Preserving the continuity of visual servoing despite changing image features

Nicolas M. Garcia
Dept. of Industrial Systems Engineering
Miguel Hernandez University
Elche (Alicante), SPAIN

Ezio Malis
I.N.R.I.A.
Sophia-Antipolis, FRANCE

***Abstract*— This paper deals with the problem occurring when features go in or out of the image during the visual servoing task. The appearance/disappearance of image features during the control task produces discontinuities in the control law that affect the performance of the system. In this paper, we propose a solution in order to avoid these discontinuities by the use of weighted image features. In particular, we redefine the camera invariant visual servoing approach in order to take into account the change of image features when zooming in or out during a positioning task. Simulations and experimental results demonstrate the improvements that can be obtained in the performance of the vision-based control task.**

I. INTRODUCTION

Visual servoing techniques have been widely investigated in the last three decades [2], [5], [6]. The visibility problem received particular attention in the recent past: *a minimum number of image features must remain in the field of view of the camera during the servoing*. Research in this field has concentrated on visual servoing methods that are able to keep always the object in the field of view. Path planning in the image space [10] is an elegant solution to the problem. When several constraints (visibility, robot mechanical limits, etc) are simultaneously considered by a path planning scheme, the camera trajectory is deviated from the optimal one [11]. By using a zooming camera we can more easily follow the planned optimal trajectory [9]. Indeed, zooming while controlling a camera [3] has the main advantage of improving the visibility of features during the servoing. Nevertheless, zooming to keep all features in the field of view it is not always possible in all cases. For this reason, we propose in this paper to study the way of allowing the temporary disappearance of image features during the control task. From this point of view, we can control the camera without imposing too strict constraints which are considered by the path planing solution and also we can use the zoom to obtain more details of the scene without the constraint of controlling the zoom to keep all features in the field of view. The key idea of the proposed approach is to allow some features to appear or disappear from the image rather than trying to keep all of them in the image. Note that, the appearance/disappearance of features during a positioning task produces discontinuities in the control task that affect to the performance of the system. So we need to ensure that the new approach avoids all the discontinuities in the control law produced by the free movement of features in the image plane. To do this, the concept of weighted features is introduced and a smooth task function based on them is defined [12]. With this task function, and defining appropriate weight functions, it is possible to control the camera directly with the standard image-based visual servoing technique but not with the invariant one. To be able to use all the interesting characteristics of the invariant visual servoing approach, it must be reformulated to take into account the weighted features. Simulation and experimental results using this new approach to the visibility problem are presented to demonstrate the efficiency of the proposed method. The experiments have been carried out using a 6 degrees of freedom eye-in-hand system composed by an industrial robot and a micro-head camera. The paper is organized as follows. In Section II, the problem with the appearance/disappearance of image features is presented. A solution is proposed in Section III. It is structured in the presentation of the concept of weighted features, the definition of a smooth task function an finally the formulation of invariant visual servoing with them. In the last section, some experimental results with an industrial robot is shown.

II. THE PROBLEM OF CHANGING IMAGE FEATURES

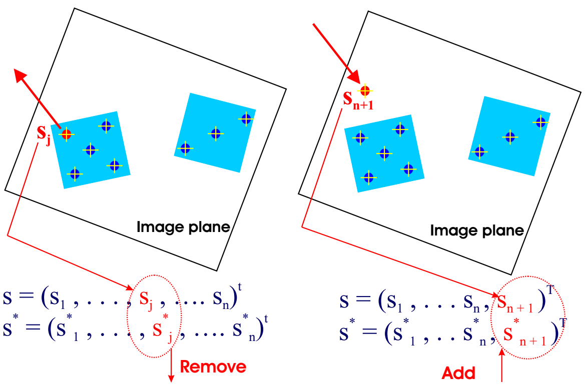
In this section, we describe more in details the discontinuity problem that occurs when some features go in/out of the image during the vision-based control. A simple simulation illustrate the effects of the discontinuities on the control law and on the performances of the visual servoing.

