Histograms-based Visual Servoing

Quentin Bateux, Eric Marchand

Histograms-based Visual Servoing

Quentin Bateux[1] and Eric Marchand[1]

***Abstract*—In the direct visual servoing methods such as photometric framework, the images as a whole are used to define a control law. This can be opposed to the classical visual servoing approaches that relies on geometric features and where image processing algorithms that extract and track visual features are necessary. In this paper, we propose a generic framework to consider histogram as a visual feature. A histogram is an estimate of the probability distribution of a variable (for example the probability of occurrence in an intensity, color, or gradient orientation in an image). We show that the framework we proposed applies, but is not limited to, a wide set of histograms and allows the definition of efficient control laws. Statistical comparisons are presented from simulation results and real robots experiments including navigation tasks are also provided.**

***Index Terms*—Visual Servoing; Visual-Based Navigation**

I. Introduction

VISUAL servoing techniques aim to control a dynamic system by using visual features extracted in images provided by one or multiple cameras. This approach is extensively presented in [12][3]. A positioning task is achieved by regulating to zero an error in the image space. This error is usually computed from visual features related to geometric features extracted and tracked by computer vision algorithms and by comparing the displacement of matching features in both the current image and a target image. In order to avoid the bottleneck represented by the tracking process [15], direct visual servoing [5] was introduced and various features and cost functions were then studied.

The original direct visual servoing approach, photometric visual servoing [5][4], was designed by taking as an error the direct subtraction pixel-wise between the current and desired image making it a purely photometric approach (image intensities only). Due to a pure pixel-wise comparison, photometric visual servoing is very precise but does not react well in situations where global illumination is less controlled or in presence of occlusions, which may impact the final precision of the positioning. This is why various strategies and refinements were considered to improve convergence, robustness and precision of the positioning tasks. [9] used a template illumination adaptation in order to adapt the current image so that its global illumination matches the illumination conditions of the desired one through the use of a histogram adaptation process (as proposed for tracking in [18]). [8] changed the cost function by computing the mutual information (MI) between the two images. The resulting control law maximizes the shared information (which is actually an entropy measure) between current and desired images. In a recent work [1], we proposed to consider the intensity histogram of an image as the visual feature. The control is then derived by minimizing a distance between the two histograms computed on both images. This approach was validated by controlling a 6 degrees of freedom (DoF) robot. However, intensity histograms have serious drawbacks when it comes to global illumination changes. Control laws based on more elaborate histograms could be able to overcome some of these issues. In this paper, we thus extend this idea and propose a general framework able to consider any kind of histograms. We show that this general framework applies easily to, e.g., intensity histograms (as in [1]), histograms of oriented gradients (HOG [7]) and color histograms [17].

In many computer vision fields such as tracking or image retrieval, histograms as global descriptors have been used successfully. One major example is the Mean Shift algorithm presented in [6] where a color histogram is used to track an object by iteratively converging toward the minima of distance between two histograms. Other examples exist in the image retrieval field [16] to describe globally an image, to detect pedestrians [7] or to detect and match key-points such as in the well-known SIFT descriptor [14] where orientation histograms are considered. Dealing with our visual servoing and to illustrate the generality of the proposed approach, we select here two kinds of histograms that are likely to possess interesting properties in terms of robustness, namely the HOG histogram and the Hue-Saturation histogram.

This paper has the following structure. The next section recalls the basics of visual servoing. Section 3 presents the generic framework used to generate control laws from a general expression of a histogram; Sections 4 details the advantages of the chosen histograms and then the computation of their associated control laws from the presented framework. Section 5 compares the methods through their convergence areas, and then provides an experimental validation of each of the selected methods, performed on a 6 DoF robot for a positioning task and a 1 DoF robot for a visual path following task.

II. Visual servoing

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 that reflects, in the image space, this positioning error and that needs to be minimized. Considering the actual pose of

Manuscript received: 08, 31, 2015; Revised 12, 18, 2015; Accepted 01, 23, 2016.

This paper was recommended for publication by Editor Jana Kosecka upon evaluation of the Associate Editor and Reviewers' comments.

[1]Quentin Bateux and Eric Marchand are with Université de Rennes 1, IRISA, Inria Rennes, Lagadic team, France. quentin.bateux@irisa.freric.marchand@irisa.fr

the camera $\mathbf{r}$ the problem can therefore be written as an optimization process:

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

where $\rho(.)$ is the error function between the desired pose and the current one, and $\widehat{\mathbf{r}}$ the pose reached after the optimization process (servoing process), which 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 typically has to minimize the difference between **s(r)** and the desired configuration **s*** which, given that a suitable distance between the two feature sets can be defined, leads to:

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

This visual servoing task is achieved by iteratively applying a velocity to the camera. This requires the knowledge of the interaction matrix $\mathbf{L_s}$ of **s(r)** that links the variation of $\dot{\mathbf{s}}$ to the camera velocity and which is defined as $\dot{\mathbf{s}}(\mathbf{r}) = \mathbf{L_s}\mathbf{v}$ where $\mathbf{v}$ is the camera velocity. This equation leads to the expression of the velocity that needs to be applied to the robot. The control law is classically given by:

$$\mathbf{v} = -\lambda \mathbf{L_s}^+ (\mathbf{s}(\mathbf{r}) - \mathbf{s}^*) \tag{3}$$

where λ is a positive scalar gain defining the convergence speed of the control law and $\mathbf{L_s}^+$ is the pseudo-inverse of the interaction matrix. To avoid extraction and tracking over frame of geometrical features (such as points, lines, etc), in [5][4] the notion of direct (or photometric) visual servoing where the visual feature is the image considered as a whole has been introduced. In this case the feature $\mathbf{s}$ becomes the image itself ($\mathbf{s}(\mathbf{r}) = \mathbf{I}(\mathbf{r})$). This means that the optimization process becomes [4]:

$$\widehat{\mathbf{r}} = \arg\min_{\mathbf{r}} ((\mathbf{I}(\mathbf{r}) - \mathbf{I}^*)^\top (\mathbf{I}(\mathbf{r}) - \mathbf{I}^*)) \tag{4}$$

where $\mathbf{I}(\mathbf{r})$ and $\mathbf{I}^*$ are respectively the image seen at the position $\mathbf{r}$ and the template image (both of N pixels). Each image is represented as a single column vector. The control law is inspired by the Levenberg-Marquardt (LM) optimization approach. It is given by (as in [4]):

$$\mathbf{v} = -\lambda (\mathbf{H_I} + \mu \; diag(\mathbf{H_I}))^{-1} \mathbf{L_I}^\top (\mathbf{I}(\mathbf{r}) - \mathbf{I}^*) \tag{5}$$

where μ is the Levenberg-Marquardt parameter which is positive scalar that defines the behavior of the control law, from a steepest gradient behavior (μ small) to a Gauss-Newton's behavior (μ large). Where $\mathbf{H_I} = \mathbf{L_I}^\top \mathbf{L_I}$ and $\mathbf{L_I}$ is the interaction matrix that can be expressed as:

$$\mathbf{L_I} = -\nabla \mathbf{I}^\top \mathbf{L_x} \tag{6}$$

and $\mathbf{L_x}$ is the interaction matrix of a point (see details in [4]).

