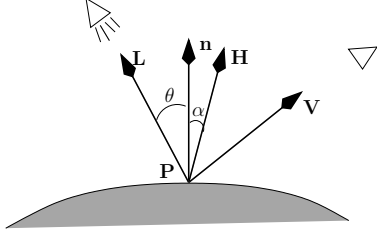


Figure 2. Blinn-Phong model in the general case [2].

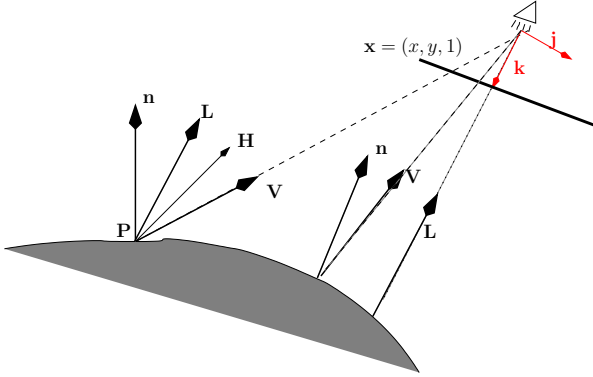


Figure 3. Blinn-Phong model for a non-planar object in the specific case where camera and light source are located at the same position with the same direction.

while considering u_2 , we have (since $\partial \mathbf{L} / \partial x = \partial \mathbf{L} / \partial y = 0$)

$$\begin{cases} \frac{\partial u_2}{\partial x} = -\mathbf{k}^\top \frac{\partial \mathbf{n}}{\partial x} \\ \frac{\partial u_2}{\partial y} = -\mathbf{k}^\top \frac{\partial \mathbf{n}}{\partial y} \\ \frac{\partial u_2}{\partial t} = -\mathbf{k}^\top \frac{\partial \mathbf{n}}{\partial t} \end{cases} \quad (14)$$

Note that we clearly see on the above equations the contribution introduced by a non planar surface (i.e., when $\partial \mathbf{n} / \partial x \neq 0$ and $\partial \mathbf{n} / \partial y \neq 0$).

To compute $\partial \mathbf{n} / \partial t$, we introduce the matrix ${}^c \mathbf{R}_o$ which describes the rotation between the camera and the object frames such that $\mathbf{n} = {}^c \mathbf{R}_o {}^o \mathbf{n}$ where ${}^o \mathbf{n}$ express \mathbf{n} in the object frame. Therefore,

$$\frac{\partial \mathbf{n}}{\partial t} = {}^c \dot{\mathbf{R}}_o {}^o \mathbf{n} = {}^c \dot{\mathbf{R}}_o {}^c \mathbf{R}_o^\top \mathbf{n} = -\boldsymbol{\omega} \times \mathbf{n} \quad (15)$$

where $\boldsymbol{\omega}$ is the camera angular velocity expressed in its own frame. Consequently, by using (13) or (14) in (12), we have

$$\begin{aligned} \dot{u}_1 &= \mathbf{n}^\top \frac{\partial \mathbf{H}}{\partial x} \dot{x} + \mathbf{n}^\top \frac{\partial \mathbf{H}}{\partial y} \dot{y} + (\mathbf{H} \times \mathbf{n})^\top \boldsymbol{\omega} \\ &+ \underbrace{\mathbf{H}^\top \frac{\partial \mathbf{n}}{\partial x} \dot{x} + \mathbf{H}^\top \frac{\partial \mathbf{n}}{\partial y} \dot{y}}_{\text{non planar component}} \end{aligned} \quad (16)$$

and

$$\dot{u}_2 = (\mathbf{n} \times \mathbf{k})^\top \boldsymbol{\omega} - \underbrace{\mathbf{k}^\top \frac{\partial \mathbf{n}}{\partial x} \dot{x} - \mathbf{k}^\top \frac{\partial \mathbf{n}}{\partial y} \dot{y}}_{\text{non planar component}} \quad (17)$$

Finally, if we use (16) and (17) in (11), we obtain the revised optical flow constraint when the light source direction coincides with the observer direction.

