Visual Servoing with Photometric Gaussian Mixtures as Dense Feature

Nathan Crombez, El Mustapha Mouaddib, Guillaume Caron, François Chaumette

Visual Servoing with Photometric Gaussian Mixtures as Dense Feature

Nathan Crombez, El Mustapha Mouaddib, Guillaume Caron, François Chaumette

***Abstract*—The direct use of the entire photometric image information as dense feature for visual servoing brings several advantages. First, it does not require any feature detection, matching or tracking process. Thanks to the redundancy of visual information, the precision at convergence is really accurate. However, the corresponding highly nonlinear cost function reduces the convergence domain. In this paper, we propose a visual servoing based on the analytical formulation of Gaussian mixtures to enlarge the convergence domain. Pixels are represented by 2D Gaussian functions that denotes a "power of attraction". In addition to the control of the camera velocities during the servoing, we also optimize the Gaussian spreads allowing the camera to precisely converge to a desired pose even from a far initial one. Simulations show that our approach outperform the state of the art and real experiments show the effectiveness, robustness and accuracy of our approach.**

***Index Terms*—Visual servoing, Photometric Gaussian Mixture, large convergence domain**

I. Introduction

VISUAL servoing is a closed-loop control method for dynamic systems, such as robots, which uses visual information as feedback. Visual information is usually obtained by a digital camera directly set into motion by the system i.e. the eye-in-hand configuration. In another way, the camera may be stationary, external to the system in motion. Visual perception can be made with one or several cameras. In our case, we are particularly interested in eye-in-hand configuration with one camera. More precisely, the control law is based on visual information acquired by a camera in order to move a robot effector to a desired pose, implicitly defined by an image.

The visual information extracted from images are usually geometric features such as image points, straight lines, shapes and even 3D poses [2]. Exploiting these features requires complex image processing like their robust extraction, their matching and their real-time spatiotemporal tracking in the images acquired during the servoing.

To avoid these disadvantages, it has been proposed to directly use all image intensities as visual features (luminance feature) [3][4]. This approach known as photometric visual servoing only requires as image processing the computation of the image spatial gradient. Visual servoing based on the luminance feature has shown very accurate positioning even with approximated depths, partial occlusions, specular and low-textured environments. However, the highly nonlinear photometric cost function induces a relatively small convergence domain. Because of that, the visual difference between the initial image and the desired one should not be too large.

Several methods have been proposed to improve the pure photometric approach. In [5], the authors propose to adapt the desired image at each iteration of the visual servoing according to the illumination of the current image. The sum of conditional variance between the desired and the adapted current image is computed to achieve direct visual servoing. A robust M-estimator has been directly integrated to the method in order to reject the pixels of the current image which are too different from the desired one [6]. Another approach [7] compares the current and the desired images by computing their mutual information. The mutual information is a similarity measure that represents the quantity of visual information shared by two images. Using this similarity measure, the control law is naturally robust to partial occlusions and illumination variations. Another improvement is to represent the images by intensity histograms [8]. The use of these compact global descriptors increases robustness to noise and illumination changes. The control law minimizes the Matusita distance between the intensity histogram of the desired image and the histogram of the current image. However, even with all these improvements, the convergence domain remains relatively narrow.

Recently, photometric moments have been introduced as visual feature [9]. Pixel intensities of the whole image are considered to compute a set of photometric moments which represent essential characteristics of the image. The structure of the interaction matrix presents nice decoupling properties. This method has shown interesting results even for large displacement. However, the modeling of the interaction matrix is made under a constant image border hypothesis which leads to failures when parts of the scene enter or leave the camera field-of-view. A recent improvement has been proposed to counter this issue [10]. A spatial weighting function is included into the photometric moments formulation. However, this improvement alters the invariance properties of moments which disturbs the control of the camera rotation around the two axes orthogonal to the camera optical axis. Kernel-based Visual Servoing (KBVS) [11][12] also enables the independent control of the rotational and the translational motions. The KBVS approach proposes to use kernels as image operators. Gaussian kernels and spatial Fourier transform are used as visual features to control four degrees of freedom (dof) (the three translations and the rotation around the optical axis).

Part of this work has been presented at IROS'15 [1]

Nathan Crombez is with the LE2I laboratory, University of Technology Belfort-Montbliard, Beflort, France. e-mail:nathan.crombez@utbm.fr

El Mustapha Mouaddib and Guillaume Caron are with the MIS laboratory, University of Picardie Jules Verne, Amiens, France. e-mail: {mouaddib, guillaume.caron}@u-picardie.fr

François Chaumette is with Inria at Irisa Rennes, France. e-mail: Francois.Chaumette@inria.fr

As photometric moments, KBVS methods do not resolve the difficulties regarding the control of the rotations around the two orthogonal axes to the optical camera axis. They also encounter difficulties in dealing with appearance and disappearance of parts of the scene. Wavelets-based visual servoing has also been recently proposed [13]. The authors have developed a visual servoing control law using the decomposition of the images by the multiple resolution wavelet transform (MWT). Shearlet, a natural extension of wavelet, has also been introduced as visual feature for visual servoing [14]. The interaction matrix has been developed for shearlet transform coefficients but its computation is numerical and estimated offline. Similarly to wavelets, shearlets are very efficient in denoising and to represent anisotropic images features. Wavelets and shearlets have demonstrated interesting results in terms of robustness, stability and accuracy for six dof. However the convergence domain remains relatively tight for both methods.

In this paper, we propose a new approach that keeps the same advantages as the pure photometric method and considerably increases the convergence domain. For this purpose, each image pixel is represented as a Gaussian function w.r.t. its luminance information. This can be seen as an enlargement of the attraction power of each pixel. The combination of every pixel Gaussian function forms the Photometric Gaussian Mixture that represents the image.

Gaussian mixtures as visual servoing features have also been studied in [15]. In the latter work, image point features are modeled as a Gaussian mixture. Thanks to the representation of the geometric features as Gaussian distributions the points matching and tracking are not necessary. However, the points detection is still a crucial step and points extracted from the images during the servoing must be exactly the same as the points extracted from the desired image. In other words, the desired points should always be in the camera field-of-view and the points detector must have a 100% repeatability. Authors chose to set the same variance for each Gaussian function which induces ambiguities. Similarly, but in a different context, Gaussian mixtures have also been exploited in [16] for automatic 3D point clouds registration. Usually, 3D registration are realized in two steps: first, a coarse initialization using feature extraction and matching methods and then a refinement using Iterative Closest Point (ICP) based algorithms. The size of the region of convergence is significantly extended by the use of the Gaussian mixtures and the usual first step of initialization can be avoided. Moreover, this method handles local convergence problems of standard ICP algorithms.

The goal of our visual servoing is to minimize the difference between the desired Gaussian mixture and the Gaussian mixture computed from the current image varying over time. The interaction matrix is a key concept of visual servoing. It expresses the relationship between the image variations and the camera displacements. The photometric Gaussian mixtures of the images have an analytical formulation, thus enabling the determination of the analytical form of this interaction matrix. Our method does not need feature detection and has been extensively evaluated in experiments with a real camera on an actual industrial robot.

While we already published this idea in [1], two important contributions are proposed in this paper: a new model of photometric Gaussian has been designed (Section II-A) and a new modeling of the interaction matrix using the Green's theorem has been developed (Section III-A). Comparisons between the two photometric Gaussian models (Section II-C) and an evaluation of their robustness to noise (Section V-C) have been conducted. Both highlight the benefits of the new model. In addition, we present a strategy to initialize and to control the Gaussian extension (Section III-C). New simulations in complex virtual 3D scenes (Section V-B), comparisons with state-of-the-art approaches (Section V-D) and experimentations in real conditions (Section V-E) have also been carried out.

The remainder of this paper is organized as follows. First, Section II presents the formulation of Gaussian mixtures of 2D images. The influence of the Gaussian parameter is also studied. After that, Section III describes the development of the interaction matrix for the Photometric Gaussian Mixture feature. Section IV explains the control scheme to minimize our cost function. Then Section V shows simulation results, comparisons and real experiments. Finally, conclusions and future works are presented in Section VI.

II. Photometric Gaussian Mixture as visual feature

A digital image described in a 2D discrete space is derived from an analog signal in a 2D continuous space by a sampling process. This digitization can be modelled as the convolution of the continuous signal by a Dirac comb. Therefore, the obtained digital image can be seen as a comb of pulses whose pulses height depends on the pixels intensity. As mentioned above, considering that image pixels have a power of attraction, pixels represented as Dirac pulses (Fig. 1) have a zero attraction everywhere except to their own position. That is why, instead of considering a pixel of an image as a pulse we propose to represent it as a 2D Gaussian. Figure 1 illustrates, for an one-dimensional case, one of the advantages of this new signal representation: the difference between the desired and the current signal varies continuously, while it remains constant for Dirac pulses. This power of attraction enlarges the convergence domain of the visual servoing.

A. Photometric Gaussian Mixture

2D Gaussian is a suitable characterization of the power of attraction that we want to assign to pixels. Indeed, the further away from a pixel location the smaller the attraction of this pixel. A Gaussian function is defined on $[-\infty, +\infty]$, therefore a pixel has an influence on every other image pixel. Moreover, a Gaussian function is differentiable which is substantial to express analytically the interaction matrix (Section III).

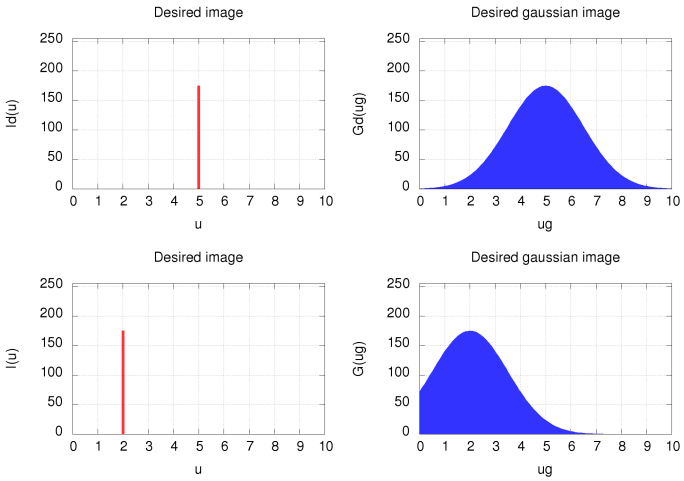


Fig. 1: Image pixels represented as pulses (red) and their associated Gaussian functions (blue)

Considering two uncorrelated variables $\overline{x} \in \mathbb{R}, \overline{y} \in \mathbb{R}$ which compose the vector $\overline{\mathbf{x}} = (\overline{x}, \overline{y})$, the two-dimensional Gaussian function is the distribution function given by:

$$f(x,y) = A \exp\left(-\left(\frac{(x-\overline{x})^2}{2\sigma_x^2} + \frac{(y-\overline{y})^2}{2\sigma_y^2}\right)\right) \quad (1)$$

where A is the amplitude coefficient, $\overline{\mathbf{x}}$ is the expected value and $\boldsymbol{\sigma} = (\sigma_x, \sigma_y)$ is the variance. In the following, we do not use this terminology because our use of the Gaussian is not statistical. To avoid ambiguities, we name $\overline{\mathbf{x}}$ as the Gaussian center and $\boldsymbol{\sigma}$ as the Gaussian extensions along $\vec{x}$ and $\vec{y}$ axes.