Following the idea of direct visual servoing that avoids features extraction, we shall consider in the reminder of this paper new and more compact features: histograms. In [1], we have considered intensity histograms. In the next section we first consider the theoretical foundation of histogram based visual servoing and show how it is able to consider various kinds of histograms: intensity, color histograms or histogram of oriented gradients (HOG), in fact any object that is based on the estimation of a probability distribution obtained from pixel values.

III. Histogram-based visual servoing (HBVS)

A histogram is an estimate of the probability distribution of a variable. This distribution describes in a synthetic way this variable, and it can be applied to any kind of object, in our case, a digital image with its pixel intensities or alternately color, orientation, norm of the gradients, etc. Histograms can then be used to compare images with a limited quantity of information, as opposed to the direct pixel-wise subtraction (SSD). Furthermore, this reduction of the information quantity also induces more robustness to local variations, and several kinds of values can be used to compute the histogram. For example by taking the gradient orientation of each pixel instead of its intensity value, the comparison can be resilient towards global illumination variations. This very flexible nature makes the histogram a very interesting candidate to perform global direct visual servoing. It is also important to point out that histograms in their most basic form do not capture spatial information, and therefore a strategy to make histogram-based methods spatial-sensitive will be presented. In this section, we show how generic histograms can be considered to build a visual control law able to achieve 6 DoF positioning task.

A. Theoretical foundations of HBVS

In this paragraph, we recall the formal definition of histograms using probability distribution and the classical distances that are used to compare two histograms.

1) Histogram: definition and overview: A histogram is a statistical object that associates to each pixel information in an image a given 'bin' in the histogram. This pixel value can be a scalar, such as in the intensity histogram and the HOG histogram, or it can be a vector if the image is composed of multiple planes, as in the case of color histograms. This histogram thus represents the statistical distribution of the pixel values in the image. For example, the intensity histogram is expressed as:

$$\mathbf{p_I}(i) = \frac{1}{N_\mathbf{x}} \sum_{\mathbf{x}}^{N_\mathbf{x}} \delta(\mathbf{I}(\mathbf{x}) - i) \tag{7}$$

where $\mathbf{x}$ is a 2D pixel position in the image plane, the pixel intensity $i \in [0, 255]$ if we use images with 256 gray-levels, $N_\mathbf{x}$ the number of pixels in the image $\mathbf{I}(\mathbf{x})$, and $\delta(\mathbf{I}(\mathbf{x}) - i)$ the Kronecker's function defined such as:

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$\mathbf{p_I}(i)$ is nothing but the probability for a pixel of the image $\mathbf{I}$ to have the intensity i. Other kinds of histograms built from color or HOG will be defined in section IV.

2) Distance between histograms: Visual servoing is the determination of the motion that allows a dynamic system to minimize an error between a current view and a desired view of a scene. In order to perform this operation, it is mandatory to be able to compare these two views by defining a distance $\rho(.)$ between them. Since, in our case, images are

described as histograms, we need to define a suitable distance to compare them. Classically, histograms are compared bin-wise. One formulation of this method is the Matusita distance and is expressed such as [2]:

$$\rho(\mathbf{I},\mathbf{I}^*) = \sum_{i}^{N_c} \left(\sqrt{\mathbf{p_I}(i)} - \sqrt{\mathbf{p_{I^*}}(i)}\right)^2 \tag{9}$$

where N_c is the number of bins considered in the histograms. Unlike the Bhattacharyya measure that increases as the overlap of the two compared vectors increases, the Matusita distance (based on the Bhattacharyya measure, see [2] for further details on these measures) is null when the two compared histograms are similar, which fits better our minimization scheme.

B. Interaction matrix and resulting control law

By definition, the interaction matrix links the variation of the visual features to the camera motion [3]. This definition leads to the generic expression of the interaction matrix, $\mathbf{L} = \frac{\partial \boldsymbol{s}}{\partial \mathbf{r}}$, where $\mathbf{r}$ is the camera position. In our context, it links the variation of the cost function $\rho(.)$ to the camera motion. It is then defined by:

$$\mathbf{L}_H = \frac{\partial \rho(\mathbf{I},\mathbf{I}^*)}{\partial \mathbf{r}} \tag{10}$$

where $\rho(\mathbf{I},\mathbf{I}^*)$ obviously depends on the nature of the histogram considered; each being defined by its probability density descriptor $\mathbf{p_I}(i)$. We can then develop this expression and obtain:

$$\begin{aligned}\frac{\partial \rho(\mathbf{I},\mathbf{I}^*)}{\partial \mathbf{r}} &= \frac{\partial}{\partial \mathbf{r}}\left[\sum_{i}^{N_c}\left(\sqrt{\boldsymbol{p}_\mathbf{I}(i)} - \sqrt{\boldsymbol{p}_{\mathbf{I}^*}(i)}\right)^2\right] \\ &= 2N_c \sum_{i}^{N_c}\left(\frac{\partial}{\partial \mathbf{r}}\sqrt{\boldsymbol{p}_\mathbf{I}(i)}\left(\sqrt{\boldsymbol{p}_\mathbf{I}(i)} - \sqrt{\boldsymbol{p}_{\mathbf{I}^*}(i)}\right)\right)\end{aligned} \tag{11}$$

with

$$\frac{\partial}{\partial \mathbf{r}}\sqrt{\boldsymbol{p}_\mathbf{I}(i)} = \frac{1}{2\sqrt{\boldsymbol{p}_\mathbf{I}(i)}}\frac{\partial \boldsymbol{p}_\mathbf{I}(i)}{\partial \mathbf{r}} \tag{12}$$