2.2. Light source on the camera and the image plane parallel to a planar target

In this section, we consider the following particular case where the light source is still mounted on the camera (with the same direction than the optical axis) and where the image plane is parallel to a planar target, i.e. $\mathbf{L} = \mathbf{n} = -\mathbf{k}$. It becomes then possible to explicitly compute $\mathbf{n}^\top \partial \mathbf{H} / \partial x$, $\mathbf{n}^\top \partial \mathbf{H} / \partial y$ and $(\mathbf{H} \times \mathbf{n})^\top \boldsymbol{\omega}$ involved in (16). We obtain after tedious computations

$$\begin{cases} \mathbf{n}^\top \frac{\partial \mathbf{H}}{\partial x} = -\frac{1}{D} \frac{x}{\|\mathbf{x}\|^2} \\ \mathbf{n}^\top \frac{\partial \mathbf{H}}{\partial y} = -\frac{1}{D} \frac{y}{\|\mathbf{x}\|^2} \\ (\mathbf{H} \times \mathbf{n})^\top \boldsymbol{\omega} = \frac{2}{D} (y\omega_x, -x\omega_y, 0) \end{cases} \quad (18)$$

with $D = 2(2\|\mathbf{x}\|(1+\|\mathbf{x}\|))^{1/2}$ leading finally to

$$\begin{cases} \dot{u}_1 = -\frac{1}{D} \left(\frac{x}{\|\mathbf{x}\|^2} \dot{x} + \frac{y}{\|\mathbf{x}\|^2} \dot{y} - 2y\omega_x + 2x\omega_y \right) \\ \dot{u}_2 = 0 \end{cases} \quad (19)$$

that we can plug in (11) to obtain the revised optical flow constraint for a planar object parallel to the image plane when the light source direction coincides with the observer direction.

Before presenting some experimental results of target tracking, we first recall some fundamentals in visual servoing.

3. Visual servoing

Visual servoing techniques requires to design a set of visual features \mathbf{s} from the visual measurements $\mathbf{x}(\mathbf{r}_k)$ ($\mathbf{s} = \mathbf{s}(\mathbf{x}(\mathbf{r}_k))$ where \mathbf{r}_k is the camera pose at time k). A control law has to be designed also so that these features \mathbf{s} reach a desired value \mathbf{s}^* obtained for $\mathbf{r} = \mathbf{r}^*$, defining a correct realization of the task. The control principle is thus to regulate the error vector $\mathbf{s} - \mathbf{s}^*$ to zero. Such an approach needs the knowledge of the interaction matrix which describes the relationship between the relative camera-object velocity \mathbf{v} and the time variation of the selected visual features $\dot{\mathbf{s}}$

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v} + \frac{\partial \mathbf{s}}{\partial t} \quad (20)$$

with $\mathbf{v} = (\mathbf{v}, \boldsymbol{\omega})$ where $\mathbf{v} = (v_x, v_y, v_z)$ is the camera linear velocity and $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ its angular velocity, while $\partial \mathbf{s} / \partial t$ represents the time variation of \mathbf{s} due to potential motions of the object.

This matrix is usually used to design a control law whose aim is to cancel the vision-based task function [4]

$$\mathbf{e} = \widehat{\mathbf{L}}_{\mathbf{s}}^{\dagger} (\mathbf{s} - \mathbf{s}^*). \quad (21)$$

where $\widehat{\mathbf{L}}_{\mathbf{s}}$ is a model or an approximation of $\mathbf{L}_{\mathbf{s}}$. A classical control law is then given by

$$\mathbf{v} = -\lambda \mathbf{e} - \frac{\partial \widehat{\mathbf{e}}}{\partial t} \quad (22)$$

where λ is a positive gain. Various ways to estimate $\partial \widehat{\mathbf{e}} / \partial t$ exist. It is for example possible to use, assuming a constant target velocity, a simple integrator to vanish the steady state tracking error, leading at time k to

$$\left. \frac{\partial \widehat{\mathbf{e}}}{\partial t} \right|_k = \left. \frac{\partial \widehat{\mathbf{e}}}{\partial t} \right|_{k-1} + \mu \mathbf{e}_k = \mu \sum_{j=0}^{k-1} \mathbf{e}_j \quad \text{with} \quad \left. \frac{\partial \widehat{\mathbf{e}}}{\partial t} \right|_0 = 0 \quad (23)$$

with μ a positive gain that we have to tune.