An image is composed by $N \times M$ pixels, each pixel has a location $\mathbf{u} = (u, v)$ and a luminance $I(\mathbf{u})$. We note $\mathbf{I}(\mathbf{r})$ an image acquired from a camera pose $\mathbf{r}$ represented by a vector $(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$. $\mathbf{I}(\mathbf{r})$ is the stacking of every $I(\mathbf{u})$ where $\mathbf{u}$ belongs to the discrete set of the acquired image coordinates grid $\mathbf{U}$.

a) Gaussian center $\overline{\mathbf{x}}$: we want the highest attraction of a pixel to be located at its own position. We therefore consider each image pixel $\mathbf{u}$ as a Gaussian function centred around its location. Thus, we have: $\overline{\mathbf{x}} = \mathbf{u}$.

b) Gaussian amplitude A: in order to keep pixel distinctiveness, under the temporal luminance constancy hypothesis, pixels must attract the other pixels that have the same intensity, Gaussians amplitude are thus related to pixels intensity: $A = I(\mathbf{u})$.

c) Gaussian extension $\boldsymbol{\sigma}$: considering the Gaussian extension as a power of attraction, there is no reason to give more priority to a pixel than another or to give more priority to an image axis than the others. Thus, every pixel has an equal extension λ_g along the $\vec{u}$ and $\vec{v}$ image axes: $\sigma_u = \sigma_v = \lambda_g$.

Following this parametrization, the photometric Gaussian function g is:

$$g(\mathbf{I}, \mathbf{u_g}, \mathbf{u}, \lambda_g) = I(\mathbf{u}) E(\mathbf{u_g} - \mathbf{u}) \quad (2)$$

where $\mathbf{u_g}$ are the Gaussian function coordinates and

$$E(\mathbf{u_g} - \mathbf{u}) = \exp\left(-\frac{(u_g - u)^2 + (v_g - v)^2}{2\lambda_g^2}\right), \quad (3)$$

for compactness.

A Gaussian mixture is defined as the sum combination of a finite number of Gaussian density functions. We denote the Photometric Gaussian Mixture associated with an image $\mathbf{I}$ computed with an extension parameter λ_g by $\mathbf{G}(\mathbf{I}, \lambda_g)$ (Fig. 2). More precisely, a spatial sampling of the Photometric Gaussian Mixture associated with the image $\mathbf{I}$ at $\mathbf{u_g}$ is expressed as:

$$G(\mathbf{I}, \mathbf{u_g}, \lambda_g) = \sum_{\mathbf{u} \in \mathbf{U}} g(\mathbf{I}, \mathbf{u_g}, \mathbf{u}, \lambda_g) = \sum_{\mathbf{u} \in \mathbf{U}} I(\mathbf{u}) E(\mathbf{u_g} - \mathbf{u}) \quad (4)$$

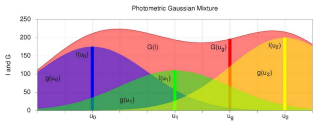


Fig. 2: A Photometric Gaussian Mixture $\mathbf{G}(\mathbf{I})$ (red) of a one-dimensional image composed by 3 pixels $I(u_0)$, $I(u_1)$ and $I(u_2)$ (respectively the blue, green and yellow pulses). $\mathbf{G}(\mathbf{I})$ is the sum combination of every Gaussian function related to image pixels.

Note that another interpretation of $G(\mathbf{I}, \mathbf{u_g}, \lambda_g)$ consists to consider the intensity of every pixel $\mathbf{u}$ around $\mathbf{u_g}$ with a weight proportional to the distance between $\mathbf{u}$ and $\mathbf{u_g}$. $\mathbf{G}$ is the stacking of every G. Fig. 3 shows several Gaussian mixtures $\mathbf{G}(\mathbf{I}, \lambda_g)$ of a same image $\mathbf{I}$ for different extension parameters λ_g.

B. Role of the Gaussian extension

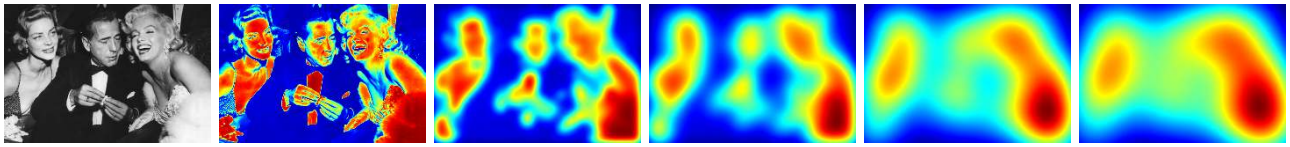
Visual servoing based on the pure photometric feature (pixels luminance of the entire images) is designed to minimize the difference between a desired image $\mathbf{I}(\mathbf{r}^*)$ and images acquired during the servoing $\mathbf{I}(\mathbf{r})$. Vectors $\mathbf{r}^*$ and $\mathbf{r}$ respectively represent the desired and the current camera poses. Because of the limited convergence domain, the displacement between $\mathbf{r}^*$ and $\mathbf{r}$ must be short enough to guarantee a large photometric overlapping between the initial image and the desired one.

With a large extension parameter λ_g, the power of attraction of every pixel is more important. Thus, the Gaussian mixture representation (e.g. Fig. 3f) offers more chance to have overlapping areas between the representation of the desired image and the representation of the current image. On the contrary, it is interesting to note that for a small extension parameter, the Gaussian mixture (e.g. Fig. 3b) and the original image (Fig. 3a) are similar. This can be verified by:

$$\lim_{\lambda_g \longrightarrow 0} G(\mathbf{I}, \mathbf{u_g}, \lambda_g) = I(\mathbf{u})|_{\mathbf{u}=\mathbf{u_g}} \quad (5)$$

and consequently,

$$\lim_{\lambda_g \longrightarrow 0} \mathbf{G}(\mathbf{I}, \lambda_g) = \mathbf{I}. \quad (6)$$



(a) (b) (c) (d) (e) (f)

Fig. 3: Influence of λ_g: (a) Grayscale image $\mathbf{I}$, (b-f) Representations of the Gaussian mixture $\mathbf{G}(\mathbf{I}, \lambda_g)$ for respectively $\lambda_g = \{0.1, 5.0, 10.0, 20.0, 25.0\}$

The comparison of cost functions shape obtained for different λ_g confirms these observations. Let us consider that Fig. 3a shows the desired image. Fig. 4a shows the cost function obtained by computing the differences between the desired image $\mathbf{I}(\mathbf{r}^*)$ and images $\mathbf{I}(\mathbf{r})$ obtained around the desired pose w.r.t. translations along $\vec{u}$ and $\vec{v}$ image axes. Fig. 4(b-f) show cost functions obtained by computing the differences between the desired Gaussian mixture $\mathbf{G}(\mathbf{I}(\mathbf{r}^*), \lambda_g)$ and the Gaussian mixture $\mathbf{G}(\mathbf{I}(\mathbf{r}), \lambda_g)$ for different extension parameter λ_g values. We can see that the higher λ_g is, the larger the convex domain. On the contrary, when λ_g is low, the cost function (Fig. 4b) has the same tight shape than the photometric one (Fig. 4a). It appears thus appropriate to start the visual servoing with a high λ_g to ensure the convergence and then decrease it for obtaining a high accuracy. The control of this supplementary degree of freedom simultaneously to the classical ones (i.e. the six dof of the camera) is discussed in Section III-C.

C. Photometric Gaussian models comparison

In [1], we had proposed the following photometric Gaussian expression (Model 1):

$$g(\mathbf{I}, \mathbf{u_g}, \mathbf{u}, \lambda_g) = \exp\left(-\frac{(u_g - u)^2 + (v_g - v_i)^2}{2I^2(\mathbf{u})\lambda_g^2}\right) \quad (7)$$

where the extension parameter is related to the pixel intensity: $\sigma_u = \sigma_v = \lambda_g I(\mathbf{u})$ and where every pixel has the same Gaussian amplitude: $A = 1$.

Due to the direct relation between the extension parameter and the pixels intensity, Model 1 is less robust to image noise than the new proposed model. Indeed, noise may generate low and high intensity pixels distributed over the images that affect the shape of the Gaussian mixture. To highlight that problem, Gaussian mixtures have been computed from one-dimensional images for ease of reading. The first row of Fig. 5 shows three one-dimensional images. The first one is noiseless, and noisy pulses that follow normal distributions with different variances have been added on the two others. The second and the third rows show the Gaussian mixtures computed using respectively Model 1 [1] and the new model presented in this paper (Section II-A). Se can see that a small noise has a significant influence on the shape of the Gaussian mixture when the Model 1 is considered. The noisy pixels produce narrow Gaussians and the resulting Gaussian mixtures are less convex and do not provide the expected representation of the initial images. On the contrary, the new model (2) always produces a smooth Gaussian mixture.

These dissimilarities between the two photometric Gaussian models induce different photometric Gaussian mixtures and thus different visual servoing behaviors. Experimental results (Section V-C) validate these observations and illustrate the higher performances of the new model.

III. Modeling of the Interaction Matrix

The key of visual servoing is the interaction matrix $\mathbf{L_s}$ which links the time variation of visual features $\mathbf{s}$ to the camera velocity $\mathbf{v} = (\boldsymbol{\upsilon}, \boldsymbol{\omega})$ where $\boldsymbol{\upsilon}$ and $\boldsymbol{\omega}$ are respectively the linear and angular components [2]:

$$\dot{\mathbf{s}} = \mathbf{L_s}\mathbf{v} \quad (8)$$

In order to use Photometric Gaussian Mixtures as visual features, we have to develop the interaction matrix for $\mathbf{s} = \mathbf{G}$ (Eq. 4). The Photometric Gaussian Mixture based Visual Servoing objective is to regulate to 0 the following error:

$$\boldsymbol{\epsilon} = \mathbf{G}(\mathbf{I}(\mathbf{r}), \lambda_g) - \mathbf{G}(\mathbf{I}(\mathbf{r}^*), {\lambda_g}^*) = \mathbf{G} - \mathbf{G}^* \quad (9)$$

where $\mathbf{G}(\mathbf{I}(\mathbf{r}^*), {\lambda_g}^*)$ is the Gaussian mixture associated with the desired image computed with a fixed extension parameter ${\lambda_g}^*$ and $\mathbf{G}(\mathbf{I}(\mathbf{r}), \lambda_g)$ is the Gaussian mixture associated with the image acquired during the servoing computed with the extension parameter λ_g.

We present in this paper two methods to perform the modeling of $\mathbf{L_G}$. As in [9], the first approach is based on the Green's theorem which permits to avoid the computation of image gradients. The second is based on the assumption that the 3D scene itself is a Gaussian mixture viewed by a camera. Comparison results (Section V-A) show that this assumption offers good approximations of the interaction matrix.

A. Method #1: Green's Theorem Modeling (GTM)

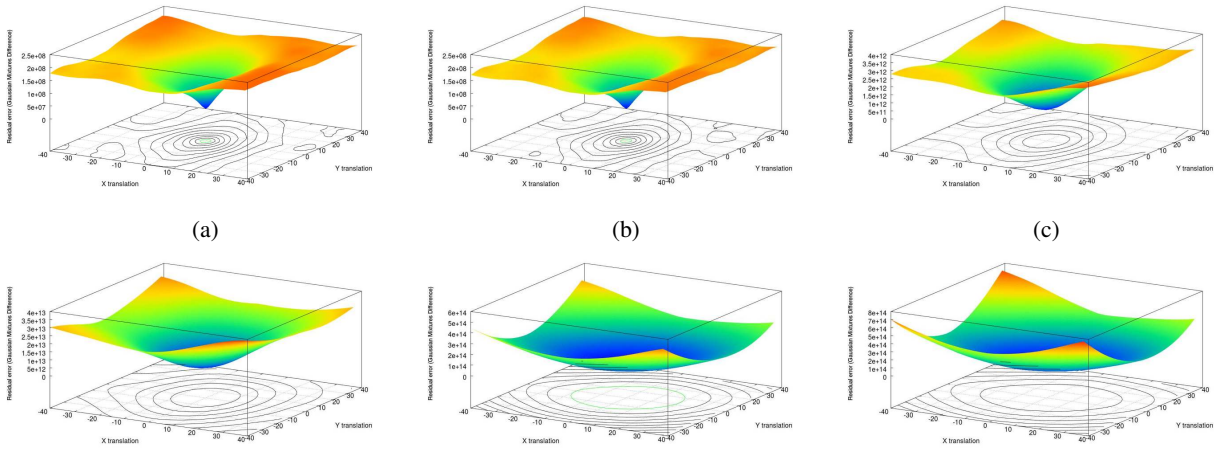
In (4), the only term that is time dependent is the image intensity $I(\mathbf{u})$ that varies due to the camera displacements. Consequently:

$$\dot{G}(\mathbf{u_g}) = \sum_{\mathbf{u}} \left(\dot{I}(\mathbf{u}) E(\mathbf{u_g} - \mathbf{u})\right), \quad (10)$$

