Omnidirectional Visual Servoing using the Normalized Mutual Information

Bertrand Delabarre, Guillaume Caron, Eric Marchand

Omnidirectional Visual Servoing using the Normalized Mutual Information

Bertrand Delabarre * **Guillaume Caron** ** **Eric Marchand** *

* *Université de Rennes 1, IRISA, Lagadic team, Rennes, France (e-mail: bertrand.delabarre@irisa.fr, eric.marchand@irisa.fr).*
** *Université de Picardie, Jules Verne, Amiens, France (e-mail: guillaume.caron@u-picardie.fr)*

Abstract: Geometrical features have been a key element of visual servoing for several years. Recently, different works have shown how using all the information within the images can lead to successful servoing tasks. In particular, works using mutual information have been proposed and tested on perspective cameras. In this paper we propose to adapt this approach to vision systems following the unified sphere model for central cameras using a normalized version of the mutual information. This will permit to apply the technique to large fields of view with a more reliable similarity function. Several experiments are performed on a cartesian robot with a fisheye camera to validate our approaches.

Keywords: Vision, visual servoing, mutual information, information theory, robotics.

1. INTRODUCTION

Visual servoing uses the information provided by a vision sensor to control the movements of a dynamic system (Hutchinson et al. (1996); Chaumette and Hutchinson (2006); Chesi and Hashimoto (2010)). This approach requires to extract and track visual information (usually geometric features) from the image in order to design the control law. This tracking process is one of the bottlenecks in the development of visual servoing techniques.

Other works have tried to circumvent these problems by using directly the information provided by the entire image (Nayar et al. (1996); Deguchi (2000); Kallem et al. (2007); Collewet and Marchand (2011); Dame and Marchand (2011)). Features are no longer extracted from the image. Those works have began with Nayar et al. (1996) and later on Deguchi (2000), where the images were reduced to eigenspaces. Later works have used directly the whole images. In Collewet and Marchand (2011), a control law was proposed that minimizes the error between the current image and the desired one. In that case the vector of visual feature is nothing but the image itself and the error to be regulated is the sum of squared differences (the SSD). This approach features many advantages: it does not require any matching or tracking process. Furthermore since the image measurements are nothing but the pixel intensity, there is no error in the feature extraction process leading to a very precise realization of the task. The method was later extended to omnidirectional camera in Caron et al. (2010). Kallem et al. (2007) also considered the pixels intensity with a kernel-based method that leads to a highly decoupled control law. However, this approach can not control the 6 degrees of freedom of the robot and it is also very limited in the case of appearance variations.

As previously stated, considering image intensities (Collewet and Marchand (2011)) is quite sensitive to modification of the environment and more robust registration functions should be considered. To solve this problem, the considered approach does not use directly the luminance of the pixels but the information contained in the images. The visual feature is the mutual information defined by Shannon (1948). The mutual information (built from the image entropy) of two random variables (images) measures their mutual dependence. This function does not directly compare intensities of the two images but the distribution of the information in the images. Considering two images, the higher the mutual information (MI), the better the alignment between the two images is. Mutual information was considered for positioning and navigation task using visual servoing in Dame and Marchand (2011). Since this is an entropy based measure, it is very robust towards illumination changes, occlusions or multimodality.

In this paper, we extend the approach proposed in Dame and Marchand (2011) in two directions. First, we consider a new expression for mutual information which is normalized. This measure, proposed by Studholme and Hawkes (1999) to perform image alignment, is bounded which makes it easier to interpret when used in an optimization process. Then, we extend the proposed method to the use of the unified model of central projection cameras (Barreto (2001)). Omnidirectional cameras allow to consider more information on the surrounding environment which improves the efficiency of the approach.

This paper is organized as follows. First, direct visual servoing approaches are discussed, before describing our approach based on normalized mutual information. The adaptation to the use of the unified model of central projection cameras is then exposed. Finally, several experiments are exposed to validate our work.

2. DIRECT VISUAL SERVOING

Visual servoing tasks have used geometrical features for a long time (Chaumette and Hutchinson (2006)). Recently, works have proposed new direct approaches. They achieve a direct visual servoing using the information provided by the whole image, thus eliminating the need to extract and track features. In Collewet and Marchand (2011) the authors used directly

the pixel luminance whereas in Dame and Marchand (2011) a measure of mutual information was chosen.

2.1 Positioning Task

The aim of a positioning task is to reach a desired pose of the camera $\mathbf{r}^*$, starting from an arbitrary initial pose. To achieve that goal, one needs to define a cost function which will indicate whether or not the camera is going in the right direction. Most of the time this cost function f is a dissimilarity measure, function of the camera pose, which needs to be minimized by controlling the camera. Considering the actual pose of the camera $\mathbf{r}$ the visual servoing problem can therefore be written as an optimization process:

$$\widehat{\mathbf{r}} = \arg\min_{\mathbf{r}} f(\mathbf{r}, \mathbf{r}^*) \tag{1}$$

where $\widehat{\mathbf{r}}$, the pose reached after the optimization (servoing process), is the closest possible to $\mathbf{r}^*$ (optimally $\widehat{\mathbf{r}} = \mathbf{r}^*$). For example, considering a set of geometrical features $\mathbf{s}$, the task will typically have to minimize the difference between **s(r)** and the desired configuration **s*** which leads to:

$$\widehat{\mathbf{r}} = \arg\min_{\mathbf{r}} (\mathbf{s}(\mathbf{r}) - \mathbf{s}^*). \tag{2}$$