The expression then becomes:

$$\frac{\partial \rho(\mathbf{I},\mathbf{I}^*)}{\partial \mathbf{r}} = 2N_c \sum_{i}^{N_c}\left(\frac{\partial \boldsymbol{p}_\mathbf{I}(i)}{\partial \mathbf{r}}\left(\frac{\sqrt{\boldsymbol{p}_\mathbf{I}(i)} - \sqrt{\boldsymbol{p}_{\mathbf{I}^*}(i)}}{2\sqrt{\boldsymbol{p}_\mathbf{I}(i)}}\right)\right) \tag{13}$$

Finally, we get this generic expression of the interaction matrix required to design histogram-based control laws:

$$\mathbf{L}_H = N_c \sum_{i}^{N_c}\left(\frac{\partial \boldsymbol{p}_\mathbf{I}(i)}{\partial \mathbf{r}}\left(1 - \frac{\sqrt{\boldsymbol{p}_{\mathbf{I}^*}(i)}}{\sqrt{\boldsymbol{p}_\mathbf{I}(i)}}\right)\right) \tag{14}$$

This 1×6 expression can then be used inside a minimization scheme, such as a non-linear minimization like a Levenberg-Marquardt to form the following control law:

$$\mathbf{v} = -\lambda(\mathbf{H}_H + \mu\ diag(\mathbf{H}_H))^{-1}\mathbf{L}_H^\top \rho(\mathbf{I},\mathbf{I}^*) \tag{15}$$

where λ and μ are positive scalars and $\mathbf{H}_H = \mathbf{L}_H^\top \mathbf{L}_H$. It is crucial to note that $\mathbf{L}_H$ being of size 1×6, it allows us to control only 1 DoF when integrated into a control scheme because the minimization problem is under-constrained. This poses an important issue since our goal is to control at least 6 DoF. Therefore an extension to this method had to be developed to in order extend the control to more DoF. This is presented in Section III-D.

C. Introducing second order B-Spline

From the definition of the interaction matrix in equation (14), the Kronecker's function ($\delta(.)$ used in equation (7)) poses an issue because of its non-differentiability. In order to compute the derivative of the error function (the Matusita distance given in equation (9)) defined by the difference of histograms, as in [8], our histograms are smoothed as they are computed using a second-order B-Splines (derivable once) in order to approximate and smooth our bins. By applying this method the expression of the intensity histogram would for example become:

$$\mathbf{p_I}(i) = \frac{1}{N_x}\sum_{\mathbf{x}}^{N_\mathbf{x}} \phi(\mathbf{I}(\mathbf{x}) - i) \tag{16}$$

where $N_\mathbf{x}$ is the number of pixels, $\mathbf{x} = \{x, y\}$ a single image pixel, $\phi(.)$ a B-spline differentiable once and $\mathbf{I}$ the image reduced to N_c intensity values (each histogram then contains N_c bins).

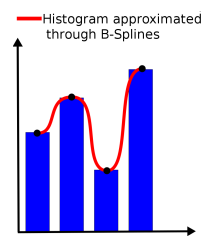

Figure 1. Smoothing and approximating a histogram by using a B-spline

D. Towards 6 DoF control

As expected from previous works in computer vision using intensity histograms distances [6], processing a single histogram on the whole image fails to have sensibility to more than translations on the x/y axis. In the tracking field, [11] proposes to extend the sensibility of histogram-based method to more DoF through the use of multiple histograms throughout the image. Here we adapt this idea by equally dividing the image in multiple areas and by associating a histogram to each area. It is necessary to stack the resulting error vectors and interaction matrices. The global interaction matrix then becomes:

$$\mathbf{L}_H = \begin{bmatrix}\mathbf{L}_{H_1} & \mathbf{L}_{H_2} & \dots & \mathbf{L}_{H_n}\end{bmatrix}^\top \tag{17}$$

where $\mathbf{L}_{H_i}$ is the interaction matrix given by the i-th histogram distance, using the control law defined in the previous section. $\mathbf{L}_H$ is now of size $n \times 6$ and the cost term $\rho(\mathbf{I},\mathbf{I}^*)$ in equation 15 becomes a vector of size $n \times 1$. By using 6 or more histograms on the image, it then becomes possible to control every six degrees of freedom of the robot, the minimization problem of equation 15 no longer being under-constrained.

After several experiments, separating the image evenly in 36 sub-parts proved to be an adequate choice. Indeed, less areas reduce the decoupling of the DoF and more increase the sensibility to outliers in the scene. Using such approach, the cost function features a clear minimum, with a rather large valley. Concerning the convexity of this cost function, experiments showed that keeping around 60 percents of shared pixels between areas in the current image and desired areas seems to ensure belonging to the convex area of the cost function (for a suitable number of histogram bins). The choice of this parameter being purely empirical.

IV. Defining more robust histogram-based control laws

Intensity histograms in visual servoing have been presented in [1]. As stated, this paper generalizes this approach and shows that it can also apply to color histograms and histograms of oriented gradients which have proved to be very efficient image descriptors in computer vision.

A. Using color histograms

Since the intensity image is basically a reduced form of the color image, it makes sense to experiment with a richer descriptor which is the color histogram. This descriptor has been widely used in several computer vision domains, such as indexing [19][13] or tracking [6][17]. The main interest of using color histograms over simple intensity histograms is that they capture more information and can therefore prove to be more reliable and versatile in some conditions. They also share the lightness property of the intensity histogram which allows this family of descriptors to be applied to a wide range of real-time applications. Our goal here is to tackle the global illumination sensitivity which represents a serious drawback when considering the intensity histogram-based visual servoing method. This issue greatly restricts the field of application of this method. The idea here is to abstract our method from the intensity of the pixels to rely solely on the colorimetric information which is more robust to global changes in illumination.