Considering, as usual, that the temporal luminance constancy hypothesis is ensured [3][9][10], which means the optical flow constraint (OFC) [17] is valid, we have [3]:

$$\dot{I}(\mathbf{u}) = -\frac{\partial I(\mathbf{u})}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{r}} \dot{\mathbf{r}} = -\nabla \mathbf{I}^\top \mathbf{L_u} \mathbf{v} \quad (11)$$

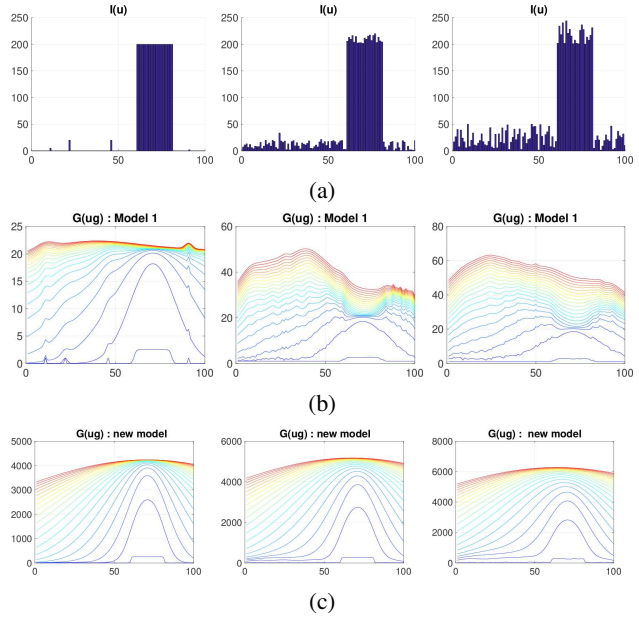
where $\mathbf{L_u}$ is the interaction matrix related to an image pixel point. Knowing the intrinsic parameters of the perspective



(a) (b) (c)

(d) (e) (f)

Fig. 4: Cost functions shapes comparison: (a) Photometric dense features, (b-f) Gaussian mixtures for respectively $\lambda_g = \{0.1, 5.0, 10.0, 20.0, 25.0\}$.



(a) (b) (c)

Fig. 5: Comparison of Gaussian mixtures computed using two different photometric Gaussian models. (a) One-dimensional images, (b-c) Gaussian mixtures computed respectively with Model 1 and the new model. The color variation indicates 20 values of λ_g (from 0.01 to 1 for Model 1 and from 0.5 to 100 for the new model).

projection model: α_u, α_v for the horizontal and vertical scale factors of the camera photosensitive matrix and u_0, v_0 for the principal point coordinates, $\mathbf{L_u}$ can be decomposed as:

$$\begin{aligned}\mathbf{L_u} &= \mathbf{L_{ux}}\mathbf{L_x} \\ &= \begin{bmatrix}\alpha_u & 0 \\ 0 & \alpha_v\end{bmatrix}\begin{bmatrix}\frac{-1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & \frac{-1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x\end{bmatrix}\end{aligned} \tag{12}$$

where $\mathbf{L_x}$ is the so called interaction matrix related to a point $\mathbf{x} = (x, y)$, expressed in the normalized image plane of the perspective projection model [2].

Injecting (11) in (10), we obtain:

$$\dot{G}(\mathbf{u_g}) = -\sum_{\mathbf{u}} \left(\nabla \mathbf{I}^\mathsf{T} E(\mathbf{u_g} - \mathbf{u})\mathbf{L_u}\right)\mathbf{v} \tag{13}$$

from which we deduce by analogy with (8):

$$\begin{aligned}\mathbf{L}_G &= -\sum_{\mathbf{u}} \left(\nabla \mathbf{I}^\mathsf{T} E(\mathbf{u_g} - \mathbf{u})\mathbf{L_u}\right) \\ &= [L_{G_{vx}}\ L_{G_{vy}}\ L_{G_{vz}}\ L_{G_{wx}}\ L_{G_{wy}}\ L_{G_{wz}}]\end{aligned} \tag{14}$$

Each component of the interaction matrix $\mathbf{L}_G$ involves image gradients $\nabla \mathbf{I} = [\frac{\delta I}{\delta u}, \frac{\delta I}{\delta v}]^\mathsf{T}$. With regards to the pure photometric feature [2], these image gradients are an approximation of the actual image derivatives computed using image processing methods (derivative filters) along horizontal and vertical image axes. Concerning the photometric moments [9], authors proposed to avoid the image gradients computation using the Green's theorem. This simplification step can also be employed for the Photometric Gaussian Mixture feature.

For example, let us consider $L_{G_{vx}}$ the component of $\mathbf{L}_G$ corresponding to the camera translational velocity in x. We suppose that the scene is planar and that the depth of all the points is equal to a constant value Z[1]:

$$\begin{aligned}L_{G_{vx}} &= -\sum_{\mathbf{u}} \left(\begin{bmatrix}\frac{\partial I}{\partial u}E \\ \frac{\partial I}{\partial v}E\end{bmatrix}\begin{bmatrix}\alpha_u & 0 \\ 0 & \alpha_v\end{bmatrix}\begin{bmatrix}-\frac{1}{Z} \\ 0\end{bmatrix}\right) \\ &= \frac{\alpha_u}{Z}\sum_{\mathbf{u}}\left(\frac{\partial I}{\partial u}E\right)\end{aligned} \tag{15}$$

We note $Q = EI$ and $P = 0$, then:

$$\frac{\partial Q}{\partial u} = \frac{\partial E}{\partial u}I + E\frac{\partial I}{\partial u} \quad , \quad \frac{\partial P}{\partial v} = 0 \tag{16}$$

The Green's theorem gives us:

$$\sum_{\mathbf{u}}\left(\frac{\partial Q}{\partial u} - \frac{\partial P}{\partial v}\right) = \sum_{\partial u} P + \sum_{\partial v} Q, \tag{17}$$

[1]One can note that, as for the photometric moments [9], it is also possible to model the interaction matrix considering that $\frac{1}{Z} = Ax + By + C$. This part is left for future work.

and given that $\frac{\partial P}{\partial v} = 0$ from equation (16):

$$\sum_{\mathbf{u}} \frac{\partial Q}{\partial u} = \sum_{\partial v} Q \tag{18}$$

Substituting (16) in (18), we obtain:

$$\sum_{\mathbf{u}} \frac{\partial E}{\partial u} I + \sum_{\mathbf{u}} E \frac{\partial I}{\partial u} = \sum_{\partial v} EI \tag{19}$$

Therefore, we have:

$$\sum_{\mathbf{u}} \frac{\partial I}{\partial u} E = \sum_{\partial v} EI - \sum_{\mathbf{u}} \frac{\partial E}{\partial u} I \tag{20}$$

As in [9] we consider being under the zero border assumption. Indeed if the pixel intensities $I(\mathbf{u})$ lying on the border of the image are all zero, the term $\sum_{\partial v} EI$ in (20) is equal to zero. Then, replacing (20) in (15), we finally get:

$$L_{G_{vx}} = -\frac{\alpha_u}{Z} \sum_{\mathbf{u}} \left(I \frac{\partial E}{\partial u} \right) \tag{21}$$

where $\frac{\partial E}{\partial u}$ is easy to express from (3):

$$\nabla \mathbf{E}^{\mathsf{T}} = \begin{bmatrix} \frac{\delta E}{\delta u} \\ \frac{\delta E}{\delta v} \end{bmatrix} = \begin{bmatrix} \frac{(u_g - u)}{\lambda_g^2} E \\ \frac{(v_g - v)}{\lambda_g^2} E \end{bmatrix} \tag{22}$$

The interaction matrix of each component can be developed following the same way. We obtain:

$$\begin{aligned}
L_{G_{vx}} =& -\frac{\alpha_u}{Z} \sum_{\mathbf{u}} \frac{\partial E}{\partial u} I \\
L_{G_{vy}} =& -\frac{\alpha_v}{Z} \sum_{\mathbf{u}} \frac{\partial E}{\partial v} I \\
L_{G_{vz}} =& \frac{1}{Z} \left(\alpha_u \sum_{\mathbf{u}} \frac{\partial E}{\partial u} Ix + \alpha_v \sum_{\mathbf{u}} \frac{\partial E}{\partial v} Iy + 2 \sum_{\mathbf{u}} EI \right) \\
L_{G_{wx}} =& \alpha_u \sum_{\mathbf{u}} \frac{\partial E}{\partial u} Ixy + \alpha_v \sum_{\mathbf{u}} I \frac{\partial E}{\partial v} (1 + y^2) + 3 \sum_{\mathbf{u}} EIy \\
L_{G_{wy}} =& -\alpha_u \sum_{\mathbf{u}} \frac{\partial E}{\partial u} I(1 + x^2) - \alpha_v \sum_{\mathbf{u}} \frac{\partial E}{\partial v} Ixy - 3 \sum_{\mathbf{u}} EIx \\
L_{G_{wz}} =& \alpha_u \sum_{\mathbf{u}} \frac{\partial E}{\partial u} Iy - \alpha_v \sum_{\mathbf{u}} \frac{\partial E}{\partial v} Ix
\end{aligned} \tag{23}$$

It is important to note that the image gradients are not needed anymore to compute the interaction matrix.
We propose in the next subsection a second method to model the interaction matrix that is easier and faster to compute.

B. Method #2: "Photometric Gaussian Consistency" (PGC)

Photometric Gaussian Mixtures are representations of images - images which are the results of the projection of a 3D scene on a 2D plane. To model the interaction matrix, we consider here that the Gaussian mixtures are direct representations of the 3D scene itself. In other words, we consider that the scene is already a Gaussian mixture. This assumption can be seen as an approximation of the projection of the 3D Gaussians on the 2D Gaussian mixtures. Thus, the temporal luminance consistency is adapted to the temporal photometric Gaussian consistency:

$$g(\mathbf{u_g} + \Delta \mathbf{u_g}, t + \Delta t) = g(\mathbf{u_g}, t), \tag{24}$$

where $\mathbf{u}$, $\mathbf{I}$ and λ_g are omitted in function g (25) for compactness. A first order Taylor development of (24) gives:

$$\frac{\partial g}{\partial \mathbf{u_g}} \dot{\mathbf{u}}_\mathbf{g} + \frac{\partial g}{\partial t} = \nabla \mathbf{g}^{\mathsf{T}} \dot{\mathbf{u}}_\mathbf{g} + \dot{g} = 0 \tag{25}$$

with $\nabla \mathbf{g}^{\mathsf{T}}$ the spatial gradient of $g(\mathbf{u_g}, t)$ and $\dot{g}$ its time derivation, thus leading to a relationship similar to the optical flow constraint (11).