This visual servoing task is achieved by iteratively applying a velocity to the camera. This requires the knowledge of the interaction matrix (also known as image Jacobian) $\mathbf{L_s}$ of **s(r)** that links the variation of $\dot{\mathbf{s}}$ to the camera velocity and which is defined as:

$$\dot{\mathbf{s}}(\mathbf{r}) = \mathbf{L_s}\mathbf{v} \tag{3}$$

where $\mathbf{v}$ is the camera velocity. This equation leads to the expression of the velocity that is applied to the robot as it is defined in Chaumette and Hutchinson (2006) by:

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}_\mathbf{s}}^{+} (\mathbf{s}(\mathbf{r}) - \mathbf{s}^*) \tag{4}$$

where λ is a convergence factor tuned to improve convergence speed.

2.2 Using the mutual information

In Dame and Marchand (2011), it was proposed to use the mutual information defined by Shannon (1948) as the cost function of the positioning task:

$$\widehat{\mathbf{r}} = \arg\max_{\mathbf{r}} \mathrm{MI}\left(\mathbf{I}(\mathbf{r}), \mathbf{I}^*\right). \tag{5}$$

The mutual information can be defined as the quantity of information shared by two signals. It is an entropy-based measure and its main advantage is that it is very robust. Since it is based on the information itself rather than on its representation it is resistant to many variations as occlusions or illumination changes. That is the reason why it can be very useful, as the conditions when effecting a visual servoing task can change with time. The computation of mutual information between two images is given by the following equation:

$$\mathrm{MI}(\mathbf{I}(\mathbf{r}), \mathbf{I}^*) = \mathrm{H}(\mathbf{I}(\mathbf{r})) + \mathrm{H}(\mathbf{I}^*) - \mathrm{H}(\mathbf{I}(\mathbf{r}), \mathbf{I}^*) \tag{6}$$

where $\mathrm{H}(\mathbf{I}(\mathbf{r}))$ is the measure of entropy of the image $\mathbf{I}(\mathbf{r})$ and $\mathrm{H}(\mathbf{I}(\mathbf{r}), \mathbf{I}^*)$ the joint entropy of the images $\mathbf{I}(\mathbf{r})$ and $\mathbf{I}^*$. They are defined as:

$$\mathrm{H}(\mathbf{I}(\mathbf{r})) = -\sum_{i=0}^{N_{c_\mathbf{I}}} p_\mathbf{I}(i) \log\left(p_\mathbf{I}(i)\right) \tag{7}$$

$$\mathrm{H}(\mathbf{I}(\mathbf{r}), \mathbf{I}^*) = -\sum_{i=0}^{N_{c_\mathbf{I}}} \sum_{j=0}^{N_{c_{\mathbf{I}^*}}} p_{\mathbf{II}^*}(i,j) \log\left(p_{\mathbf{II}^*}(i,j)\right) \tag{8}$$

where N_c is the dynamic of the considered image (typically 255), $p_\mathbf{I}(i)$=$\Pr(\mathbf{I}(\mathbf{x})$=$i)$ the probability distribution of i and $p_{\mathbf{II}^*}(i,j)$=$\Pr(\mathbf{I}(\mathbf{x})$=$i \cap \mathbf{I}^*(\mathbf{x})$=$j)$ the joint probability distribution function. The problem of that cost function is that it has no fixed upper bound. If the values of entropy of $\mathbf{I}(\mathbf{r})$ and $\mathbf{I}^*$ change, the maximum possible value of $\mathrm{MI}(\mathbf{I}(\mathbf{r}), \mathbf{I}^*)$ changes thus making it impossible to compare values in different situations. Using a normalized mutual information, the task has an identifiable goal that it is trying to reach. This makes it easier to evaluate and makes comparison possible between different situations.

2.3 Using the Normalized Mutual Information

To get rid of the drawbacks of mutual information we propose in this paper to use a normalized measure of mutual information as defined in Studholme and Hawkes (1999). It is expressed as:

$$\mathbf{NMI}(\mathbf{I}(\mathbf{r}), \mathbf{I}^*) = \frac{\mathrm{H}(\mathbf{I}(\mathbf{r})) + \mathrm{H}(\mathbf{I}^*)}{\mathrm{H}(\mathbf{I}(\mathbf{r}), \mathbf{I}^*)}. \tag{9}$$

This measure is bounded. To prove it, let us first consider $\mathbf{I}(\mathbf{r})$ and $\mathbf{I}^*$ not sharing any information. This results in $\mathbf{H}(\mathbf{I}(\mathbf{r}), \mathbf{I}^*) = \mathbf{H}(\mathbf{I}(\mathbf{r})) + \mathbf{H}(\mathbf{I}^*)$ leading to $\mathbf{NMI}(\mathbf{I}(\mathbf{r}), \mathbf{I}^*) = \mathbf{1}$. For the upper bound, by definition:

$$\mathbf{H}(\mathbf{I}(\mathbf{r}), \mathbf{I}^*) \geq \max\left[\mathbf{H}(\mathbf{I}(\mathbf{r})), \mathbf{H}(\mathbf{I}^*)\right] \tag{10}$$

which gives:

$$\frac{1}{\mathbf{H}(\mathbf{I}(\mathbf{r}), \mathbf{I}^*)} \leq \frac{1}{\max\left[\mathbf{H}(\mathbf{I}(\mathbf{r})), \mathbf{H}(\mathbf{I}^*)\right]}. \tag{11}$$