1) Cost function: In order to reduce the sensibility of the method regarding to the global illumination variation, we choose here to focus more on the color information and to set aside the intensity part of the image as proposed in [17]. In order to do that, we perform a change in the color space, from RGB (Red/Green/Blue) to HSV (Hue/Saturation/Value) which separates the color information (Hue and Saturation) from the pure intensity (Value). Since in this case the histogram has to be computed not only from one but two image planes, the expression becomes:

$$\boldsymbol{p_I}(i) = \sum_{x}^{N_x} \left(\phi(\mathbf{I}_H(\mathbf{x})\mathbf{I}_S(\mathbf{x}) - i)\right) \tag{18}$$

where $\mathbf{I}_H(\mathbf{x})$ and $\mathbf{I}_S(\mathbf{x})$ are respectively the Hue and Saturation image planes of the HSV color image from the camera. A second way of integrating color information in an image is to synthesize an intensity image by selecting as a source the two color channels of the HSV original image (H and S) and to sum them in one plane. By doing that, we can use the original control law that applies to intensity histograms and still alleviate the sensibility to illumination changes in the image, provided that the light source can be considered as a white light source. In that case the visual servoing is very similar to the one proposed in [1]. This latter approach is reported in section IV-A3

2) Interaction matrix for color histogram: From the expression of the histogram-based interaction matrix (14), we only need to compute the derivative of expression (18) to obtain an expression that can be computed directly from the image values. To do so, we apply derivation chain rules:

$$\frac{\partial}{\partial \boldsymbol{r}}\left(\boldsymbol{p_I}(i)\right) = \sum_{\mathrm{x}}^{N_{\mathrm{x}}} \left(\frac{\partial}{\partial \boldsymbol{r}}\left(\phi(\mathbf{I}_H(\mathbf{x})\mathbf{I}_S(\mathbf{x}) - i)\right)\right) \tag{19}$$

As in [5], from the Optical Flow Constraint Equation (OFCE), which imply assuming that we are working with Lambertian surfaces only, we can obtain the following equation:

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{r}}\left(\phi(\mathbf{I}_H(\mathbf{x})\mathbf{I}_S(\mathbf{x}) - i)\right) \\
&= \frac{\partial}{\partial i}\left(\phi(\mathbf{I}_H(\mathbf{x})\mathbf{I}_S(\mathbf{x}) - i)\right) \frac{\partial}{\partial \boldsymbol{r}}\left(\mathbf{I}_H(\mathbf{x})\mathbf{I}_S(\mathbf{x})\right) \\
&= \frac{\partial}{\partial i}\left(\phi(\mathbf{I}_H(\mathbf{x})\mathbf{I}_S(\mathbf{x}) - i)\right) \\
&\qquad \left(\frac{\partial}{\partial \boldsymbol{r}}\left(\mathbf{I}_H(\mathbf{x})\right)\mathbf{I}_S(\mathbf{x}) + \mathbf{I}_H(\mathbf{x})\frac{\partial}{\partial \boldsymbol{r}}\left(\mathbf{I}_S(\mathbf{x})\right)\right) \\
&= \frac{\partial}{\partial i}\left(\phi(\mathbf{I}_H(\mathbf{x})\mathbf{I}_S(\mathbf{x}) - i)\right)\left(\mathbf{L_H}\mathbf{I}_S(\mathbf{x}) + \mathbf{I}_H(\mathbf{x})\mathbf{L_S}\right)
\end{aligned} \tag{20}$$

where $\mathbf{L_H}$ and $\mathbf{L_S}$ are (from [5]):

$$\begin{aligned}
\mathbf{L_H} &= -(\nabla_x \mathbf{I_H}(\mathbf{x})\mathbf{L}_x + \nabla_y \mathbf{I_H}(\mathbf{x})\mathbf{L}_y) \\
\mathbf{L_S} &= -(\nabla_x \mathbf{I_S}(\mathbf{x})\mathbf{L}_x + \nabla_y \mathbf{I_S}(\mathbf{x})\mathbf{L}_y)
\end{aligned} \tag{21}$$

where $\mathbf{L}_{\mathrm{x}}$ and $\mathbf{L}_{\mathrm{y}}$ are the lines corresponding to the x and y-coordinates of $\mathbf{L_x}$ the interaction matrix related to a single 2D point $\mathbf{x} = (x, y)$ in the image, and is defined in [3] as:

$$\mathbf{L_x} = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix} \tag{22}$$

where Z is the depth between the camera and a projected pixel in the scene. In all our experiments as we do not wish to require additional equipment to measure depth, we will make the assumption that this value is constant and set as a rough estimation of the final camera-scene distance for the current experiment. Finally:

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{r}}\left(\boldsymbol{p_I}(i)\right) = \\
&\sum_{\mathrm{x}}^{N_{\mathrm{x}}} \left(\frac{\partial}{\partial i}\left(\phi(\mathbf{I_H}(\mathbf{x})\mathbf{I_S}(\mathbf{x}) - i)\right)\left(\mathbf{L_H}(\mathbf{x})\mathbf{I_S}(\mathbf{x}) + \mathbf{I_H}(\mathbf{x})\mathbf{L_S}(\mathbf{x})\right)\right)
\end{aligned} \tag{23}$$

3) Integrating hue and saturation information into a single histogram: In order to keep complexity low, an alternative method to consider color is to artificially create a single intensity plane that is based on color information instead of the intensity information. The control law associated to this method would be the same as the one used for intensity histogram-based visual servoing [1]. Nevertheless it

gains interesting properties in terms of increased robustness regarding global illumination changes, without the increase in computational cost induced by the more convoluted control law associated to the previous method. To do so, we exploit the properties of the HSV color-space as in the previous part, and we exclude the Value component from the image. Following the same reasoning as in the design of the HS histogram, we choose here to use only the Hue and Saturation information, by creating the synthetic image following this protocol [20]:

- converting the RGB input image into HSV color space;
- computing the new plane values such as: $I_{HS-Synth} = \frac{I_H(\mathbf{x})+I_S(\mathbf{x})}{2}$;
- normalizing the new plane values between 0 and 255.

We then input this intensity image into the regular pipeline of the intensity histogram-based control law.

B. Histogram of Oriented Gradients

The histogram of oriented gradients (HOG) differs from the intensity histogram in that it does not directly classify the values of the raw image pixels but relies on the computation of the orientation of the gradient for each pixel. HOG has been introduced first by [7] in order to define a new descriptor to detect humans in images. This descriptor proved to be very powerful and is now widely used for recognition purposes such as human faces [10]. Still, in order to increase robustness of our method regarding illumination changes, relying solely on gradients can prove to be very beneficial since gradient orientation is invariant to illumination changes, either global or local.