Assuming this temporal photometric Gaussian consistency and following a similar reasoning, (4) is now decomposed as:

$$\begin{aligned}
\dot{G} &= \sum_{\mathbf{u}} \dot{g} = -\sum_{\mathbf{u}} \nabla \mathbf{g}^{\mathsf{T}} \dot{\mathbf{u}}_\mathbf{g} \\
&= -\sum_{\mathbf{u}} \left(\frac{\partial (\mathbf{I}(\mathbf{u}) E(\mathbf{u_g} - \mathbf{u}))}{\partial \mathbf{u_g}} \right) \dot{\mathbf{u}}_\mathbf{g} \\
&= -\sum_{\mathbf{u}} \left(\mathbf{I}(\mathbf{u}) \frac{\partial E(\mathbf{u_g} - \mathbf{u})}{\partial \mathbf{u_g}} \right) \mathbf{L}_{\mathbf{u_g}} \mathbf{v}
\end{aligned} \tag{26}$$

Here, $\mathbf{L}_{\mathbf{u_g}}$ is out of the summation because it does not depend on $\mathbf{u}$ while, in the Green's theorem based modeling, $\mathbf{L}_\mathbf{u}$ is inside the summation (14). The computation is consequently faster with (26). As in the previous modeling (Section III-A), the expression of each component of the interaction matrix $\mathbf{L}_G$ does not contain image gradients but Gaussian derivatives $\nabla \mathbf{E_g}$:

$$\nabla \mathbf{E_g}^{\mathsf{T}} = \begin{bmatrix} \frac{\delta E}{\delta u_g} \\ \frac{\delta E}{\delta v_g} \end{bmatrix} = \begin{bmatrix} -\frac{(u_g - u)}{\lambda_g^2} E \\ -\frac{(v_g - v)}{\lambda_g^2} E \end{bmatrix} = -\nabla \mathbf{E}^{\mathsf{T}} \tag{27}$$

Then, an analytic formulation of the interaction matrix of each component can be developed:

$$\begin{aligned}
L_{G_{vx}} =& \alpha_u K_u / Z \\
L_{G_{vy}} =& \alpha_v K_v / Z \\
L_{G_{vz}} =& -\left(\alpha_u K_u x_g + \alpha_v K_v y_g \right) / Z \\
L_{G_{wx}} =& -\alpha_u K_u x_g y_g - \alpha_v K_v (1 + y_g^2) \\
L_{G_{wy}} =& \alpha_u K_u (1 + x_g^2) + \alpha_v K_v (x_g y_g) \\
L_{G_{wz}} =& -\alpha_u K_u y_g + \alpha_v K_v x_g
\end{aligned} \tag{28}$$

where $K_u = \sum_\mathbf{u} I \frac{\partial E}{\partial u_g}$ and $K_v = \sum_\mathbf{u} I \frac{\partial E}{\partial v_g}$.

It is interesting to note that even if the two modeling approaches are different, the expressions of the interaction matrix are close (see (23)). More precisely, the components $L_{G_{vx}}$ and $L_{G_{vy}}$ are strictly identical since $\nabla \mathbf{E}^{\mathsf{T}} = -\nabla \mathbf{E_g}^{\mathsf{T}}$ while the other ones are more simple. Furthermore K_u and K_v are computed once for every $\mathbf{u_g}$ while in the Green's method a lot of terms have to be computed.

C. Extension parameter modeling

The influences of the extension parameter observed in Section II-B show that it is interesting to start the servoing using a high extension parameter and to finish it with a small one. Indeed, this respectively ensures to enlarge the

convergence domain while keeping the convergence accuracy at least similar to the pure photometric case. The extension parameter λ_g is optimized during the visual servoing as well as the camera velocities $\mathbf{v}$. To this end, we compute the derivative of the Photometric Gaussian Mixture with respect to the parameter λ_g:

$$\nabla\Lambda_{G(\mathbf{u_g})} = \sum_{\mathbf{u}} \frac{\partial g(\mathbf{I}, \mathbf{u_g}, \mathbf{u}, \lambda_g)}{\partial \lambda_g} \tag{29}$$

$$= \sum_{\mathbf{u}} \left(I(\mathbf{u}) \frac{\partial E(\mathbf{u_g}, \mathbf{u})}{\partial \lambda_g} \right) \tag{30}$$

$$= \sum_{\mathbf{u}} \left(I(\mathbf{u}) \frac{(\mathbf{u_g} - \mathbf{u})^2}{\lambda_g^3} \exp\left(-\frac{(\mathbf{u_g} - \mathbf{u})^2}{2\lambda_g^2} \right) \right) \tag{31}$$

IV. Implementation and validation

We extend the classical control law [2] to:

$$\mathbf{v}_\lambda = -\mu \widehat{\mathbf{L}}^+_{\mathbf{G}_\lambda} (\mathbf{G} - \mathbf{G}^*) \tag{32}$$

where $\mathbf{v}_\lambda = (\boldsymbol{\upsilon},\ \boldsymbol{\omega},\ \delta\lambda_g)$ is composed of the three linear and three angular camera velocities, and the extension parameter increment. The convergence can be improved by tuning the gain μ. A Gauss-Newton scheme is used here to validate the use of the Photometric Gaussian Mixture feature. Other optimization algorithms like the Efficient Second order Minimization [18] or the Levenberg-Marquardt [19] could be used. $\widehat{\mathbf{L}}^+_{\mathbf{G}_\lambda}$ is the pseudo inverse of the extended interaction matrix $\mathbf{L}_{\mathbf{G}_\lambda}$ related to the Gaussian mixture $\mathbf{G}$. The stacking of every $\mathbf{L}_G$ interaction matrices (14) related to Gaussian mixture values at every location $\mathbf{u_g}$ and the equation (31), gives the global interaction matrix:

$$\mathbf{L}_{\mathbf{G}_\lambda} = \begin{bmatrix} & \vdots & \\ \mathbf{L}_{G(\mathbf{u_g})} & & \boldsymbol{\Lambda}_{G(\mathbf{u_g})} \\ & \vdots & \end{bmatrix}, \tag{33}$$

where $\mathbf{L}_{G(\mathbf{u_g})}$ is computed with (23) or (28), depending on the chosen model, and $\boldsymbol{\Lambda}_{G(\mathbf{u_g})}$ from (31). For both methods, the computation of $\mathbf{L}_{\mathbf{G}_\lambda}$ requires the scene depth Z from the camera (12). In this paper, we consider the depth of each point as constant and equal to the distance between the scene and the camera at the desired pose ($\widehat{Z} = Z^*$). That is why the pseudo-inverse of the interaction matrix is noted $\widehat{\mathbf{L}}^+_{\mathbf{G}_\lambda}$. This control law is locally asymptotically stable when $\widehat{\mathbf{L}}^+_{\mathbf{G}_\lambda} \mathbf{L}_{\mathbf{G}_\lambda} > 0$.

The choice of initial and desired λ_g must ensure a large convergence and a high accuracy. These two aspects are respectively satisfied by initializing λ_g with a large value and by ending the servoing with a very small value. We propose a two-steps extension parameter evolution strategy:

- Step 1 to enlarge the convergence domain: we initialize λ_g with a large value λ_{gi} and the desired extension parameter is initialized following ${\lambda_g}^* = \lambda_{gi}/2$.
- Step 2 to accurately finish the visual servoing: both λ_g and ${\lambda_g}^*$ are set with a same small value.

The transition from the first to the second step is performed when $|\lambda_g - {\lambda_g}^*| < S$ where S is an empirically fixed value. More details about the choice of the latter parameters values are provided in Section V-F.

V. Experimental results

The two proposed interaction matrix modeling methods, i.e. the Green Theorem Modeling (GTM) and the Photometric Gaussian Consistency (PGC), are compared in Section V-A. Then, challenging experiments that highlight the contributions of using Photometric Gaussian Mixtures as visual features are presented in Section V-B. Section V-C studies the robustness of the proposed method when some noise is added to the images. In Section V-D, our method is opposed with three state-of-the-art approaches. Section V-E presents experiments conducted with a real robot in 3D environments. Finally, the initialization of the required parameters are discussed in Section V-F.

A. Comparisons between GTM and PGC

Experiment #V1 (Fig. 6): The first simulated experiment demonstrates the concept of the proposed approaches for 2 dof (v_x and v_y). A single untextured object is present in the camera field-of-view. At the desired camera pose, the object is projected on the top right corner of the image (Fig. 6d). At the initial camera pose, the object is projected on the bottom left corner of the acquired image (Fig. 6b). The projections of the object on the desired and initial images have not any photometric overlapping area. The initial extension parameter has been chosen such that λ_g is high enough to induce a sufficient overlapping area between the two Gaussian mixtures. In the first step of the extension parameter evolution strategy, the desired Gaussian mixture is computed using a fixed ${\lambda_g}^* = 0.5 * \lambda_{gi}$ where $\lambda_{gi} = 90.0$. The transition to the second step is performed when $|\lambda_{gi} - {\lambda_g}^*| < 0.1$. Both extension parameters ${\lambda_g}^*$ and λ_g are set to 1.0 (as explained previously in Section IV).

As seen on Fig. 6, the two modeling methods (GTM and PGC) give strictly the same results because the components $L_{G_{vx}}$ and $L_{G_{vy}}$ are strictly identical. In the beginning, the error does not decrease exponentially because of the large error between the desired and the initial images. But after a few iterations, it decreases exponentially (Fig. 6h and Fig. 6l). Despite the large displacement between the desired and the initial poses, the camera has converged to the desired one as well as the Gaussian expansion parameter (Fig. 6i and 6m). This example shows how the Photometric Gaussian Mixture drastically enlarges the convergence domain of the photometric visual servoing while maintaining the final accuracy. In this situation, the pure photometric feature [3] could not at all drives the camera to the desired pose.

Experiment #V2 (Fig. 7): This experiment has been conducted using a complex scene (textured plane) containing several shapes and controlling six dof. Note that some parts of the scene which are present in the desired image (Fig. 7a) are not visible in the initial one (Fig. 7b) and vice versa. The image and its associated Gaussian mixture at the end of the servoing (Fig. 7d) illustrate that the camera has reached its desired pose. However, due to the visual appearances and disappearances of

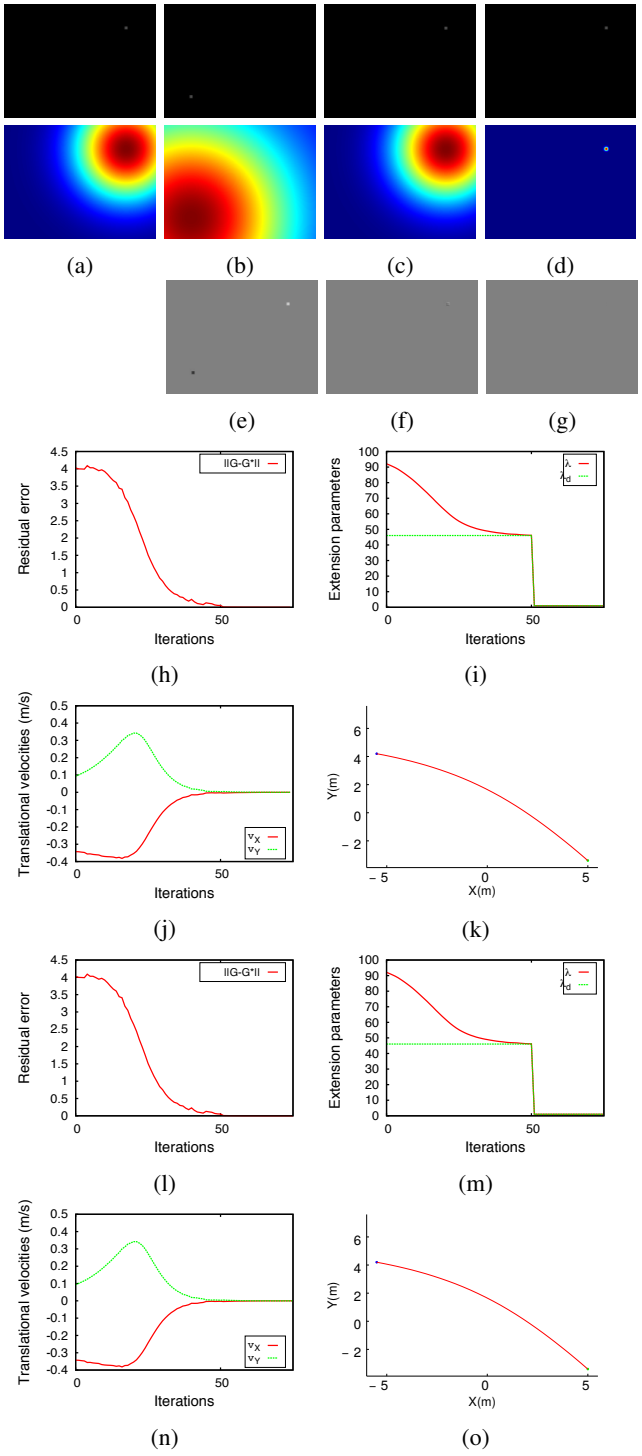


Fig. 6: Experiment #V1: Comparison between GTM (from h to k) and PGC (from l to o). (a) Desired image and its Gaussian mixture. (b) Initial image and its Gaussian mixture, (c) Image and its Gaussian mixture just before switching, (d) Final image and its Gaussian mixture. (e)-(g) Image of difference, (h) and (l) Residual error, (i) and (m) Extension parameters, (j) and (n) Velocities, (k) and (o) Trajectories.