Knowing this, the upper bound of the NMI can be shown as:

$$\mathbf{NMI} \leq \frac{\max\left[\mathbf{H}(\mathbf{I}(\mathbf{r})), \mathbf{H}(\mathbf{I}^*)\right](1+\beta)}{\max\left[\mathbf{H}(\mathbf{I}(\mathbf{r})), \mathbf{H}(\mathbf{I}^*)\right]} \text{ with } \beta \leq 1 \tag{12}$$

$$\mathbf{NMI} \leq 1 + \beta. \tag{13}$$

The best case scenario being $\mathbf{I}(\mathbf{r})$ and $\mathbf{I}^*$ sharing all their information, therefore $\beta = 1$, equation (13) becomes:

$$\mathbf{NMI}(\mathbf{I}(\mathbf{r}), \mathbf{I}^*) = 2. \tag{14}$$

2.4 Shape of the cost function

The goal of this subsection is to see if the shape of the chosen cost function makes it suitable for an optimization process. In order to do that, a comparison between two images is made. On a reference image, a template of 100x100 pixels is extracted from a determined chosen starting position in the image. Then, a current image is chosen and a first patch is extracted beginning with a translation in the image of tx=-10 pixels and ty=-10 pixels with relation to the starting point of the template. The value of NMI between those two images of 100x100 pixels is then computed which gives the coordinate of the point (-10,-10) on the 3D plot. That process is iterated incrementing tx and ty until they both reach a value of 10 pixels. Figure 1 shows the different comparisons made when images are reduced to a dynamic of 8 grey levels. A first comparison was made to show the shape of the NMI in a nominal case. The function shows in that conditions a marked maximum and a smooth shape with no local minima, making the NMI fit for an optimization process in these conditions. Then, the target image underwent an illumination variation to see the effects on the NMI. Here

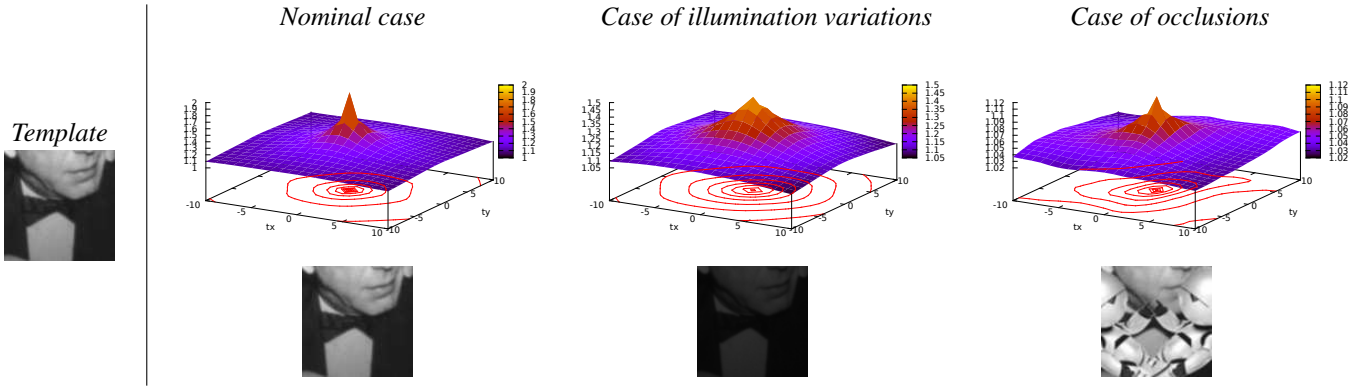

Fig. 1. Shape of the NMI cost function. Below the shapes are shown the compared patches when $tx = 0$ and $ty = 0$ pixel.

again, the function is smooth and possesses a marked maximum when (tx,ty) equals to (0,0). This shows why the NMI can be used even if the light changes in the scene. Finally, the target image was occluded to show the effects of an occlusion on the NMI. The resulting shape is impacted by this large occlusion but even in those conditions no clear local maxima can be seen and the global maxima in (0,0) is still very marked which shows the suitability of the NMI for visual servoing in varying conditions.

2.5 Control law

$\mathbf{e}$ being the task function which needs to be regulated to zero, the control law giving the velocity to apply is defined as in Samson et al. (1991) by:

$$\mathbf{v} = -\lambda \widehat{\mathbf{L_e}}^{+} \mathbf{e}^{\top} \tag{15}$$

where λ is a convergence factor and $\widehat{\mathbf{L_e}}$ is the interaction matrix associated to the task. As the normalized mutual information needs to be maximized, the task is defined as the regulation to zero of the NMI derivative that is nothing but the interaction matrix $\mathbf{L}_{\mathrm{NMI}}$ related to NMI :

$$\mathbf{e} = \mathbf{L}_{\mathrm{NMI}}. \tag{16}$$

The task and the velocity having the same dimension, the control law becomes, as in Dame and Marchand (2011):

$$\mathbf{v} = -\lambda \mathbf{H}_{\mathrm{NMI}}^{-1} \mathbf{L}_{\mathrm{NMI}}^{\top} \tag{17}$$