1) Cost function: For a pixel $\mathbf{x}$, gradient orientation is defined as:

$$\theta(\mathbf{I}(\mathbf{x})) = \text{atan}\left(\frac{\nabla_y \mathbf{I}(\mathbf{x})}{\nabla_x \mathbf{I}(\mathbf{x})}\right) \tag{24}$$

where $\nabla_x \mathbf{I}(\mathbf{x})$ and $\nabla_y \mathbf{I}(\mathbf{x})$ are respectively the horizontal and vertical gradient values. The histogram of oriented gradient is then defined, as in the previous case by:

$$\boldsymbol{p}(i) = \frac{1}{N_x}\sum_{\mathbf{x}}^{N_x} \parallel \nabla \mathbf{I}(\mathbf{x}) \parallel \phi\left(\theta(\mathbf{I}(\mathbf{x})) - i\right) \tag{25}$$

where ϕ is a B-spline of at least second order and $\parallel \nabla \mathbf{I}(\mathbf{x}) \parallel$ is the norm of the gradient, weighting out the contribution of the pixels of weaker gradient that are more prone to possess a less defined information (for example an uniform texture with some image noise will be a weak, randomly oriented gradient field). One major advantage of working with HOG is that this descriptor is not sensitive to variations of the global illumination.

2) Interaction matrix for HOG: In the same way as the previous color histogram method, we start directly by computing the derivative of the cost function (25)

$$\begin{aligned}\frac{\partial \boldsymbol{p}_{\mathbf{I}}(i)}{\partial \mathbf{r}} &= \frac{\partial}{\partial \mathbf{r}}\left(\frac{1}{N_x}\sum_{\mathbf{x}}^{N_x} \parallel \nabla \mathbf{I}(\mathbf{x}) \parallel \phi\left(\theta(\mathbf{I}(\mathbf{x})) - i\right)\right) \\ &= \frac{1}{N_x}\sum_{\mathbf{x}}^{N_x}\Bigg(\frac{\partial}{\partial \mathbf{r}}\left(\parallel \nabla \mathbf{I}(\mathbf{x}) \parallel\right)\phi(\theta(\mathbf{I}(\mathbf{x})) - i) \\ &\quad + \parallel \nabla \mathbf{I}(\mathbf{x}) \parallel \frac{\partial}{\partial \mathbf{r}}\left(\phi(\theta(\mathbf{I}(\mathbf{x})) - i)\right)\Bigg)\end{aligned} \tag{26}$$

where

$$\begin{aligned}\frac{\partial}{\partial \mathbf{r}}\left(\parallel \nabla \mathbf{I}(\mathbf{x}) \parallel\right) &= \frac{\partial}{\partial \mathbf{r}}\left(\sqrt{(\nabla_x \mathbf{I}(\mathbf{x}))^2 + (\nabla_y \mathbf{I}(\mathbf{x}))^2}\right) \\ &= \frac{\left(2\nabla_x \mathbf{I}(\mathbf{x})\frac{\partial}{\partial \mathbf{r}}\left(\nabla_x \mathbf{I}(\mathbf{x})\right) + 2\nabla_y \mathbf{I}(\mathbf{x})\frac{\partial}{\partial \mathbf{r}}\left(\nabla_y \mathbf{I}(\mathbf{x})\right)\right)}{2\sqrt{(\nabla_x \mathbf{I}(\mathbf{x}))^2 + (\nabla_y \mathbf{I}(\mathbf{x}))^2}} \\ &= \frac{\nabla_x \mathbf{I}(\mathbf{x})\frac{\partial}{\partial \mathbf{r}}\left(\nabla_x \mathbf{I}(\mathbf{x})\right) + \nabla_y \mathbf{I}(\mathbf{x})\frac{\partial}{\partial \mathbf{r}}\left(\nabla_y \mathbf{I}(\mathbf{x})\right)}{\parallel \nabla \mathbf{I}(\mathbf{x}) \parallel}\end{aligned} \tag{27}$$

and with,

$$\begin{aligned}&\frac{\partial}{\partial \mathbf{r}}\left(\phi(\theta(\mathbf{I}(\mathbf{x})) - i)\right) = \frac{\partial}{\partial i}\phi\left(\theta(\mathbf{I}(\mathbf{x})) - i\right)\frac{\partial}{\partial \mathbf{r}}\left(\text{atan}\left(\frac{\nabla_y \mathbf{I}(\mathbf{x})}{\nabla_x \mathbf{I}(\mathbf{x})}\right)\right) \\ &= \frac{\partial}{\partial i}\phi\left(\theta(\mathbf{I}(\mathbf{x})) - i\right)\frac{1}{1+\left(\frac{\nabla_y \mathbf{I}(\mathbf{x})}{\nabla_x \mathbf{I}(\mathbf{x})}\right)^2}\frac{\partial}{\partial \mathbf{r}}\left(\frac{\nabla_y \mathbf{I}(\mathbf{x})}{\nabla_x \mathbf{I}(\mathbf{x})}\right)\end{aligned} \tag{28}$$

with

$$\frac{\partial}{\partial \mathbf{r}}\left(\frac{\nabla_y \mathbf{I}(\mathbf{x})}{\nabla_x \mathbf{I}(\mathbf{x})}\right) = \frac{\frac{\partial}{\partial \mathbf{r}}\left(\nabla_y \mathbf{I}(\mathbf{x})\right)\nabla_x \mathbf{I}(\mathbf{x}) - \nabla_y \mathbf{I}(\mathbf{x})\frac{\partial}{\partial \mathbf{r}}\left(\nabla_x \mathbf{I}(\mathbf{x})\right)}{\left(\nabla_x \mathbf{I}(\mathbf{x})\right)^2} \tag{29}$$

We now only needs to determine the values of $\frac{\partial}{\partial \mathbf{r}}\left(\nabla_x \mathbf{I}(\mathbf{x})\right)$ and $\frac{\partial}{\partial \mathbf{r}}\left(\nabla_y \mathbf{I}(\mathbf{x})\right)$.

As for the previous method, from [5], if and only if the *optical flow constraint equation* (OFCE) hypothesis is valid, we can get the following expression:

$$\mathbf{L}_{\boldsymbol{s}} = \nabla_x \mathbf{s}(\mathbf{r})\mathbf{L}_x + \nabla_y \mathbf{s}(\mathbf{r})\mathbf{L}_y \tag{30}$$

We can thereby define $\mathbf{L}_{\nabla_x \mathbf{I}(\mathbf{x})}$ such as:

$$\begin{aligned}\mathbf{L}_{\nabla_x \mathbf{I}(\mathbf{x})} &= \frac{\partial}{\partial x}\left(\nabla_x \mathbf{I}(\mathbf{x})\right)\mathbf{L}_x + \frac{\partial}{\partial y}\left(\nabla_x \mathbf{I}(\mathbf{x})\right)\mathbf{L}_y \\ &= \nabla_x^2 \mathbf{I}(\mathbf{x})\mathbf{L}_x + \nabla_{xy}\mathbf{I}(\mathbf{x})\mathbf{L}_y\end{aligned} \tag{31}$$