3.1. Choice of the visual feature

In practice, the choice of the visual features \mathbf{s} is a key point of this approach. With a vision sensor providing 2D measurements $\mathbf{x}(\mathbf{r}_k)$, numerous potential visual features \mathbf{s} exist, since 2D data (coordinates of feature points in the image, moments, ...) as well as 3D data provided by a localization algorithm exploiting 2D data can be considered. Anyway, a robust extraction, matching (between $\mathbf{x}(\mathbf{r}_k)$ and $\mathbf{x}^* = \mathbf{x}(\mathbf{r}^*)$) and real-time spatio-temporal tracking (between $\mathbf{x}(\mathbf{r}_{k-1})$ and $\mathbf{x}(\mathbf{r}_k)$) have to be achieved. However, these tasks have proved to be difficult, as testified by the abundant literature on the subject.

In this paper we propose to radically modify the procedure by removing the extraction of geometric measurements and consequently the matching and tracking processes. To achieve this goal, we use as visual features the simplest feature that can be considered: the image intensity. The visual feature vector \mathbf{s} is nothing but the image while \mathbf{s}^* is the desired image. The error $\mathbf{s} - \mathbf{s}^*$ is then only the difference between the current and desired image, that is $\mathbf{I} - \mathbf{I}^*$ where \mathbf{I} is a vector containing image intensity of all pixels. Thus, we have:

$$\mathbf{s}(\mathbf{r}) = \mathbf{I}(\mathbf{r}) = (\mathbf{I}_{1\bullet}, \mathbf{I}_{2\bullet}, \dots, \mathbf{I}_{N\bullet}) \quad (24)$$

where $\mathbf{I}_{k\bullet}$ is the k -th line of the image. $\mathbf{I}(\mathbf{r})$ is then a vector of size $N \times M$ where $N \times M$ is the image size. Consequently, we have first to compute the interaction matrix related to the luminance I at a pixel \mathbf{x} in the image.

3.2. Interaction matrix related to the luminance in the general case

In fact, it is straightforward to compute it from our revised optical flow constraint. We simply have to express I_t from (11)

$$I_t = -\nabla I^{\top} \dot{\mathbf{x}} + kK_s u_1^{k-1} \dot{u}_1 + K_d \dot{u}_2, \quad (25)$$

writing from (12)

$$u_1 = \mathbf{M}_1 \mathbf{v} \quad \text{and} \quad u_2 = \mathbf{M}_2 \mathbf{v} \quad (26)$$

and introducing the interaction matrices \mathbf{L}_x and \mathbf{L}_y related to the coordinates x and y of \mathbf{x} [4]

$$\begin{aligned} \mathbf{L}_{\mathbf{x}} &= (\mathbf{L}_x, \mathbf{L}_y) \\ &= \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix} \end{aligned} \quad (27)$$

and we obtain the interaction matrix related to the intensity at pixel \mathbf{x} in the general case

$$\mathbf{L}_I = -\nabla I \mathbf{L}_{\mathbf{x}} + kK_s (\mathbf{n}^{\top} \mathbf{H})^{k-1} \mathbf{M}_1 + K_d \mathbf{M}_2. \quad (28)$$

Note that the interaction for a Lambertian scene is simply

$$\mathbf{L}_I = -\nabla I \mathbf{L}_{\mathbf{x}}. \quad (29)$$

However, as already stated, we cannot evaluate most of the components involved in the matrices \mathbf{M}_l (e.g., \mathbf{n} , \mathbf{L} , or even \mathbf{V}). We will therefore focus on the case when the light source is mounted on the camera as we did in section 2.1.