where $\mathbf{H}_{\mathrm{NMI}}$ is interaction matrix of $\mathbf{L}_{\mathrm{NMI}}$, hence the hessian of the normalized mutual information regarding the position of the camera. To compute those matrices let us express $u(\mathbf{r})$ the numerator and $v(\mathbf{r})$ the denominator of NMI as in equation (9):

$$\begin{aligned} u(\mathbf{r}) &= \mathbf{H}(\mathbf{I}) + \mathbf{H}(\mathbf{I}^*) \\ &= -\sum_{i,j} p_{\mathbf{II}^*}(i,j) \log\left(p_{\mathbf{I}}(i) p_{\mathbf{I}^*}(j)\right) \end{aligned} \tag{18}$$

$$\begin{aligned} v(\mathbf{r}) &= \mathbf{H}(\mathbf{I}, \mathbf{I}^*) \\ &= -\sum_{i,j} p_{\mathbf{II}^*}(i,j) \log\left(p_{\mathbf{II}^*}(i,j)\right) \end{aligned} \tag{19}$$

That being set, $\mathbf{L}_{\mathrm{NMI}}$ and $\mathbf{H}_{\mathrm{NMI}}$ are obtained by chain rules of derivation:

$$\frac{\partial u(\mathbf{r})}{\partial \mathbf{r}} = -\sum_{i,j} \frac{\partial p_{\mathbf{II}^*}(i,j)}{\partial \mathbf{r}} \log(p_{\mathbf{I}}(i) p_{\mathbf{I}^*}(j)) \tag{20}$$

$$\frac{\partial v(\mathbf{r})}{\partial \mathbf{r}} = -\sum_{i,j} \frac{\partial p_{\mathbf{II}^*}(i,j)}{\partial \mathbf{r}} \left(1 + \log(p_{\mathbf{II}^*}(i,j))\right) \tag{21}$$

$$\begin{aligned} \frac{\partial^2 u(\mathbf{r})}{\partial \mathbf{r}^2} = &-\sum_{i,j} \frac{\partial^2 p_{\mathbf{II}^*}(i,j)}{\partial \mathbf{r}^2} \log(p_{\mathbf{I}}(i) p_{\mathbf{I}^*}(j)) \\ &+ \frac{1}{p_{\mathbf{II}^*}(i,j)} \frac{\partial p_{\mathbf{II}^*}(i,j)}{\partial \mathbf{r}}^2 \end{aligned} \tag{22}$$

$$\begin{aligned} \frac{\partial^2 v(\mathbf{r})}{\partial \mathbf{r}^2} = &-\sum_{i,j} \frac{\partial^2 p_{\mathbf{II}^*}(i,j)}{\partial \mathbf{r}^2} \left(1 + \log(p_{\mathbf{II}^*}(i,j))\right) \\ &+ \frac{1}{p_{\mathbf{II}^*}(i,j)} \frac{\partial p_{\mathbf{II}^*}(i,j)}{\partial \mathbf{r}}^2 \end{aligned} \tag{23}$$

With first order derivatives, $\mathbf{L}_{\mathrm{NMI}}$ can be expressed as:

$$\mathbf{L}_{\mathrm{NMI}}(\mathbf{r}) = \frac{\frac{\partial u(\mathbf{r})}{\partial \mathbf{r}} v(\mathbf{r}) - \frac{\partial v(\mathbf{r})}{\partial \mathbf{r}} u(\mathbf{r})}{v(\mathbf{r})^2}. \tag{24}$$

To simplify the expression of $\mathbf{H}_{\mathrm{NMI}}$ let us denote:

$$\alpha(\mathbf{r}) = \frac{\partial u(\mathbf{r})}{\partial \mathbf{r}} v(\mathbf{r}) - u(\mathbf{r}) \frac{\partial v(\mathbf{r})}{\partial \mathbf{r}} \tag{25}$$

$$\beta(\mathbf{r}) = v(\mathbf{r})^2. \tag{26}$$

This gives:

$$\mathbf{H}_{\mathrm{NMI}}(\mathbf{r}) = \frac{\frac{\partial \alpha(\mathbf{r})}{\partial \mathbf{r}} \beta(\mathbf{r}) - \alpha(\mathbf{r}) \frac{\partial \beta(\mathbf{r})}{\partial \mathbf{r}}}{\beta(\mathbf{r})^2} \tag{27}$$

with:

$$\frac{\partial \alpha(\mathbf{r})}{\partial \mathbf{r}} = \frac{\partial^2 u(\mathbf{r})}{\partial \mathbf{r}^2} v(\mathbf{r}) - u(\mathbf{r}) \frac{\partial^2 v(\mathbf{r})}{\partial \mathbf{r}^2} \tag{28}$$

$$\frac{\partial \beta(\mathbf{r})}{\partial \mathbf{r}} = 2 \frac{\partial v(\mathbf{r})}{\partial \mathbf{r}} v(\mathbf{r}). \tag{29}$$