And in the same way:

$$\begin{aligned}\mathbf{L}_{\nabla_y \mathbf{I}(\mathbf{x})} &= \frac{\partial}{\partial x}\left(\nabla_y \mathbf{I}(\mathbf{x})\right)\mathbf{L}_x + \frac{\partial}{\partial y}\left(\nabla_y \mathbf{I}(\mathbf{x})\right)\mathbf{L}_y \\ &= \nabla_{yx}\mathbf{I}(\mathbf{x})\mathbf{L}_x + \nabla_y^2 \mathbf{I}(\mathbf{x})\mathbf{L}_y\end{aligned} \tag{32}$$

where $\mathbf{L}_x$ and $\mathbf{L}_y$ are defined in equation (22)

V. Experimental results and comparisons

A. 6 DoF positioning task on a gantry robot

In order to validate these approaches, we perform a positioning task using the 3 control laws designed previously: HOG, color and synthesized gray level from color planes, the initial intensity histogram being already validated by experiments in [1]. The test scene can be seen in Figs 2, 3 and 4. In order to test the robustness of the methods, this scene contains a variety of textures, from homogeneous textured-wood to specular surfaces and scattered 3D objects. Initial positioning error is such as (14.4*cm*, -17.7*cm*, 1.3*cm*, -28°, -18°, -2.4°), with a mean depth of 80*cm*. The only variable parameter in these experiments is the number of bins used in the computation of the histograms: 32 for the intensity histogram and synthesized one plane color histogram, 10 for the HOG and 8 for the Hue-Saturation histogram. The μ parameter of the LM minimization scheme is set constant at $10e^{-3}$ during all the experiments. Concerning computational complexity, the gray-level, HOG and synthesized color histogram perform

equivalently, the HS method being twice as slow due to the increased complexity of the corresponding interaction matrix. On a i7-4600 processor, we obtain an iteration time of 70ms for the 3 first methods and 140ms for the HS method. We can see that all 4 methods succeed in converging towards the expected position, despite the nature of the scene and the strong hypothesis used in the design of the control laws (presence of non-lambertian surfaces and for varying depths between the camera and the scene).

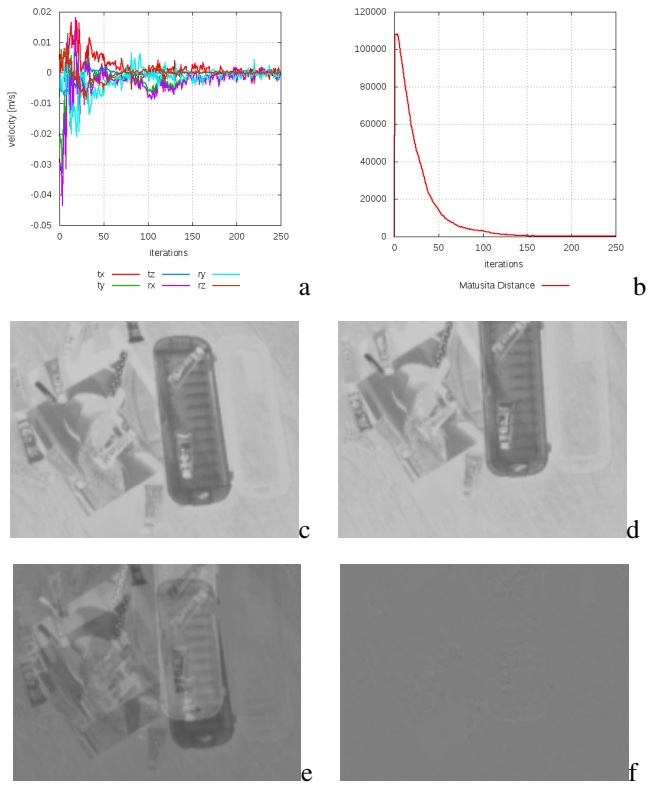

Figure 2. Synthesized gray level from Hue-Saturation histogram-based servoing. Camera velocities in m/s and rad/s in (a). (b) Matusita distance. (c) Initial image. (d) Desired image. (e) I - I* at initial position. (f) I - I* at the end of the motion

B. Comparing convergence area in simulation

1) Increasing initial position distance: Since all 4 methods are validated by real robot experiments, further testing is required in order to compare them in a more detailed way. In terms of convergence area: random initial positions are computed, with an increasing spatial noise from the desired position. For each step of increasing spatial noise, multiple trials are run. By determining for each run if the method successfully converges (spatial error below a given threshold), it gives us a percentage of successful convergence. By repeating this operation with all the 4 control laws, it is then possible to compare them in terms of convergence. For this test, 10 increases of the spatial noises are executed. For each steps, 40 runs are performed to get the percentage of convergence. The mean depth is of 10cm and the spatial gaussian noise is applied such as :

- from 0 to 1cm in the mean standard deviation for the x/y translations;

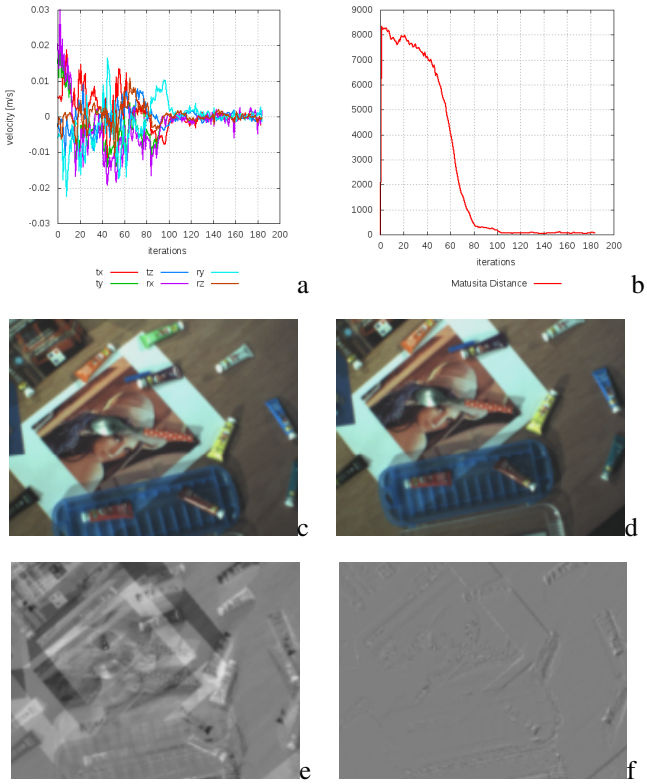

Figure 3. Hue-Saturation histogram-based servoing. Camera velocities in m/s and rad/s in (a). (b) Matusita distance. (c) Initial image. (d) Desired image. (e) I - I* at initial position. (f) I - I* at the end of the motion