3.2.1 Light source on the camera

In that case, we can easily compute the matrices \mathbf{M}_l involved in (26) by explicitly use (27) in (16) and (17). However, it is easier to introduce in \mathbf{M}_l the planar \mathbf{M}_{lp} and non planar \mathbf{M}_{lnp} components such that $\mathbf{M}_l = \mathbf{M}_{lp} + \mathbf{M}_{lnp}$ where

$$\mathbf{M}_{1p} = \begin{pmatrix} -\frac{\mathbf{n}^{\top} \partial \mathbf{H}}{Z} \frac{\partial \mathbf{H}}{\partial x} \\ -\frac{\mathbf{n}^{\top} \partial \mathbf{H}}{Z} \frac{\partial \mathbf{H}}{\partial y} \\ \frac{x}{Z} \mathbf{n}^{\top} \frac{\partial \mathbf{H}}{\partial x} + \frac{y}{Z} \mathbf{n}^{\top} \frac{\partial \mathbf{H}}{\partial y} \\ xy \mathbf{n}^{\top} \frac{\partial \mathbf{H}}{\partial x} + (1+y^2) \mathbf{n}^{\top} \frac{\partial \mathbf{H}}{\partial y} + (\mathbf{H} \times \mathbf{n})^{\top} \mathbf{i} \\ -(1+x^2) \mathbf{n}^{\top} \frac{\partial \mathbf{H}}{\partial x} - xy \mathbf{n}^{\top} \frac{\partial \mathbf{H}}{\partial y} + (\mathbf{H} \times \mathbf{n})^{\top} \mathbf{j} \\ 0 \end{pmatrix}^{\top} \quad (30)$$

and

$$\mathbf{M}_{2p} = (0 \quad 0 \quad 0 \quad (\mathbf{n} \times \mathbf{k})^{\top} \mathbf{i} \quad (\mathbf{n} \times \mathbf{k})^{\top} \mathbf{j} \quad 0), \quad (31)$$

while we have for the non planar part

$$\mathbf{M}_{1np} = \begin{pmatrix} -\frac{1}{Z}\mathbf{H}^\top \frac{\partial \mathbf{n}}{\partial x} \\ -\frac{1}{Z}\mathbf{H}^\top \frac{\partial \mathbf{n}}{\partial y} \\ \frac{x}{Z}\mathbf{H}^\top \frac{\partial \mathbf{n}}{\partial x} + \frac{y}{Z}\mathbf{H}^\top \frac{\partial \mathbf{n}}{\partial y} \\ xy\mathbf{H}^\top \frac{\partial \mathbf{n}}{\partial x} + (1+y^2)\mathbf{H}^\top \frac{\partial \mathbf{n}}{\partial y} \\ -(1+x^2)\mathbf{H}^\top \frac{\partial \mathbf{n}}{\partial x} - xy\mathbf{H}^\top \frac{\partial \mathbf{n}}{\partial y} \\ y\mathbf{H}^\top \frac{\partial \mathbf{n}}{\partial x} - x\mathbf{H}^\top \frac{\partial \mathbf{n}}{\partial y} \end{pmatrix}^\top \quad (32)$$

and

$$\mathbf{M}_{2np} = \begin{pmatrix} \frac{1}{Z}\mathbf{k}^\top \frac{\partial \mathbf{n}}{\partial x} \\ \frac{1}{Z}\mathbf{k}^\top \frac{\partial \mathbf{n}}{\partial y} \\ -\frac{x}{Z}\mathbf{k}^\top \frac{\partial \mathbf{n}}{\partial x} - \frac{y}{Z}\mathbf{k}^\top \frac{\partial \mathbf{n}}{\partial y} \\ -xy\mathbf{k}^\top \frac{\partial \mathbf{n}}{\partial x} - (1+y^2)\mathbf{k}^\top \frac{\partial \mathbf{n}}{\partial y} \\ (1+x^2)\mathbf{k}^\top \frac{\partial \mathbf{n}}{\partial x} + xy\mathbf{k}^\top \frac{\partial \mathbf{n}}{\partial y} \\ -y\mathbf{k}^\top \frac{\partial \mathbf{n}}{\partial x} + x\mathbf{k}^\top \frac{\partial \mathbf{n}}{\partial y} \end{pmatrix}^\top \quad (33)$$

3.2.2 Light source on the camera and image and target planes parallel

For experimental issue, we are interested in the case where the light source is mounted on the camera with the same direction than the optical axis and where the image plane is parallel to a planar target. Here, we simply explicitly use (27) in (19) which gives

$$\mathbf{M}_{1p} = \frac{1}{D} \begin{bmatrix} \frac{1}{\|\mathbf{x}\|^2} \frac{x}{Z} & \frac{1}{\|\mathbf{x}\|^2} \frac{y}{Z} & \frac{-1}{\|\mathbf{x}\|^2} \frac{x^2 + y^2}{Z} & y & -x & 0 \end{bmatrix} \quad (34)$$

and $\mathbf{M}_{2p} = \mathbf{0}$.