To compute the derivatives of the probabilities, Dame and Marchand (2011) used B-spline functions ϕ to perform histogram binning. The joint probability used in equation (8) therefore becomes:

$$p_{\mathbf{II}^*}(i,j,\mathbf{r}) = \frac{1}{N_{\mathbf{x}}} \sum_{\mathbf{x}} \phi\left(i - \overline{\mathbf{I}}(\mathbf{r},\mathbf{x})\right) \phi\left(j - \overline{\mathbf{I}^*}(\mathbf{x})\right) \tag{30}$$

where $N_{\mathbf{x}}$ is the wanted number of bins in the histogram and $\overline{\mathbf{I}}(\mathbf{r},\mathbf{x})$ and $\overline{\mathbf{I}^*}$ are the images processed to contain only $N_{\mathbf{x}}$ grey levels. Deriving this joint probability yields :

$$\frac{\partial p_{\mathbf{II}^*}(i,j,\mathbf{r})}{\partial \mathbf{r}} = \frac{1}{N_{\mathbf{x}}} \sum_{\mathbf{x}} \frac{\partial \phi}{\partial \mathbf{r}}\left(i - \overline{\mathbf{I}}(\mathbf{r},\mathbf{x})\right) \phi\left(j - \overline{\mathbf{I}^*}(\mathbf{x})\right) \tag{31}$$

$$\frac{\partial^2 p_{\mathbf{II}^*}(i,j,\mathbf{r})}{\partial \mathbf{r}^2} = \frac{1}{N_\mathbf{x}} \sum_\mathbf{x} \frac{\partial^2 \phi}{\partial \mathbf{r}^2} \big(i - \bar{\mathbf{I}}(\mathbf{r},\mathbf{x})\big)\, \phi\big(j - \overline{\mathbf{I}^*}(\mathbf{x})\big) \quad (32)$$

The Jacobian and Hessian of ϕ are finally given by:

$$\frac{\partial \phi}{\partial \mathbf{r}} = -\frac{\partial \phi}{\partial i} \mathbf{L}_{\bar{\mathbf{I}}} \quad (33)$$

$$\frac{\partial^2 \phi}{\partial^2 \mathbf{r}} = \frac{\partial^2 \phi}{\partial^2 i} \mathbf{L}_{\bar{\mathbf{I}}}^\mathbf{T} \mathbf{L}_{\bar{\mathbf{I}}} - \frac{\partial \phi}{\partial i} \mathbf{H}_{\bar{\mathbf{I}}} \quad (34)$$

where $\mathbf{L}_{\bar{\mathbf{I}}}$ and $\mathbf{H}_{\bar{\mathbf{I}}}$ are respectively the interaction matrix and the hessian of $\bar{\mathbf{I}}$. They are given by:

$$\mathbf{L}_{\bar{\mathbf{I}}} = \nabla \bar{\mathbf{I}}\, \mathbf{L}_\mathbf{x} \quad (35)$$

$$\mathbf{H}_{\bar{\mathbf{I}}} = {\mathbf{L}_\mathbf{x}}^T \nabla^2 \bar{\mathbf{I}}\, \mathbf{L}_\mathbf{x} + \nabla \bar{\mathbf{I}}\, \mathbf{H}_\mathbf{x} \quad (36)$$

where $\nabla \bar{\mathbf{I}}$ is the gradient of $\bar{\mathbf{I}}$, $\mathbf{L}_\mathbf{x}$ is the interaction of a point (Chaumette and Hutchinson (2006)) and $\mathbf{H}_\mathbf{x}$ its hessian matrix.

3. CASE OF CENTRAL CAMERAS

Defining $\mathbf{L}_{\bar{\mathbf{I}}}$ requires to compute the image gradient and the definition of the interaction matrix related to a point. To be able to consider a wide range of cameras, the unified model of central projection cameras (Barreto (2001)) is used. With such a projection model there exists various ways to compute these matrices.

3.1 Projection Model

Since the work of Barreto (2001), a unified projection model for central projection cameras was designed. This model describes a family of cameras from perspective to catadioptric ones with particular shape mirrors. Furthermore, Ying and Hu (2004) showed this model could be used for fish-eye lenses.

According to this model, a central projection camera can be modelled by a first projection on a sphere with coordinates $(0,0,\xi)$ in the camera frame followed by a perspective projection on the image plane. Such a model can be defined using parameter ξ which depends intrinsically on the catadioptric camera mirror parameters.

Knowing intrinsic parameters $\gamma = \{p_x, p_y, u_0, v_0, \xi\}$, a 3D point $\mathbf{X} = (X,Y,Z)$ is first projected on a unitary sphere and then in the image plane as $\mathbf{x} = (x,y,1)$. The relationship between $\mathbf{X}$ and $\mathbf{x}$ can be expressed as:

$$\mathbf{x} = pr_\gamma(\mathbf{X}) \text{ with } \begin{cases} x = \dfrac{X}{Z + \xi\sqrt{X^2+Y^2+Z^2}} \\ y = \dfrac{Y}{Z + \xi\sqrt{X^2+Y^2+Z^2}} \end{cases} \quad (37)$$

$\mathbf{x}$ is the point on the virtual normalized plane and the image point in pixelic coordinates is obtained by:

$$\mathbf{u} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} p_x & 0 & u_0 \\ 0 & p_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathbf{K}\mathbf{x} \quad (38)$$