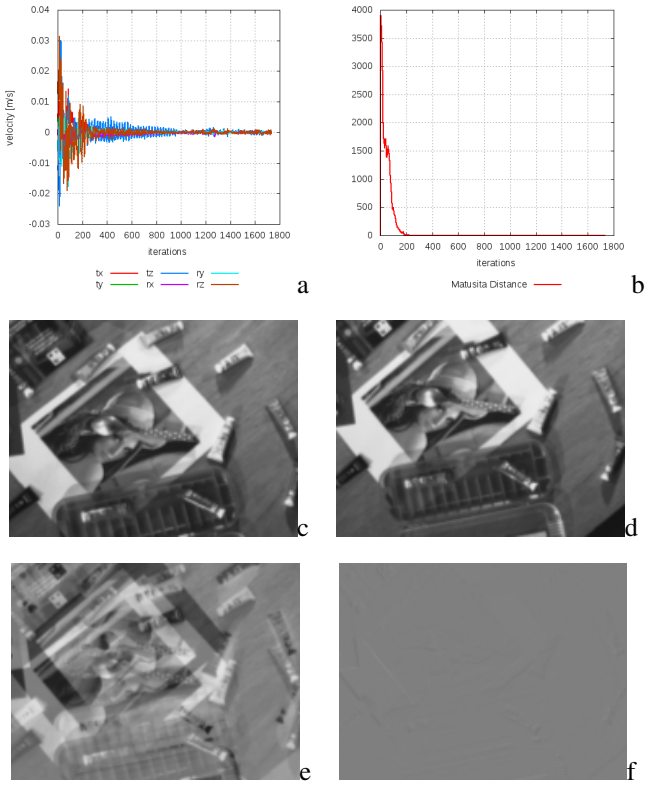

Figure 4. HOG-based servoing. Camera velocities in m/s and rad/s in (a). (b) Matusita distance. (c) Initial image. (d) Desired image. (e) I - I* at initial position. (f) I - I* at the end of the motion

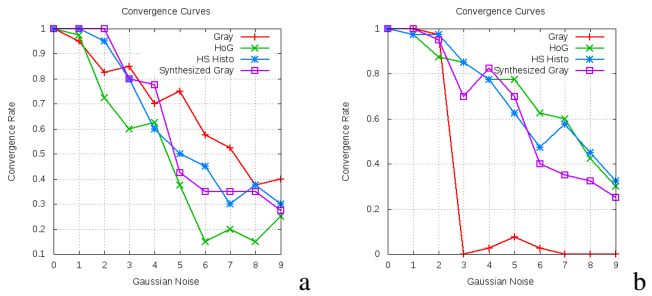

Figure 5. Convergence areas. a: with increasing spatial noise on initial position; b:with increasing spatial noise on initial position and diminishing illumination

- from 0 to 5*cm* for the depth;
- from 0 to 10°for the rotations around the x/y axis;
- from 0 to 20°for the rotations around the z axis.

The result of this test can be seen in Fig. 5(a). We can see that all four methods perform rather similarly. The three new methods do not improve the convergence radius of the intensity histogram-based method, which was to be expected since the new histograms where not chosen to address this issue but the illumination change issue.

2) Decreasing global luminosity: Since the main problem tackled with the introduction of these new histogram-based methods is to reduce the global illumination sensitivity of the HBVS, the same test is performed with an addition: at each step, in top of the increase in spatial noise, the global illumination of the scene is reduced linearly. It allows us to compare the methods in terms of sensibility to global illumination changes. The same noise as in the previous experiment is applied at each step, but additionally, the illumination of the image plane goes from 100% to 10%. The result of this test can be seen in Fig. 5(b).

The main information provided by this graphic is that all three new methods keep performing at the same level despite the increasing illumination change. This emphasizes the fact that the new methods do possess an invariance to illumination changes as hinted by the nature of the chosen histograms. It is interesting to note that this invariance do not apply in the same way for all three methods, since the color-based ones requires a change in illumination that do not alter the coloration of the scene (a change in a white light illumination, such as the sun during the main part of the day), whereas the HOG-based method does not suffer from this problem, being based purely on the orientation of the gradients: neither the change in intensity, nor color alters this property.

One interesting comment concerning the analysis of these figures is that the HOG method performs better under mild illumination decrease that under full illumination. This is due to the shape of its cost function that become smoother, with less local minima, when the illumination in the image is low. This is caused by an induced quantification in the pixels intensities that homogenize large texture-less areas with randomly oriented gradients that provide non-informative data in the histogram.

C. Navigation by visual path

Navigation by visual path consists in the autonomous navigation of a mobile robot based on the recording of a predetermined path. From this recording, key-frames are extracted and the robot navigates in order to minimize the difference between its current camera view and the first key-frame. The method then proceeds iteratively, going through all the key-frames, therefore following the same spatial path as the one corresponding to where the key-frames where acquired (as illustrated in the Fig. 6). In the first experiment, the robot navigates at first around 20 meters inside corridors lit by both artificial and natural lighting through large windows. Then in the second experiment, it navigates around 20 meters in an outdoor environment. Some pedestrian activity occurs during the experiment, creating important occlusions. The robot is a non-holonomic Pioneer robot with 2 DoF. Here, only the rotation is controlled by visual servoing, the forward translation being fixed. Since the goal of this experience is to test the robustness of the histogram-based method, no elaborate scheme of key-frames selection is performed: the key-frames are acquired at a fixed rate (1.5Hz). In order to benefit from the maximal invariance with respect to light variation, we choose here to navigate using the HOG-based visual servoing. It is interesting to comment the fact that we do not propose a quantitative measure concerning the match between the 3D path recorded and the path undertaken by the robot because our goal is to follow a path in the image-space. As such, our goal is only to go from a starting point to a destination, independently of the eventual minor rerouting that can occur due to partial occlusions or light changes for example.