4. Experimental results

Our goal is to perform, using the visual servoing scheme detailed in section 3, a tracking task with respect to a moving object. Therefore, as already stated, we have to maintain a rigid link between the target to track and the camera.

The camera is mounted on a 6 degrees of freedom gantry robot and a light-ring is located around the camera lens (see

Figure 4). Therefore the light direction is aligned with the camera optical axis as on Figure 3. This is the unique light in the scene. The considered object is planar (a photo), it is attached to a motorized rail that allows to control its motion. 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 interaction matrix has been computed using equation (28) with \mathbf{M}_1 given by (34) and $\mathbf{M}_2 = \mathbf{0}$. Object velocity is 1 cm/s. Control law is computed on a Core 2 Duo 3Gz PC running Linux. Image are acquired at 66Hz using an IEEE 1394 camera with a resolution of 320x240. The size of the vector \mathbf{s} is then 76800. Despite this size, the interaction matrix \mathbf{L}_I can be recomputed at each iteration.



Figure 4. Camera and light-ring mounted on the robot end-effector and view of the object to track and of the camera. The object (a photo) is attached to a motorized rail that allows to control its motion.

As can be seen on the images of Figures 6 and 5 (first rows), specularities can be seen in the image acquired by the camera. The first experiment shows the behavior of the control law when the interaction matrix was computed only under Lambertian hypothesis (given by equation (29)). Images on Figures 6 show that the tracking quickly failed leading to incoherent robot motion (the second row shows the error $\mathbf{I} - \mathbf{I}^*$; when the error is null the image must be completely gray).

The same experiment was now considered but with a full illumination model. As shown in Figure 5 the tracking is perfectly achieved since the error $\mathbf{I} - \mathbf{I}^*$ is almost null despite the occurrence of a specularity which shows the importance of such terms in the tracking process (we have $k = 100$ and $K_s = 200$). When the velocity is constant the object is perfectly tracked as can be seen on Figure 7c where $\|\mathbf{I} - \mathbf{I}^*\|$ is depicted. Error in the image remains small except when the object stops or accelerates (see the peaks in Figure 7c). The camera velocity (see Figure 7a) shows a pure motion along the x (± 1 cm/s) axis that corresponds to the motion of the motorized rail. For each pixel except during accelerations and decelerations, $|I - I^*| < 5$. Figure 7b depicts the camera position in the robot frame reference. It shows that despite only one degree of freedom of the object is controlled, a 6 dof motion of the robot has to be performed.

In the second experiment, we move the object by hand



Figure 5. First experiment: Tracking considering the complete interaction matrix that integrates specularly, diffuse and ambient terms.



Figure 6. First experiment: Tracking considering the interaction matrix under Lambertian hypothesis. As can be seen tracking failed quickly leading to incoherent robot motion. First row shows the image acquired by the camera while $\mathbf{I} - \mathbf{I}^*$ is shown on the second row.

as seen on the first row of Figure 9. The related images acquired by the camera are shown on the second row. We then have a more complex 3D object motion. Note that all the 6 dof of the robot are controlled (see Fig. 8b). The error $\|\mathbf{I} - \mathbf{I}^*\|$ is shown in Figure 8c while $\mathbf{I} - \mathbf{I}^*$ is shown on the third line of Figure 9. When the object is moving the error is more important than during the previous experiment. This is due to the fact that, since the object velocity is obviously no more constant (as testified by Figure 9a), it is no longer possible to consider the integral term involved in (23) leading to classical tracking errors. In contrast, when the motion stops the camera moves to reduce these errors (iterations 5300 and 9500).