parts of the scene during the servoing, the residual error does not follow a perfect exponential decrease (Fig. 7h and Fig. 7l). The Gaussian expansion parameter has converged to the value

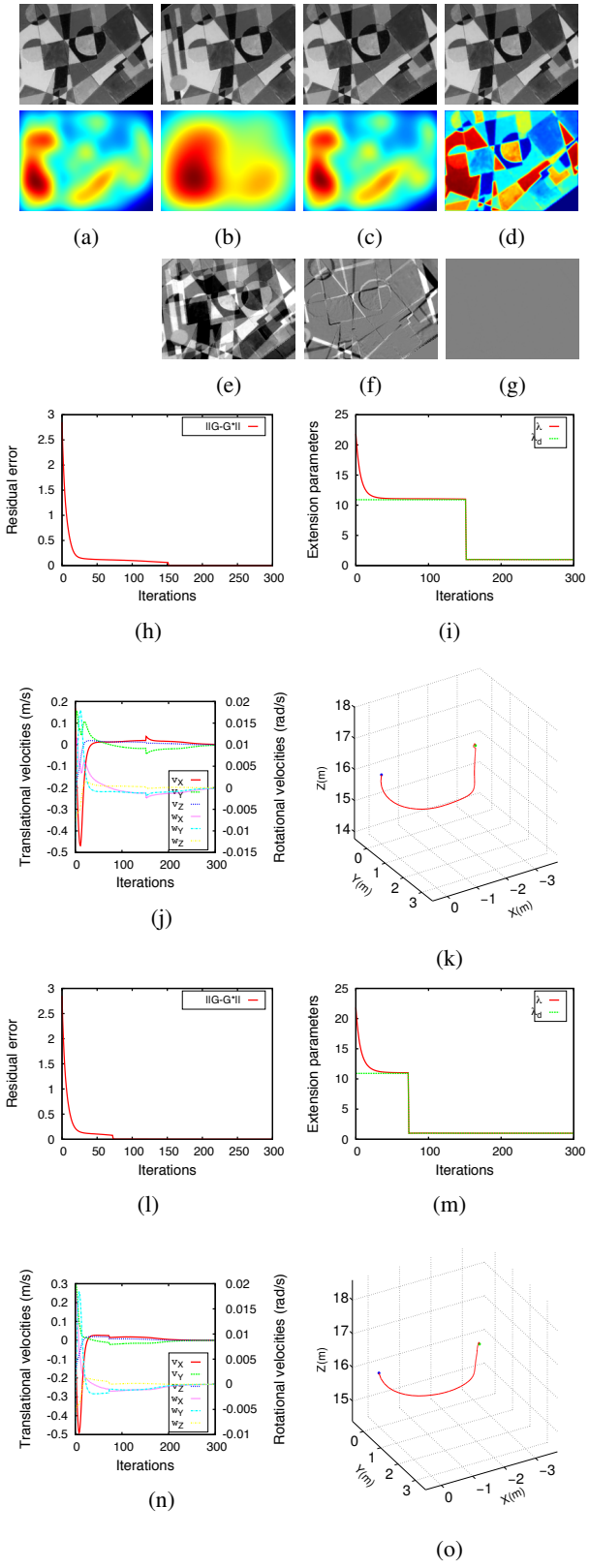


Fig. 7: Experiment #V2: Comparison between GTM (from h to k) and PGC (from l to o). (a) Desired image and its Gaussian mixture. (b) Initial image and its Gaussian mixture, (c) Image and its Gaussian mixture just before switching, (d) Final image and its Gaussian mixture. (e)-(g) Image of difference, (h) and (l) Residual error, (i) and (m) Extension parameters, (j) and (n) Velocities, (k) and (o) Trajectories.

used for the desired Gaussian mixture (Fig. 7i and Fig. 7m). The initial and final difference images (Fig. 7e and Fig. 7g) highlight the large gap at the beginning of the task and the accuracy of the convergence at its end.

Even if the GTM modeling is mathematically more rigorous than the PGC one, the obtained visual servoing behaviors are very close. PGC can actually be seen as a very good approximation of GTM. We use the PGC modeling for the following experimentations because of its smaller processing time.

B. Visual servoing in complex and 3D virtual environments

Experiment #V3 (Fig. 8): In addition to a complex textured plane, we consider a very challenging positioning task. As can be seen on the initial image of differences (Fig. 8b and 8d), the desired and the initial images do not share a lot of overlapped photometric information. In addition, the textured plane is partially outside the camera field-of-view in the initial image (Fig. 8). The displacements on $(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$[2] is given by $(10.31m, -4.39m, 0.1m, -4.58°, 0°, -13.75°)$. Despite this very large displacement and the high difference between the initial and the desired images, the Photometric Gaussian Mixture-based Visual Servoing has successfully controlled the camera motion to precisely reach the desired pose. The difference image at convergence is null. Of course, because of the very small initial overlapped photometric information, the trajectory taken by the camera to converge to the desired pose (Fig. 8k) is not straight.

Experiment #V4 (Fig. 9): In this simulation, a 3D scene is considered. We suppose that the depth of the scene is unknown. We used the same value ($\widehat{Z} = 50m$) as approximation for every pixel. In simulation, this depth is available and could be used in Equations (28) and (32), but we choose to "ignore" it in order to have relevant comparison with real scenes for which the depth is unknown. The displacement between the desired and the initial camera poses is $(26.95m, -5.02m, -11.14m, 14.40°, -27.54°, -5.88°)$. We can observe in Fig. 9e that image differences are also very large. Fig. 9c corresponds to the iteration just before the step transition. The camera converges perfectly to the desired pose (Fig. 9g) with a final pose error equal to $(1.6mm, 2.6mm, 4.1mm, 0.02°, 0.01°, 0.02°)$. We insist that the transformation between the initial and desired poses, and particularly the orientations around the two orthogonal axes to the optical camera axis, are very important and make this case very challenging. The visual differences between the initial and the desired images also reflect the challenging nature of this experiment.

C. Evaluation of the robustness to noise

Fig. 10 and Table I describe an experiment that evaluates the noise robustness of the proposed approach where the two models ((2) and (7)) of photometric Gaussian (Section II-C)

[2]In the following, all the differences between desired and initial image poses are given in this order: $(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$, with t_x, t_y and t_z are in meters, θ_x, θ_y and θ_z are in degrees while positioning errors at convergence are given in millimeters.

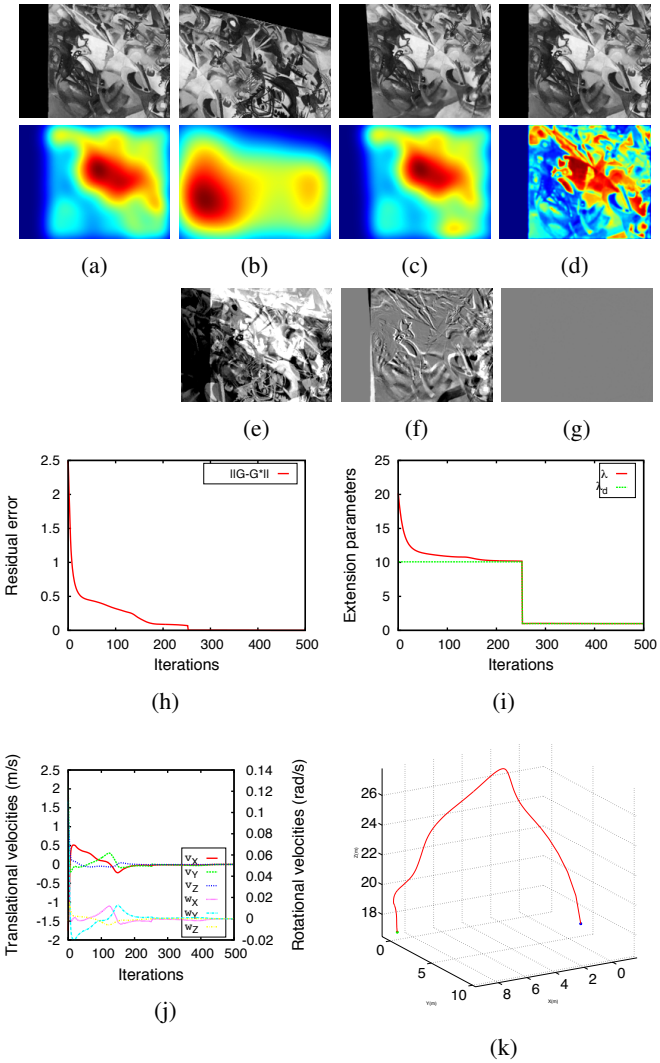


Fig. 8: Experiment #V3: Complex scene and large difference. (a) Desired image and its Gaussian mixture, (b) Initial image and Gaussian mixture, (c) Image and Gaussian mixture just before switching, (d) Final image and Gaussian mixture. (e)-(g) Image of difference, (h) Residual error, (i) Extension parameters, (j) Velocities, (k) Trajectory.

are compared. The displacement between the desired and the initial camera poses is always the same for each experiment: $(6.94m, 7.71m, 0.76m, 41.56°, 33.19°, 60.34°)$. First, visual servoing is carried out without any image noise (Fig. 10a). Then, a Gaussian noise is added on both the desired and the current images (Fig. 10b-e). More precisely, the noise added on the desired image is static and the noise added on the current images varies at each iteration. Between every experiment, the intensity of the noise is enhanced increasing its standard deviation σ.

Table I shows the final errors at convergence for the photometric Gaussian model proposed in [1] and the new model proposed in this paper. As we can see, photometric Gaussian mixtures based visual servoing is particularly robust against image noise. As expected from the theoretical expressions (Section II-C), the proposed new model of photometric

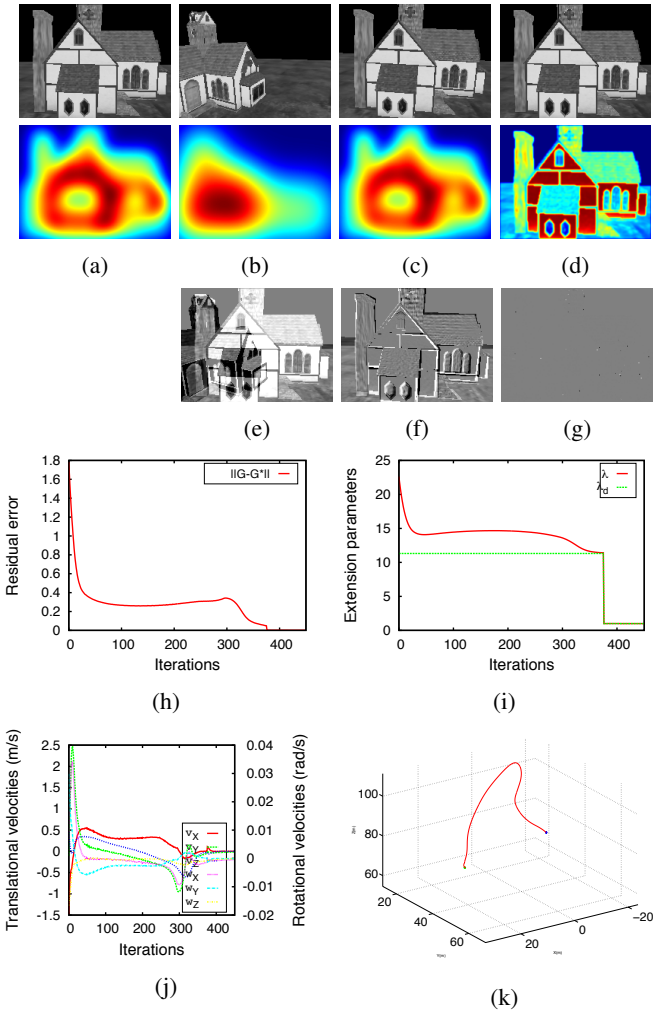


Fig. 9: Experiment #V4: Virtual 3D scene and large difference. (a) Desired image and its Gaussian mixture, (b) Initial image and its Gaussian mixture, (c) Image and its Gaussian mixture just before switching, (d) Final image and its Gaussian mixture. (e)-(g) Image of difference, (h) Residual error, (i) Extension parameters, (j) Velocities, (k) Trajectory.