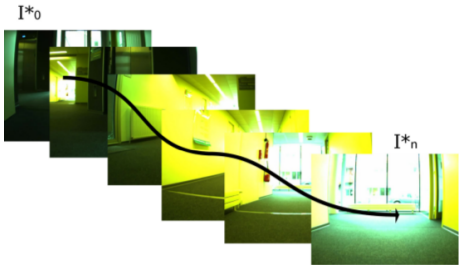

Figure 6. Illustration of the visual path approach

As we can see in the Fig. 7, the proposed method succeeds in navigating indoor along the visual path, even in presence of important discrepancies due to turns in the path that can be seen for example in 7(n) or the presence of a pedestrian that was recorded in the reference visual path, as seen in 7(e). The second experiment depicted in Fig. 8 has been performed outdoor in order to validate the invariance to mild illumination changes due to meteorological conditions (variably cloudy conditions) and shows similar good performances.

VI. Conclusion and perspectives

In this paper extending [1], we presented a generic framework to compute Histogram-Based Visual Servoing control laws that can apply to any kind of histograms. This framework

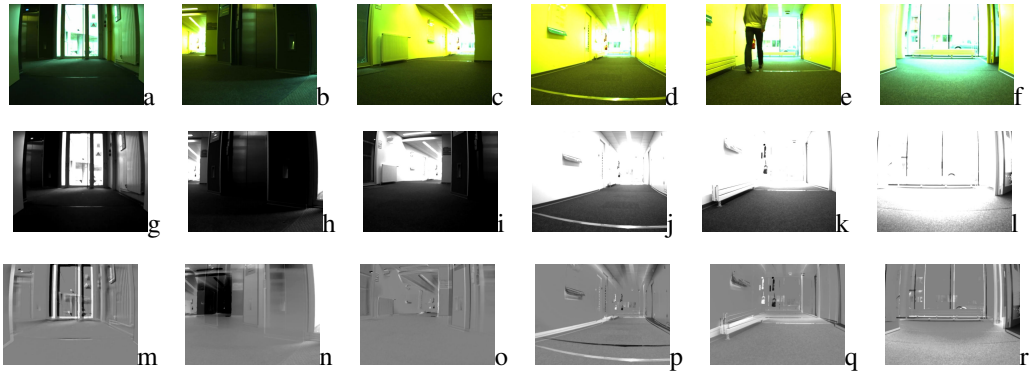

Figure 7. Samples of the navigation experiment in the indoor scene: first line is the current key-frame, second is the actual view of the robot and third line is the difference between the two previous lines.

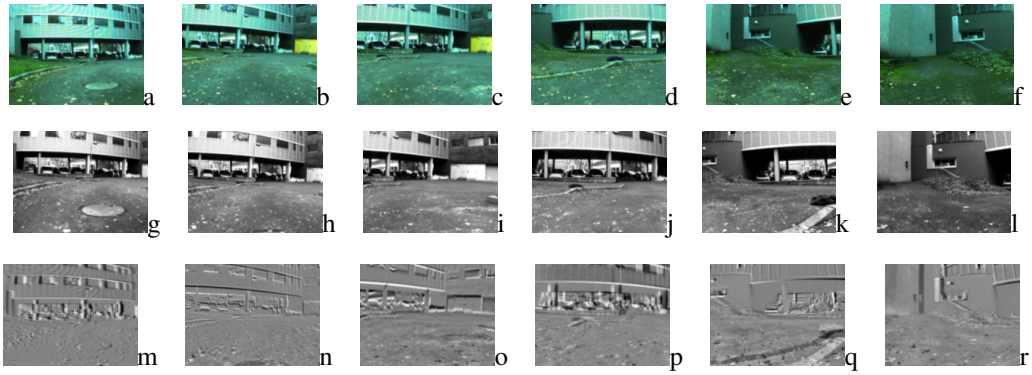

Figure 8. Samples of the navigation experiment, in the outdoor scene: first line is the current key-frame, second is the actual view of the robot and third line is the difference between the two previous lines.

was applied to three kinds of histograms, from intensity histograms (photometric or synthesized), to Hue-Saturation color histograms and to Histograms of Oriented Gradients. These methods were compared and proved their feasibility on both simulated and real experiments to validate invariance properties of the chosen histograms. Future works will concern investigating more closely the link between the sub-images positioning, like introducing an overlap or using pyramidal distribution, and the global performances. Another perspective is to devise ways to extend the convergence radius for this class of method, for example by integrating a notion of confidence map to decrease the impact of some uninformative image regions.

References

[1] Q. Bateux and E. Marchand. Direct visual servoing based on multiple intensity histograms. In *IEEE ICRA'15*, Seattle, May 2015.

[2] S.-H. Cha and S. Srihari. On measuring the distance between histograms. *Pattern Recognition*, 35(6):1355–1370, 2002.

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

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

[5] C. Collewet, E. Marchand, and F. Chaumette. Visual servoing set free from image processing. In *IEEE ICRA'08*, May 2008.

[6] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. on PAMI*, 25(5):564–577, May 2003.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE CVPR'05*, pages 886–893, 2005.

[8] A. Dame and E. Marchand. Mutual information-based visual servoing. *IEEE Trans. on Robotics*, 27(5):958–969, October 2011.

[9] B. Delabarre and E. Marchand. Visual servoing using the sum of conditional variance. In *IEEE/RSJ IROS'12*, October 2012.

[10] O. Déniz, G. Bueno, J. Salido, and F. De la Torre. Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*, 32(12):1598–1603, 2011.

[11] G. Hager, M. Dewan, and C. Stewart. Multiple kernel tracking with ssd. In *IEEE CVPR'04*, pages 790–797, June 2004.

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

[13] S. Jeong, C.S. Won, and R.M. Gray. Image retrieval using color histograms generated by gauss mixture vector quantization. *CVIU*, 94(1):44–66, 2004.

[14] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[15] E. Marchand and F. Chaumette. Feature tracking for visual servoing purposes. *Robotics and Autonomous Systems*, 52(1):53–70, June 2005.

[16] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006.

[17] P. Pérez, C Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *ECCV 2002*, pages 661–675. Springer, 2002.

[18] R. Richa, R. Sznitman, R. Taylor, and G. Hager. Visual tracking using the sum of conditional variance. In *IEEE IROS'11*, September 2011.

[19] M.J. Swain and D.H. Ballard. Indexing via color histograms. In *Active Perception and Robot Vision*, pages 261–273. Springer, 1992.

[20] H. Zhang, Y. Wang, and X. Jiang. An improved shot segmentation algorithm based on color histograms for decompressed videos. In *Int. Conf on Image and Signal Processing*, pages 86–90, 2013.