5. Conclusion

In this paper we have extended the optical flow constraint equation in order to consider illumination model that removes the classical Lambertian hypothesis. We have also proposed a general framework to compute the variation of the image intensity due to camera and light motions and have provided the complete model when the light source is mounted on the camera. We then have applied this framework to a visual servoing context. We have also shown in

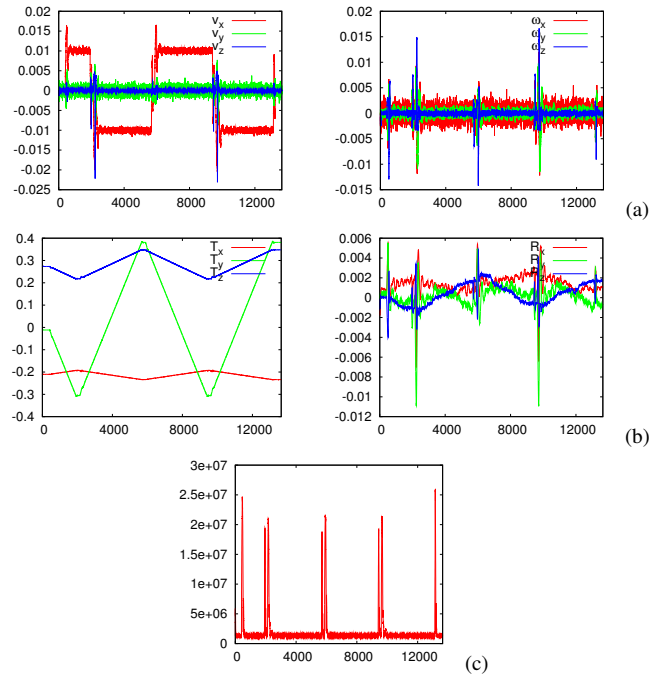


Figure 7. First experiment: Tracking considering the complete interaction matrix that integrates specularly, diffuse and ambient terms. (a) camera velocity (m/s and radian/s) (b) camera position in the robot reference frame (meter and radian) (c) error $\|\mathbf{I} - \mathbf{I}^*\|$

this paper that it is possible to use 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 image processing (except the image spatial gradient required for the computation of the interaction matrix) nor learning step. Experimental results of tracking planar objects have been presented which validate our approach.

References

- [1] M.J. Black, D.J. Fleet, and Y. Yacoob. Robustly estimating changes in image appearance. *Computer Vision and Image*