A. What happens when features appear or disappear?

Consider a standard vision-based positioning task. The goal is to bring the robot end-effector back to a reference position ($\boldsymbol{\xi}^*$) with an eye-in-hand camera. This means that a feature vector $\mathbf{s}(\boldsymbol{\xi})$, which contains the information of the current image, has to converge to a reference feature vector $\mathbf{s}^*(\boldsymbol{\xi}^*)$. We use the task function approach [12] which consists in minimizing an error vector $\mathbf{e}$:

$$\mathbf{e} = \widehat{\mathbf{L}}^{+}(\mathbf{s} - \mathbf{s}^*) \tag{1}$$

where $\widehat{\mathbf{L}}^{+}$ is the pseudo-inverse of the estimated interaction matrix, $\mathbf{s}(\boldsymbol{\xi}) = (\mathbf{s_1}, \mathbf{s_2}, \cdots, \mathbf{s_n})$, $\mathbf{s}^*(\boldsymbol{\xi}^*) = (\mathbf{s_1^*}, \mathbf{s_2^*}, \cdots, \mathbf{s_n^*})$ are the current and reference features and $\boldsymbol{\xi}$, $\boldsymbol{\xi}^*$ is a (6×1) vector containing respectively the current and reference position of the camera in the Cartesian space. A local exponential decrease of the task function can be imposed by choosing a proportional control law $\mathbf{v} = -\lambda\ \mathbf{e}$ where $\mathbf{v}$ is the velocity of the camera and λ is a positive scalar factor which tunes the speed of convergence. When one or more features appear or disappear during the servoing, the features will be added to or removed from the error vector (Figure 1). This change produce a jump discontinuity in the control law. The magnitude of the discontinuity in control law depends on the number of the features that go in or go out of the image plane at the same time, the distance between the current and reference features, and the pseudo-inverse of interaction matrix.



(a) Disappearance of image feature

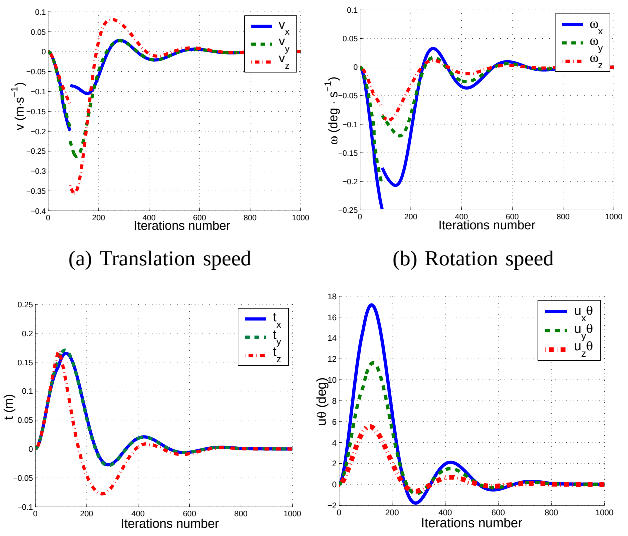
(b) Appearance of a new image feature

Fig. 1: What happens when an image feature appear or disappear in the image plane.

In the case of invariant visual servoing approach, the effect produced by the appearance/disappearance of features is more important since the invariant space $\mathcal{Q}$ used to compute the current and the reference invariant points $(\mathbf{q}, \mathbf{q}^*)$ changes with them. To show the discontinuities produced by the appearance/disappearance of features during the control, different experiments with simulated data have been carried out. One of them is the control of a camera that is moving from its reference pose due to a perturbation and at the same time is zooming in. As the camera is changing its focal length, the invariant visual servoing approach has to be used to keep it in its reference position. In Figure 2(a)(b), the discontinuities produced by the disappearance and appearance of features can be seen. If their magnitudes are sufficiently large, they will cause an unwanted variation in the position and orientation of the camera as it is shown in Figure 2. In Figure 2(c), a sharp change in the position of the camera can be appreciated.

III. A SOLUTION TO PRESERVE THE CONTINUITY OF THE CONTROL LAW

In the previous section, the continuity problem of the control law due to the appearance/disappearance of features has been shown. In this section a solution to preserve the continuity is presented. The section is organized as follows. First, the concept of weighted features is defined. Then, the definition of a smooth task function is presented



(a) Translation speed

(b) Rotation speed

(c) Translation error

(d) Rotation error

Fig. 2: Simulation of the invariant visual servoing approach: discontinuities in the control law when features disappear.

and finally the reason to reformulate the invariant visual servoing approach and its development is explained.