TABLE I: Noise robustness evaluation: final error (position and orientation) of the estimated pose regarding the used photometric Gaussian models and the standard deviation σ of the Gaussian noise (Fig. 10).

	New model	**Model 1** [1]
Noiseless	$4.27mm, 0.00°$	$14.38mm, 0.04°$
$\sigma = 0.2$	$49.05mm, 0.21°$	$99.10mm, 0.14°$
$\sigma = 0.4$	$62.10mm, 0.04°$	$77.74mm, 0.23°$
$\sigma = 0.6$	$70.41mm, 0.07°$	$835.20mm, 2.01°$
$\sigma = 0.8$	$473.01mm, 1.36°$	$2006.46mm, 5.65°$

Gaussian is more robust to noise than Model 1. Of course, the accuracy at convergence decreases as the noise intensity increases but it remains very good regarding the excessively high noises that are added.

D. Comparisons with state-of-the-art methods

In this section we oppose our method (PGC) to two state-of-the-art methods: the pure luminance (PL) [4] and the weighted photometric moments (WPM) [10]. The PL method is based on a Levenberg-Marquart optimization scheme, while the two other methods use Gauss-Newton algorithm. Simulations are conducted to ensure that the conditions of experimentation are exactly the same. Fig. 11 shows the obtained camera 3D trajectories using the three methods in the case of a small translation facing a planar scene. We can observe that the WPM approaches gives a quasi-straight 3D trajectory, while those obtained with the PL and the PGC methods are more twisted.

The convergence efficiency of the PGC (Model 1 and the new one), PL and WPM methods are then compared for the difficult case of a 3D scene. The goal of this experiment is to reach a same desired pose starting from 20 random initial camera poses. Fig. 12a shows the desired image and Fig. 12 shows the initial images generated from these 20 random poses. The size of the images is 200×150 pixels for every method. The two photometric Gaussian models for the PGC approach are also compared for several initial extension parameters λ_{gi}.

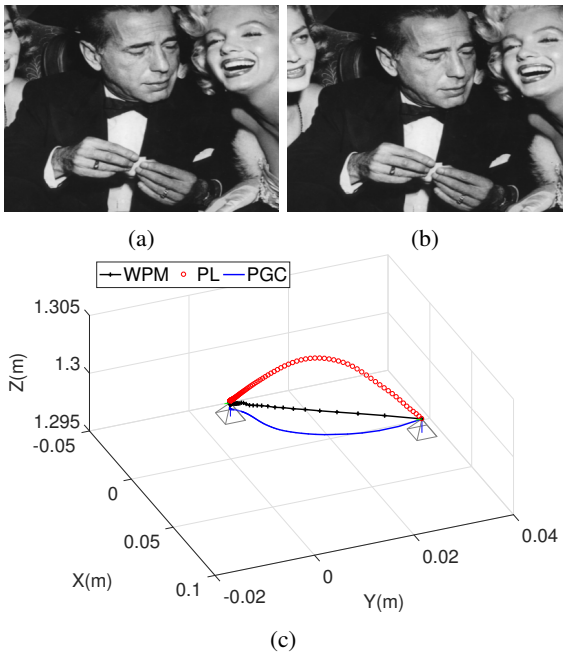
We consider that a camera has successfully converged to the desired pose when the final error is less than $(5.00mm, 5.00mm, 5.00mm, 0.1°, 0.1°, 0.1°)$. Table II provides the successful and failed convergences of this experiment. Note that each method has the same parametrization throughout the experiments. The new model of photometric Gaussian is slightly better than the Model 1 proposed in [1]. When the extension parameter of the new model is initialized with a high enough value, the PGC approach converges for almost every initial pose (and even all the 20 poses when $\lambda_{gi} = 25$). Even if the results obtained with the Model 1 are good too, it is more difficult to identify a unique initial value of λ_{gi} that works for every initial pose. The PL approach has only drove the camera to the desired pose in 3 out of the 20 initial poses and numerous iterations were required. The WPM method diverges completely for 11 poses, which is not surprising due to the large discrepancies between the initial and desired images. For the 9 poses marked with an orange check mark in Table II, it drives the camera next to the desired pose with a non negligible final visual alignment. This is probably due to the fact that a 3D scene is considered, which implies that parts of the scene appear and disappear during the camera motion, inducing ambiguities in the values of the photometric moments involved. However, the final visual alignments are low enough to complete the convergence to the desired pose by switching for instance to the PL method.

Finally, Fig. 13 reports some challenging cases with different scenes where large parts of the desired image are absent in the initial image. In these situations, only the PGC method drives successfully the camera to the desired poses. For all these cases, the extension parameter has been initialized with the same value $\lambda_{gi} = 5$.

To conclude this comparison, the PL is very accurate, fast and does not need any parameter tuning, but the convergence

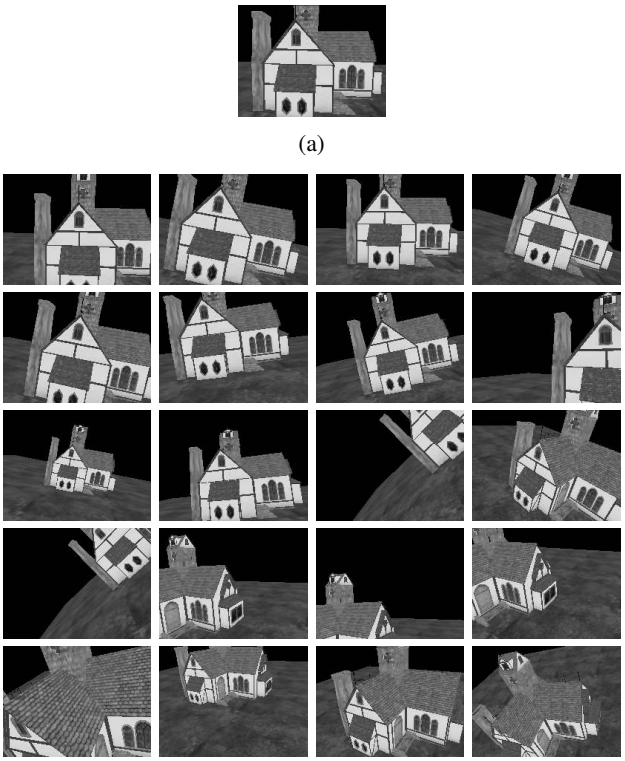
(a) Noiseless (b) $\sigma = 0.2$ (c) $\sigma = 0.4$ (d) $\sigma = 0.6$ (e) $\sigma = 0.8$

Fig. 10: Noise robustness evaluation: Desired images (first row) and Initial images (second row) increasing the standard deviation σ of the Gaussian noise. Results of this evaluation are shown in Table I.



(a) (b) (c)

Fig. 11: (a) Desired image, (b) Initial image and (c) Cameras 3D trajectory obtained using the PL, the WPM and the PGC methods.



(a) (b)

Fig. 12: PGC, PL and WPM methods comparion. (a) Desired image and (b) Initial images generated from 20 random camera poses. The results of this experiment are shown in Table II.

domain is limited. The WPM enlarges the convergence domain, it is fast and guarantees a more direct 3D trajectory, when it converges but it is inefficient in case of 3D scenes when parts appear or disappear near the center of the image. Our PGC method enlarges drastically the convergence domain and is accurate, but it is more time consuming. As a comparison, considering 100×100 pixels images, one iteration of the PL method is done in less than $20ms$. One iteration of our parallelized version of the PGC method takes $100ms$ on a Intel Core I7 2.3GHz with a GeForce GT 630M GPU running Linux. This allows a frequency for the servo loop around $10Hz$.

E. Real experiments

Three real experiments using a 6 axis industrial robot (Stäubli TX60) with a perspective camera mounted on its end-effector are presented. Figs. 14, 16 and 18 show this robot and the experimental environments. The intrinsic parameters of the camera have been estimated. The depth Z is unknown and is supposed constant for every pixel all along the motion of the camera.

Experiment #R5 (Fig. 15): The goal of this experiment is to demonstrate that the proposed visual servoing works well even under common lighting conditions with large

TABLE II: PGC, PL and WPM methods comparion. Successful (✓) and failed (✗) convergences for 20 random camera initial poses (Fig. 12). An orange mark (❍) means that the camera has converged next to the desired pose with a non negligible final visual alignment. For the two PGC approaches, several initial extension parameters have been used.

	PGC - New model					**PGC - Model 1**					**PL**	**WPM**
	12	18	25	31	40	0.1	0.15	0.2	0.25	0.3		
Pose 1	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	❍
Pose 2	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗	❍
Pose 3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	❍
Pose 4	✗	✗	✓	✓	✓	✗	✗	✓	✓	✗	✗	❍
Pose 5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	❍
Pose 6	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	❍
Pose 7	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	❍
Pose 8	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
Pose 9	✓	✓	✓	✓	✓	✗	✓	✓	✗	✗	✗	❍
Pose 10	✓	✓	✓	✗	✗	✓	✗	✓	✓	✓	✗	❍
Pose 11	✗	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗
Pose 12	✗	✗	✓	✓	✓	✗	✗	✓	✗	✗	✗	✗
Pose 13	✗	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗
Pose 14	✗	✗	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗
Pose 15	✗	✗	✓	✓	✗	✓	✓	✓	✗	✗	✗	✗
Pose 16	✗	✗	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗
Pose 17	✗	✓	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗
Pose 18	✓	✓	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗
Pose 19	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗
Pose 20	✗	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗

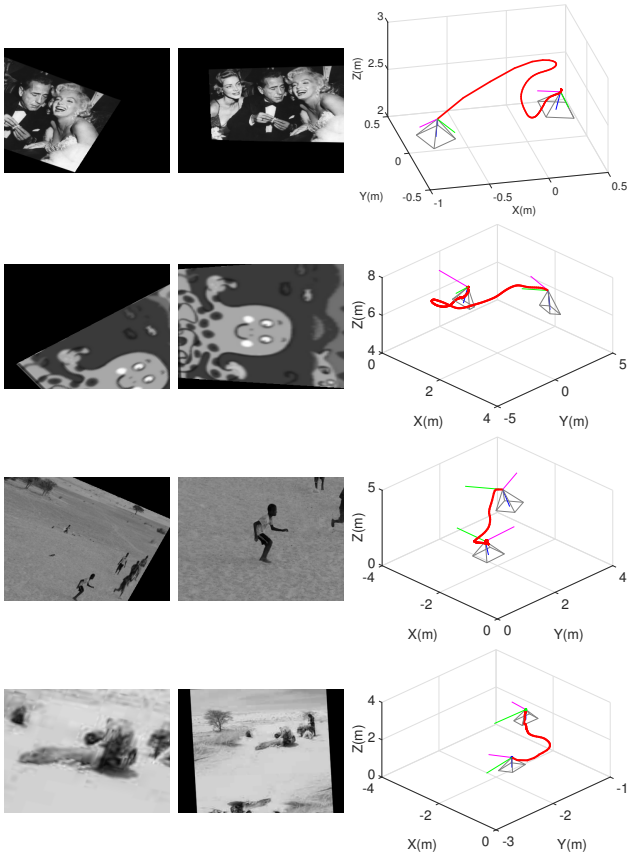


Fig. 13: Other examples with large difference between initial and desired images, and 3D trajectories for the PGC method.

differences in the images. As it can be seen in Fig. 14, the scene contains several 3D objects. The mean distance between the scene and the camera is about 0.5 meter at the desired pose. We used this distance as depth $\hat{Z}$ for every pixel to compute the interaction matrix. This scene is quasi-planar with a relative depth of approximatively $0.02m$. The initial displacement is composed by translations and rotations $(-0.043m, -0.300m, 0.018m, 20.80°, -7.84°, 5.97°)$. Several occlusions have been voluntary introduced during the visual servoing (Figs. 15h-15k). These disturbances affect the behavior of the control as it can be seen on the residual error, velocities and extension parameters curves around the iterations 500 and 1100. At convergence, the final image of differences (Fig. 15g) is not absolutely null due to a global illumination change between the beginning and the end of the experiment, but the final pose error is very small $(1.48mm, 0.97mm, 0.54mm, -0.13°, -0.21°, 0.06°)$.