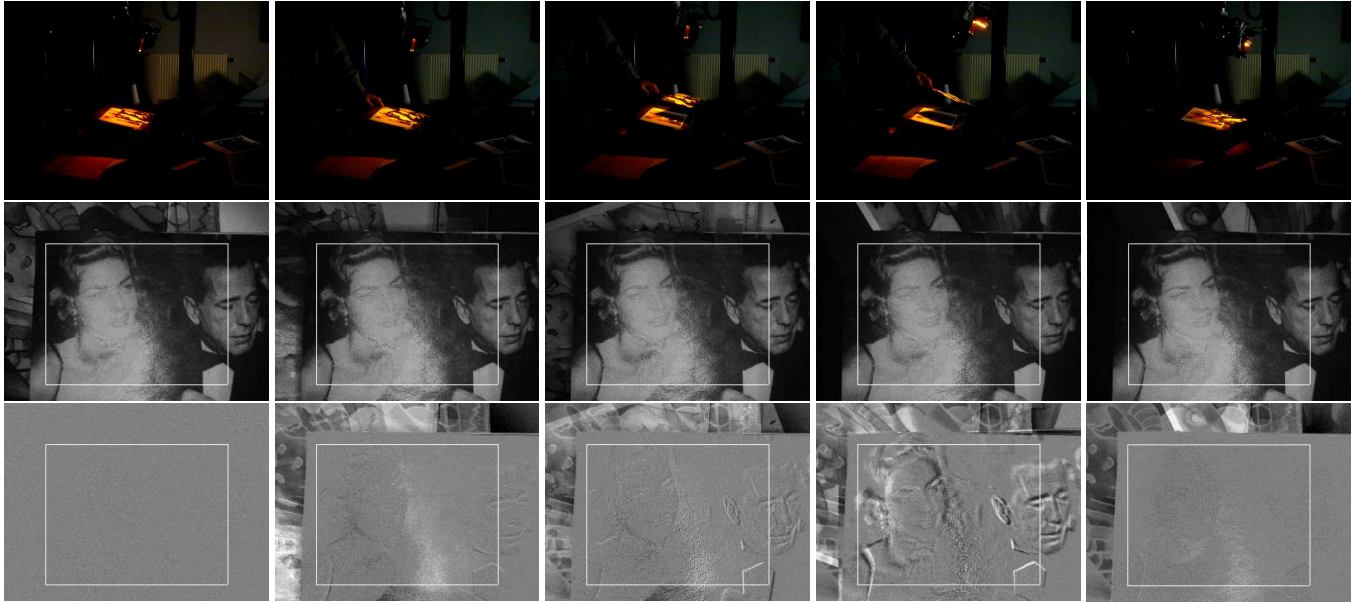


Figure 9. Second experiment: Tracking considering the complete interaction matrix that integrates specularly, diffuse and ambient terms.

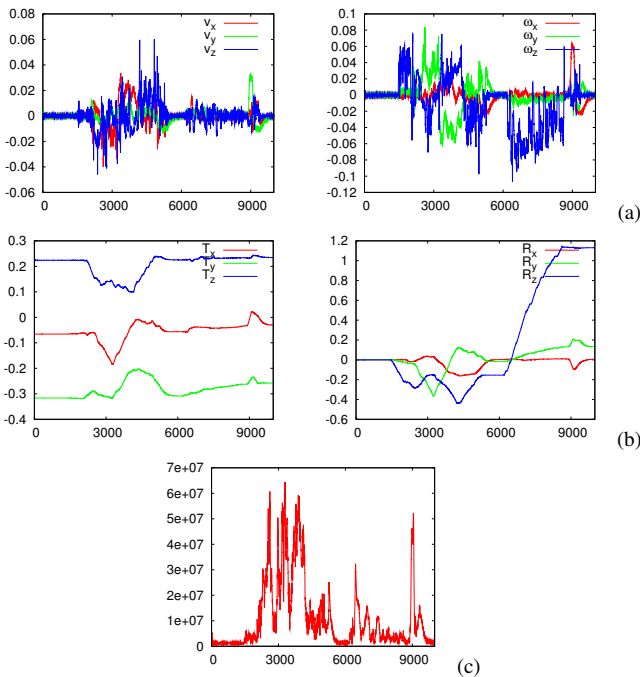


Figure 8. Second experiment: Tracking considering the complete interaction matrix that integrates specularly, diffuse and ambient terms. (a) camera velocity (m/s and radian/s) (b) camera position in the robot reference frame (meter and radian) (c) error $\| \mathbf{I} - \mathbf{I}^* \|$

Understanding, 78:8–31, 2000.

- [2] J. Blinn. Models of light reflection for computer synthesized pictures. In *ACM Conf. on Computer graphics and interactive techniques, SIGGRAPH'77*, pages 192–198, San Jose, California, 1977.
- [3] K. Deguchi. A direct interpretation of dynamic images with camera and object motions for vision guided robot control.

Int. Journal of Computer Vision, 37(1):7–20, June 2000.

- [4] 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.
- [5] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, October 1998.
- [6] H.W. Houssecker and D.J. Fleet. Computing optical flow with physical models of brightness variation. *IEEE Trans. on PAMI*, 23(6):661–673, June 2001.
- [7] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, August 1981.
- [8] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, October 1996.
- [9] S.K. Nayar, S.A. Nene, and H. Murase. Subspace methods for robot vision. *IEEE Trans. on Robotics*, 12(5):750–758, October 1996.
- [10] S. Negahdaripour. Revised definition of optical flow: Integration of radiometric and geometric cues for dynamic scene analysis. *IEEE Trans. on PAMI*, 20(9):961–979, 1998.
- [11] B.T. Phong. Illumination for computer generated pictures. *Communication of the ACM*, 18(6):311–317, June 1975.
- [12] G. Silveira and E. Malis. Real-time visual tracking under arbitrary illumination changes. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'07*, Minneapolis, USA, June 2007.
- [13] M. Tistarelli. Multiple constraints for optical flow. In *3rd European Conf. on Computer Vision, ECCV'94*, volume 1, pages 61–70, Stockholm, Sweden, May 1994.
- [14] A. Verri and T. Poggio. Motion field and optical flow: qualitative properties. *IEEE Trans. on PAMI*, 11(5):490–498, May 1989.