As this paper also deals with methods using data lying on the equivalent sphere, let us define the spherical projection function of a 3D point:

$$\mathbf{X}_\mathcal{S} = pr_\mathcal{S}(\mathbf{X}) \text{ with } \begin{cases} X_\mathcal{S} = \dfrac{X}{\sqrt{X^2+Y^2+Z^2}} \\ Y_\mathcal{S} = \dfrac{Y}{\sqrt{X^2+Y^2+Z^2}} \\ Z_\mathcal{S} = \dfrac{Z}{\sqrt{X^2+Y^2+Z^2}} \end{cases} \quad (39)$$

where $\mathbf{X}_\mathcal{S} = (X_\mathcal{S}\ Y_\mathcal{S}\ Z_\mathcal{S})^T$. On the contrary, the inverse projection function pr_γ^{-1} allows to retrieve the point on the sphere corresponding to the spherical projection of the 3D point, from $\mathbf{x}$:

$$\mathbf{X}_\mathcal{S} = pr_\gamma^{-1}(\mathbf{x}) = \begin{pmatrix} \dfrac{\xi + \sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} x \\ \dfrac{\xi + \sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} y \\ \dfrac{\xi + \sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} - \xi \end{pmatrix} \quad (40)$$

3.2 Image Plane Visual Servoing (IPVS)

Considering image plane visual servoing (*ie* considering the point $\mathbf{x}$ on the image plane), gradients computation are the same as with perspective cameras. The interaction matrix however changes. It was shown in Espiau et al. (1992) that the interaction matrix can be expressed as the product of two Jacobians:

$$\mathbf{L}_\mathbf{x} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \frac{\partial \mathbf{X}}{\partial \mathbf{r}} \quad (41)$$

where $\frac{\partial \mathbf{x}}{\partial \mathbf{X}}$ represents the movement of a point in the image with relation to the corresponding point in 3D and $\frac{\partial \mathbf{X}}{\partial \mathbf{r}}$ the movement of the 3D point with relation to the camera pose. The advantage of this formulation is that the second term of the product does not depend on the type of projection and therefore only the first part needs to be redefined. This is done by deriving (37). The interaction matrix is eventually expressed as in Barreto (2001).

3.3 Cartesian Spherical Visual Servoing (CSVS)

As described in section 3.1, the projection model first projects the points on a sphere before projecting them again, this time on the image plane. To compute gradients adapted to the omnidirectional image shape, one can therefore decide to compute them based on $\mathbf{X}_S$, the projection of $\mathbf{X}$ on the sphere. With this representation, $\mathbf{L}_{\mathbf{X}_S}$ is defined as in Hamel (2002):

$$\mathbf{L}_{\mathbf{X}_\mathbf{S}} = \frac{\partial \mathbf{X}_S}{\partial \mathbf{X}} \frac{\partial \mathbf{X}}{\partial \mathbf{r}} = \left(\frac{1}{\rho} (\mathbf{X}_\mathcal{S} \mathbf{X}_\mathcal{S}^T - \mathbf{I}_3)\ [\mathbf{X}_\mathcal{S}]_\times \right) \quad (42)$$

where $[\mathbf{X}_\mathcal{S}]_\times$ is the skew matrix of the vector $\mathbf{X}_\mathcal{S}$.

3.4 Gradient computation

To evaluate the gradients which are necessary to compute (35) and (36) for CSVS, it was chosen to act as proposed in Caron et al. (2010). A sample step is first defined. Then each point is projected from the image to the sphere and the sampling is done to determinate its neighbors on the sphere. Those point are finally projected in the image to determinate the neighborhood to be used when computing the gradient. This step only has to be done once for the whole process and can therefore be done beforehand. As it provides real values, a bilinear interpolation was used to compute values at the neighbors' locations.

4. EXPERIMENTAL RESULTS

Several experiments were performed to validate the method. They were realized on a robot with six degrees of freedom with a fisheye camera mounted on its end effector. The camera provided 320x240 input images. The test program was implemented using the ViSP (Marchand et al. (2005)) library. Three experiments are exposed in this section. The first one shows

how the proposed method manages to complete a positioning task and compares the two modelizations seen in section 3. The second and third one respectively study the effects of illumination variations and of occlusions on the results of the positioning task. The initial and final position difference introduces a translation along X axes and a rotation along Y axes typically leading to projection ambiguities and a consequent translation along Y axes making it difficult for a servoing task to perform well. During the experiments, interaction matrix is computed at each iteration. Since it is an unknown parameter, computations are made assuming a constant Z for all the pixels. An adaptive gain was used to enhance speed when the camera is at the limits of its convergence cone. For readability purposes, graphics of NMI were created depicting the evolution of (NMI-1) instead of NMI, giving a measure between 0 and 1.

4.1 Experiment 1

The aim of this experiment is to compare the two representation methods exposed in section 3. To do so, both tasks were launched from the same sets of positions and the results of the servoing tasks were monitored (see figures 2 to 5). Let us first note, that in both cases, the positioning task is correctly achieved. Cost function and trajectory are more noisy with IPVS which shows an evolution of the NMI shaky when close from convergence. This is mainly due to the fact that more approximations are done in the interaction matrix computation (especially in the image gradient computation, see Caron et al. (2010)). In both cases repositioning precision was good, with final errors being around 0.1 millimeter.