Experiment #R6 (Fig. 16 and Fig. 17): As previously, this experiment has been conducted under common lighting conditions with a large displacement between the initial and the desired camera poses $(0.0m, 0.265m, 0.040m, -0.89°, 0.05°, 27.04°)$. The visual difference between the initial and desired images is also important. The 3D scene contains various shapes of objects (monitors, robots, specular surfaces, ...) and different colors as it can be seen in Fig. 16. The relative depth of the 3D objects present in the scene is around one meter. Moreover, scene occlusions and lighting changes have been introduced during the experiment between the iterations 50 and 150. For example, the Fig. 17d shows the intrusion of an object in the camera field-of-view. This explains why the curves (Figs. 17e, 17f and 17g) are shaky. Because of our robot singularities and its limited working space, the initial displacement in terms of image differences is not as impressive as for the experiments conducted in virtual environments. However, this experiment shows that the visual servoing based on photometric Gaussian Mixtures works well even in real conditions. The final error at convergence is $(-1.39mm, 6.67mm, -1.25mm, -0.9°, 0.05°, 0.95°)$. This relative accuracy is due to variations in light conditions.

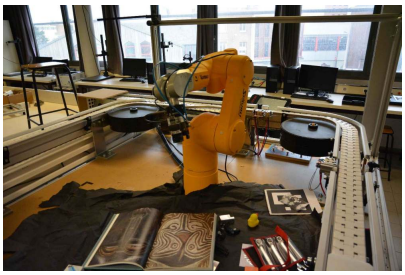


Fig. 14: Experiment #R5: Experimental environment and the robot (Staubli TX60).

Experiment #R7 (Fig. 18 and Fig. 19): The last experiment is achieved in a real environment (windows, tables, mobile robots, ...) with a large relative depth as it can be observed in Fig. 18. The displacement between the initial and the

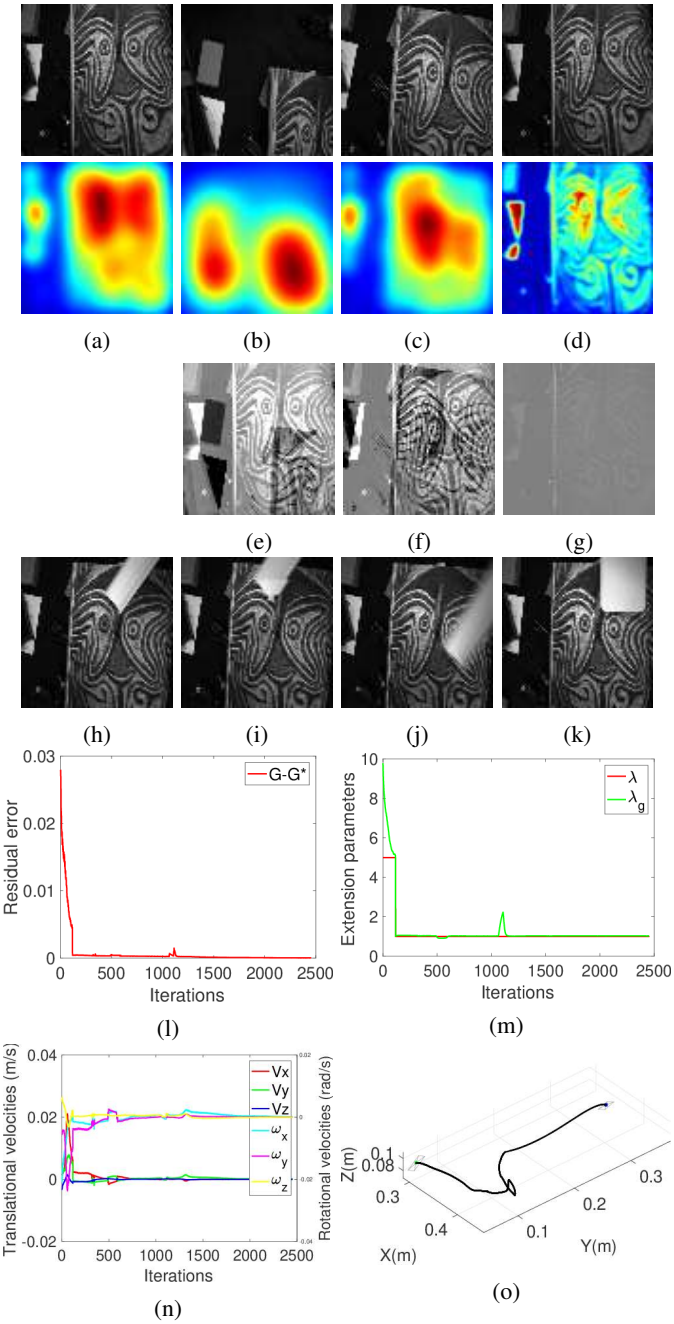


Fig. 15: Experiment #R5: Real 3D scene. (a) Initial image and Gaussian mixture, (b) Image and Gaussian mixture just before switching, (c) Image and Gaussian mixture just after switching, (d) Desired image and its Gaussian mixture, (e)-(g) Image of difference, (h)-(k) Examples of images with occlusions, (l) Residual error, (m) Extension parameters, (n) Velocities, (o) Trajectory.

desired poses is $(0.194m, 0.0584m, 0.062m, 0°, 0°, 14.10°)$. However, as usual we used the same depth $\hat{Z} = 2m$ for every pixel. As before, scene occlusions and lighting changes have been introduced during the process. The two columns of Fig. 19c show respectively external views of the environment and the image acquired by the robot camera at four moments of the visual servoing process. More precisely, the first row

Fig. 16: Experiment #6: Experimental environment and the view of the scene from the camera.

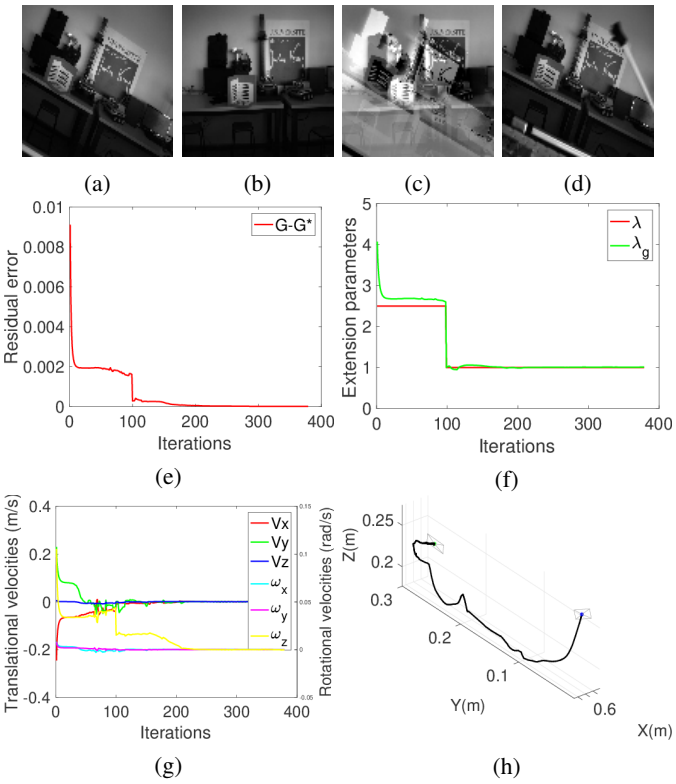


Fig. 17: Experiment #R6: Real 3D scene. (a) Initial image, (b) Desired image, (c) Image of difference, (d) Image with object intrusion, (e) Residual error, (f) Extension parameters, (g) Velocities, (h) 3D Trajectory.

corresponds to the start, the second shows an obstruction of the light source, the third highlights the intrusion of a person in the camera field-of-view, occluding the top-right part of the scene, and the fourth row corresponds to the desired state. The introduced perturbations explain why the residual error curve (Fig. 19i) is not smooth. Despite these perturbations, the visual servoing converges and remains stable thanks to the redundancy of the used information. The final error is $(7.8mm, 3.5mm, 0.5mm, 0.01°, 0.52°, 0.39°)$. The convergence is a bit less accurate than for the previous experiments. This is due to the higher relative depth of the scene and to the natural light coming from the windows (Figure 19).

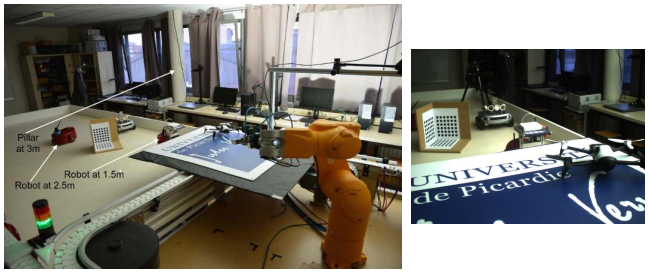


Fig. 18: Experiment #7: Experimental environment (with distances between some objects and the robot) and the view of the scene from the camera.

F. Discussion

The previous results show that our method ensures the convergence even if there is almost no overlapping in the desired and initial images within realistic 3D environments. However, as explained in Section IV two parameters are involved in our method: the initial value λ_{gi} of the extension parameter and the switching threshold. In general, the choice of these parameters depends on the images and on the difference between the desired and initial images. As it is very difficult to establish the exact relation to estimate these values, we propose an empiric strategy:

- λ_{gi} must confer to the Gaussian mixture a huge power of attraction (Fig. 3f).
- $\lambda_g{}^*$ must guarantee that the desired Gaussian mixture overlaps the current one.

The first aspect can be justified by the relation that is observed between the size of the Gaussian Mixture (close to a unimodal Gaussian for a large λ_{gi}) and the size of the image. In the presented experiments, $V2, V3$ and $V4$ have all the same λ_{gi} because the size of the images (200×150 pixels) is the same and these images are textured. In contrary, for experiment $V1$, the value of λ_{gi} is large because there is no texture in the image thus the Gaussian centers are far away. This is an unusual case, which explains why λ_{gi} has a high value. For the real experiments, the values of λ_{gi} are lower than those used in virtual scenes because the images size has been reduced to speed up the calculation time for the robot control. The link between the size of the image and λ_{gi} appears in Table III where the reported values of λ_{gi} have permitted the convergence of the visual servoing regarding three sizes of the image (Fig. 20).

TABLE III: λ_{gi} interval related to the image size.