Different experiments also showed CSVS to converge from further positions and being less affected by the modification of the scene or the depth approximation, therefore it is this method that will be used to demonstrate the robustness of the servoing task towards variations of the scene. The evolution of NMI on figure 4 is interesting as it can be related to the shape of the NMI function (see figure 1). Indeed, when far from the maximum, the function is rather monotonous and does not grow rapidly which makes the task advance slowly. On the other hand, when it comes close to the maximum the evolution becomes quicker and the NMI grows rapidly as it can be seen here around iteration 300. Also, the value of NMI does not reach its theoretical maximum of 2 (in the case of the evolution of (NMI-1) it should converge to 1). This is mainly due to the fact that to perform histogram binning B-spline functions were used, impacting the estimated probability distribution functions.

4.2 Experiment 2: light variations

The goal of this experiment is to assess the robustness of the method towards illumination changes. This was done using the CSVS approach and changing illumination conditions in several places of the room thanks to various spot lights. Even though the final positioning precision drops a little, averaging around 1 to 3 millimeters, the task is still successful and the robot stabilizes at the desired position even with distant starting points. The trajectory is moderately impacted, as the overall course of the camera stays the same even though small variations appear. Figure 6 also shows the evolution of the cost function and positioning errors which are very similar to the nominal case, which shows that the task is almost not impacted by the changes of conditions.

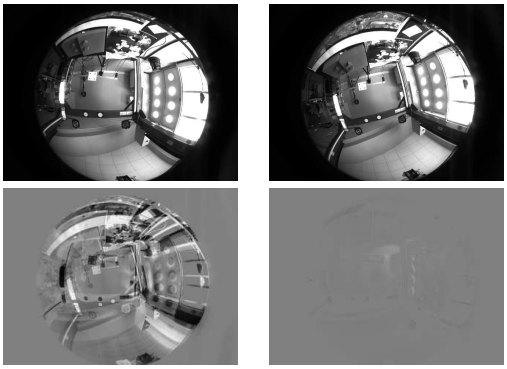

Fig. 2. On the upper line are the initial and desired images, respectively on the left and on the right. On the lower line are the initial and final error images, respectively on the left and on the right.

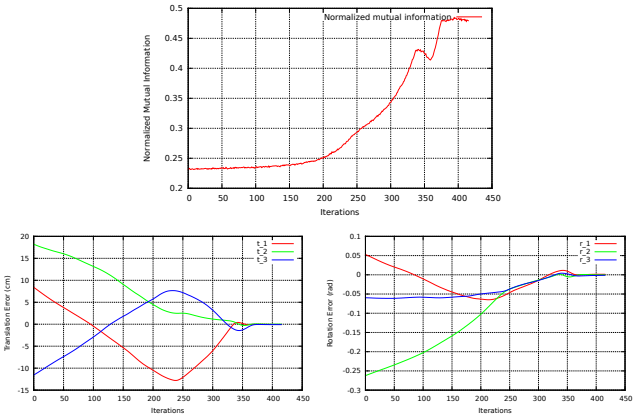

Fig. 3. CSVS method. On the upper graph is represented the evolution of normalized mutual information. The lower ones show the positioning error on the translation d.o.f (left graph) and on the rotation d.o.f (right graph).

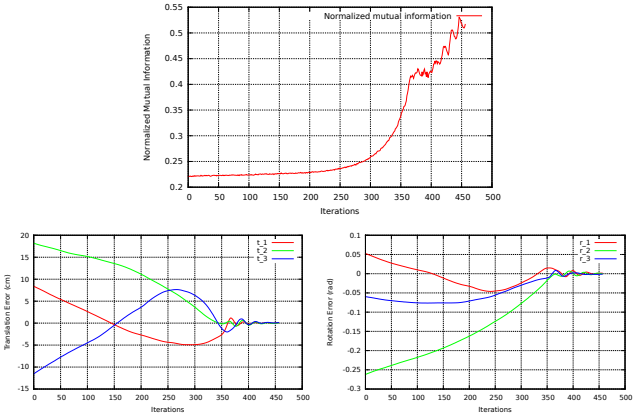

Fig. 4. IPVS method. On the upper graph is represented the evolution of normalized mutual information. The lower ones show the positioning error on the translation d.o.f (left graph) and on the rotation d.o.f (right graph).

4.3 Experiment 3: impact of occlusions

This experiment was also designed to evaluate the robustness of the method, but this time with respect to occlusions (see fig 7). In order to do that, several objects were moved and people walked in the sight of the camera during the servoing task. The task succeeds and positioning precision averages around 0.5 to 2 millimeters. Here again the camera stabilizes with an acceptable final error and the trajectory is very similar to the

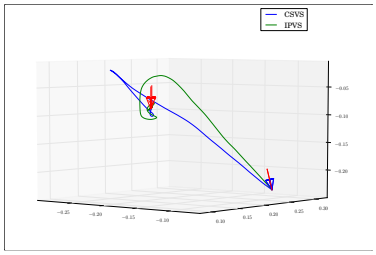

Fig. 5. Trajectory of the camera in the 3D world. Blue curve represents the CSVS method and green curve the IPVS method.

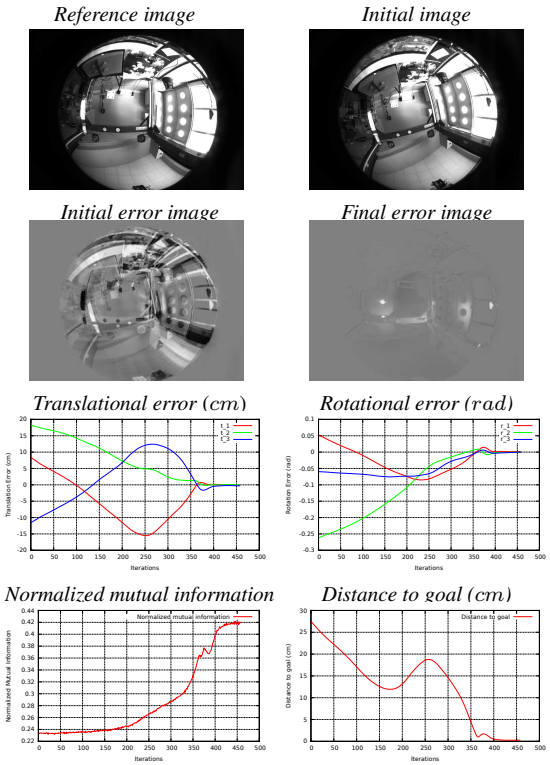

Fig. 6. Omnidirectional visual servoing using normalized mutual information: impact of illumination changes.

nominal case, which shows that the task is almost not impacted by occlusions.

5. CONCLUSION

In this paper, a new way of achieving visual servoing using the information theory was presented. The method was detailed on an omnidirectional camera and different approaches taking into consideration the resulting projection model were detailed. Experiments were realized on a cartesian robot with six degrees of freedom and robustness towards occlusions and illumination variations were demonstrated. When compared to MI, NMI shows similar results but gives better readability of its results since the upper bound sets an identifiable goal to reach.

REFERENCES

Barreto, J.P. Araujo, H. (2001). Issues on the geometry of central catadioptric images. In *Int. Conf. on Computer Vision and Pattern Recognition*. Hawai, USA.

Caron, G., Marchand, E., and Mouaddib, E. (2010). Omnidirectional photometric visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'10*, 6202–6207. Taipei, Taiwan.

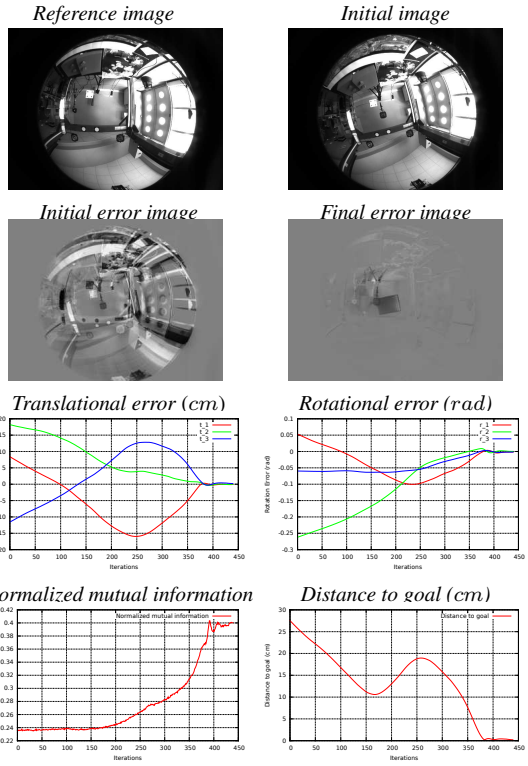

Fig. 7. Omnidirectional visual servoing using normalized mutual information: impact of occlusions.

Chaumette, F. and Hutchinson, S. (2006). Visual servo control, Part I: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4), 82–90.

Chesi, G. and Hashimoto, K. (eds.) (2010). *Visual Servoing via Advanced Numerical Methods*. Springer.

Collewet, C. and Marchand, E. (2011). Photometric visual servoing. *IEEE Trans. on Robotics*, 27(4), 828–834.

Dame, A. and Marchand, E. (2011). Mutual information-based visual servoing. *IEEE Trans. on Robotics*, 27(5).

Deguchi, K. (2000). 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.

Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3), 313–326.

Hamel, T. Mahony, R. (2002). Visual servoing of an under-actuated dynamic rigid-body system: An image-based approach. *IEEE Int. Trans. on Robotics and Automation*, Vol. 18, No. 2.

Hutchinson, S., Hager, G., and Corke, P. (1996). A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5), 651–670.

Kallem, V., Dewan, M., Swensen, J., Hager, G., and Cowan, N. (2007). Kernel-based visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and System, IROS'07*, 1975–1980. San Diego, USA.

Marchand, E., Spindler, F., and Chaumette, F. (2005). ViSP for visual servoing: a generic software platform with a wide class of robot control skills. *IEEE Robotics and Automation Magazine*, 12(4), 40–52.

Nayar, S., Nene, S., and Murase, H. (1996). Subspace methods for robot vision. *IEEE Trans. on Robotics*, 12(5), 750 – 758.

Samson, C., Espiau, B., and Borgne, M.L. (1991). *Robot Control: The Task Function Approach*. Oxford University Press.

Shannon, C.E. (1948). A mathematical theory of communication. *Bell system technical journal*, 27.

Studholme, C. Hill, D. and Hawkes, D. (1999). An overlap invariant entropy mesure of 3d medical image alignment. *Pattern Recognition*, 32(99), 71–86.

Ying and Hu (2004). Can We Consider Central Catadioptric Cameras and Fish-eye Cameras within a Unified Imaging Model ? In *European Conference on Computer Vision*, volume 1. Prague, Czech.