Image size	λ_{gi}
(40×40)	≈ 2 .. 20
(60×60)	≈ 2 .. 30
(80×80)	≈ 2 .. 40

However, a too large λ_{gi} leads either to divergence or to an imprecise control of the rotations because the Gaussian mixture is close to a unimodal Gaussian. To show the influence of λ_{gi}, we present in Fig. 20, 3D trajectories produced by the visual servoing for an easy case (small difference between desired and initial images). We can observe that the quality

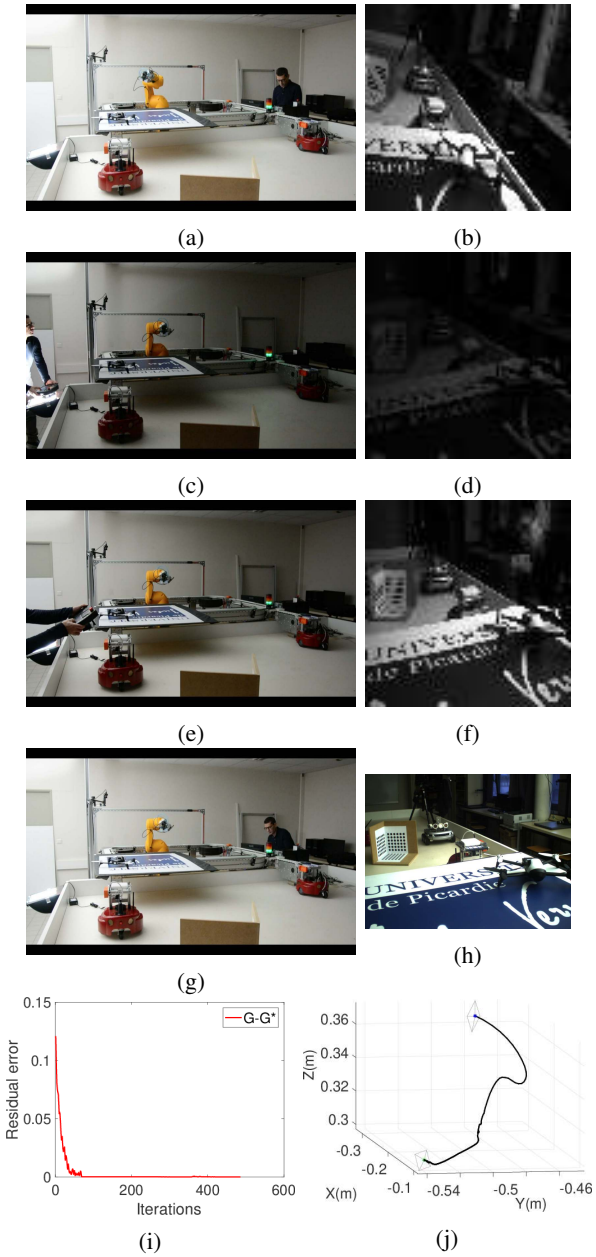


Fig. 19: Experiment #R7: Real 3D scene with voluntary disturbances. External views of the environment and the image acquired by the robot camera at four moments of the experiment: (a, b) Initialization, (c, d) High light variation, (e, f) Important occlusion and (g, h) Desired pose. (i) Residual error, (j) 3D Trajectory.

of trajectories depends on the chosen value of λ_{gi}, but the visual servoing converges for a large interval of this parameter: $\lambda_{gi} \in [2, 20]$ for a 40×40 image size. So, a value between 2 and the half image size can be assigned to λ_{gi}.

The second parameter is the switching threshold. It is not useful when the scene is very simple. For example, in Fig. 6, the scene contains only one object and a Gaussian mixture with a large value of λ_{gi} is sufficient to cancel the difference between the desired and the current images. When the scene is complex, the value of the switching threshold

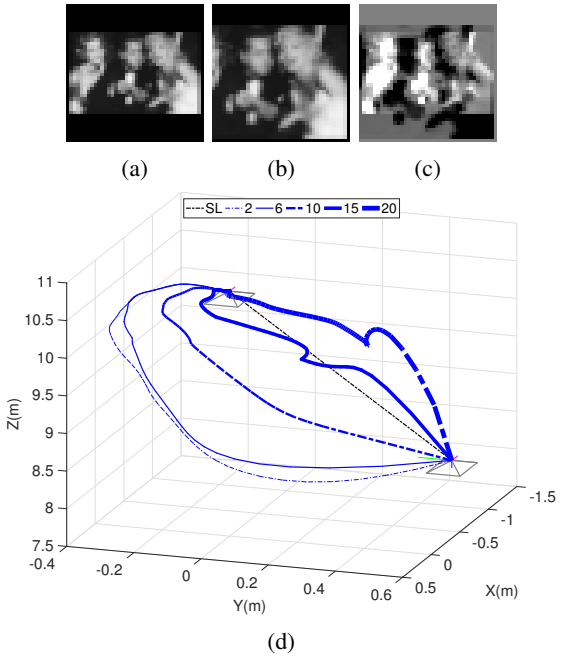


Fig. 20: Influence of the initial extension parameter: (a) desired, (b) initial and (c) difference images ((40×40) pixels) and (d) camera trajectories. Legend: SL for straight line between initial and final camera pose; 2, 6, ..., 20 are the values of λ_{gi}

.

S has consequences on the behavior of the visual servoing. If it is too small, the algorithm switches too late. Then, λ_g stays large and the visual servoing may diverge because the orientations are poorly controlled in this case. If this threshold is too high, switching happens too early and the visual servoing does not converge because the current image is too far from the desired image (local minimum). However, for all our different experiments, we set a same threshold equal to 0.1.

VI. Conclusion

We introduced in this paper Photometric Gaussian Mixtures as visual features for dense visual servoing. Limitations of the convergence domain regarding the pure photometric feature have encouraged this research. Usual dense visual servoing methods fail if there is not enough shared photometric areas between the desired and the initial images. Our basic idea was to assign a power of attraction to each image pixel. For that, instead of using images as intensity pulses, we consider every pixel as a Gaussian function. The combination of every Gaussian creates a Photometric Gaussian Mixture which is a representation of the image. Thanks to this representation, even if there is almost no overlapping between the desired and the initial images, the convergence domain is sufficiently enlarged to drive the camera to the desired pose. Beyond the power of attraction concept, Photometric Gaussian Mixtures are also tunable. Indeed, in addition to the camera velocities, the extension parameter is also optimized during the servoing. The variation of the Gaussian extension allows us to enlarge the convergence domain and to ensure a convergence as accurate as with the pure photometric feature.

To validate the Photometric Gaussian Mixtures as visual features, a Gauss-Newton control law has been used to minimize the proposed cost-function. The interaction matrix - key of the visual servoing - has been modeled for the proposed Photometric Gaussian Mixtures. We have presented and compared two modeling approaches. As for the photometric moments, the first one uses the Green's theorem to avoid the computation of the image gradients. The second approach is based on 3D assumptions and can be seen as a good approximation of the first modeling. Both simulation and real experiments have been led that confirm the validity of the two modelings, over performing the previous dense visual servoing approaches.

References

[1] N. Crombez, G. Caron, and E. Mouaddib, "Photometric gaussian mixtures based visual servoing," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 5486–5491.

[2] 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.

[3] C. Collewet, E. Marchand, and F. Chaumette, "Visual servoing set free from image processing," *IEEE ICRA*, pp. 81–86, May 2008.

[4] C. Collewet and E. Marchand, "Photometric visual servoing," *IEEE Trans. on Robotics*, vol. 27, no. 4, pp. 828–834, 2011.

[5] B. Delabarre and E. Marchand, "Visual Servoing using the Sum of Conditional Variance," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'12*, Vilamoura, Portugal, 2012, pp. 1689–1694.

[6] A. Comport, M. Pressigout, E. Marchand, and F. Chaumette, "A visual servoing control law that is robust to image outliers," vol. 1, pp. 492–497, October 2003.

[7] A. Dame and E. Marchand, "Improving mutual information based visual servoing," *IEEE ICRA*, pp. 5531–5536, May 2010.

[8] Q. Bateux and E. Marchand, "Direct visual servoing based on multiple intensity histograms," in *IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015*, 2015, pp. 6019–6024.

[9] M. Bakthavatchalam, F. Chaumette, and E. Marchand, "Photometric moments: New promising candidates for visual servoing," *IEEE ICRA*, pp. 5521–5526, May 2013.

[10] M. Bakthavatchalam, F. Chaumette, and O. Tahri, "An Improved Modelling Scheme for Photometric Moments with Inclusion of Spatial Weights for Visual Servoing with Partial Appearance/Disappearance," in *IEEE Int. Conf. on Robotics and Automation, ICRA'15*, Seattle, United States, May 2015.

[11] V. Kallem, M. Dewan, J. P. Swensen, G. D. Hager, and N. J. Cowan, "Kernel-based visual servoing," *IEEE IROS*, pp. 1975–1980, 2007.

[12] J. P. Swensen, V. Kallem, and C. N. J., "Empirical characterization of convergence properties for kernel-based visual servoing," *Visual Servoing via Advanced Numerical Methods*, pp. 23–38, 2010.

[13] M. Ourak, B. Tamadazte, O. Lehmann, and N. Andreff, "Wavelets-based 6 dof visual servoing," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3414–3419.

[14] L.-A. Duflot, A. Krupa, B. Tamadazte, and N. Andreff, "Toward Ultrasound-based Visual Servoing using Shearlet Coefficients," in *IEEE Int. Conf. on Robotics and Automation, ICRA'16*, Stockholm, Sweden, May 2016. [Online]. Available: https://hal.inria.fr/hal-01304753

[15] A. H. A. Hafez, S. Achar, and C. V. Jawahar, "Visual servoing based on gaussian mixture models." *IEEE ICRA*, pp. 3225–3230, 2008.

[16] F. Boughorbel, M. Mercimek, A. F. Koschan, and M. A. Abidi, "A new method for the registration of three-dimensional point-sets: The gaussian fields framework," *Image Vision Comput.*, vol. 28, no. 1, pp. 124–137, 2010.

[17] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

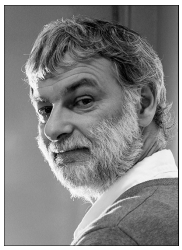
[18] E. Malis, "Improving vision-based control using efficient second-order minimization techniques," *IEEE ICRA*, pp. 1843–1848, 2004.

[19] L. Hammouda, K. Kaaniche, H. Mekki, and M. Chtourou, "Real-time visual servoing based on new global visual features," *Informatics in Control, Automation and Robotics*, pp. 183–196, 2014.

Crombez Nathan conducted his doctoral research at the MIS laboratory and received the Ph.D degree in computer vision for robotics from the University of Picardie Jules Verne, Amiens, France, in 2015. Since 2018, he is Associate Professor ("Maître de Conférences") in the EPAN (Environment Perception and Autonomous Navigation) Research Group of the le2i laboratory at the University of Technology Belfort-Montbliard (Belfort, France). His research interests are mainly focused on the perception and the navigation of robots and autonomous vehicles based on unconventional vision systems.

El Mustapha Mouaddib received the Ph.D. degree in robotics and the Habilitation degree from the University of Picardy Jules Verne, France, in 1991 and 1999, respectively. Since 2001, he is a full professor in the University of Picardy Jules Verne (France). His main research interests are computer vision and artificial perception for mobile robotics. From 1995, he has been a head of Perception on robotics group, where he is involved in research projects on omnidirectional vision and structured light. Since 2010, he is a manager of a big research program about building and using a virtual 3D model of Cathedral of Amiens. He has been an associate editor of IEEE ICRA and IEEE IROS and he is associate editor of IEEE RA-L (Robotics and Automation Letters) journal (2015-2018).

Guillaume Caron is Associate Professor ("Maître de Conférences") since 2011 and he is heading the Robotic Perception group of the MIS laboratory since 2016 at Université de Picardie Jules Verne (France). He received the Ph.D. degree in robotics from the same university in 2010. He spent one year (2010-2011) as a postdoctoral associate at INRIA Rennes (France), in the Lagadic group. He was also a visiting research scholar at the University of Osaka (Japan), in the Yagi laboratory, for two months in 2013. His research interests include artificial vision for robotics, real-time visual tracking and servoing.



François Chaumette (M'02, SM'09, F'13) was graduated from École Nationale Supérieure de Mécanique, Nantes, France, in 1987. He received the Ph.D degree in computer science from the University of Rennes in 1990. Since 1990, he has been with Inria at Irisa in Rennes. His research interests include robotics and computer vision, especially visual servoing and active perception.

Dr Chaumette received the AFCET/CNRS Prize for the best French thesis in automatic control in 1991. He also received the 2002 King-Sun Fu Memorial Best IEEE Trans. on Robotics and Automation Paper Award. He was Founding Senior Editor of the IEEE Robotics and Automation Letters. He is currently in the Editorial Board of the Int. Journal of Robotics Research, and Senior Editor of the IEEE Trans. on Robotics.