A. *Weighted features*

A possible solution to preserve the continuity of the control law is to weight the features depending on their position in the image. The weights are used in order to anticipate in some way the possible discontinuities produced in the control law by the appearance/disappearance of image features. The key idea in this formulation is that every feature (points, lines, moments, etc) has its own weight which may be a function of image coordinates (u, v) and/or a function of the distance between feature points and an object which would be able to occlude them, etc. In this paper, the weights are computed by a function that depends on the position of image feature (u, v). Representing the weight as γ_y, the function that has been used and tested to compute the magnitude of the weights is:

$$\gamma_y(x) = \begin{cases} e^{-\frac{(x-x_{med})^{2n}}{(x-x_{min})^m (x_{max}-x)^m}} & x_{min} < x < x_{max} \\ 0 & x = \{x_{min}, x_{max}\} \end{cases}$$

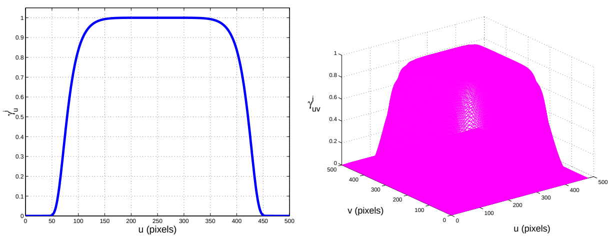
The function weight $\gamma_y(x)$ is a bell-shaped function which is symmetrical respect to $x_{med} = \frac{x_{min}+x_{max}}{2}$.With n, m parameters, the shape of the bell function can be controlled. Their values must be chosen according to the following conditions:

$$\begin{cases} \gamma_y(x_{min} + \beta(x_{max} - x_{min})) & \geq 1 - \alpha \\ \gamma_y(x_{min} + \dfrac{\beta}{2}(x_{max} - x_{min})) & \leq \alpha \end{cases} \qquad (2)$$

where $0 < \alpha < 0.5$ and $0 < \beta < 0.5$. If the conditions (2) are verified then the following conditions are verified too:

$$\begin{cases} \gamma_y(x_{max} - \beta(x_{max} - x_{min})) & \geq 1 - \alpha \\ \gamma_y(x_{max} - \dfrac{\beta}{2}(x_{max} - x_{min})) & \leq \alpha \end{cases} \qquad (3)$$

where $0 < \alpha < 0.5$ and $0 < \beta < 0.5$. In these simulations, a camera with an image size of 500x500 pixels is chosen. For each feature with (u_i, v_i) coordinates, a weight $\gamma_y(x)$ for its u_i and v_i coordinates can be calculated using the definition of the function $\gamma_y(x)$ and has been denoted as ($\gamma_u^i = \gamma_y(u_i)$ and $\gamma_v^i = \gamma_y(v_i)$ respectively). For this camera, $x_{min} = 0$, $x_{max} = 500$ and of course $x_{med} = 250$ and then n and m values can be chosen, for example if α and β is fixed to $\alpha = 0.1$ and $\beta = 0.2$ (it means that the function loses 10% of its maximum value when the distance of the image point from the border is the 20% of the image size (see Figure 3(a)), then n and m values will be 3. Finally, for every image feature, a total weight that will be denoted (γ_{uv}^i) is computed by multiplying the weight of its u coordinate (γ_u^i) by the weight of its v coordinate (γ_v^i). In Figure 3(b), the total weight γ_{uv}^i ,which has been computed for all the image points, is represented. Looking at it carefully, we can see that the magnitude of γ_{uv}^i tends to zero near the border and to one near the center of the image.



(a) Weight $\gamma_u^i = \gamma_y(u_i)$ (b) Total weight $\gamma_{uv}^i = \gamma_u^i \cdot \gamma_v^i$

Fig. 3: Features Weight

B. Smooth Task function

Suppose that n matched points are available in both images, current and reference. Every point has a weight γ_{uv}^i which is used to build the following task function [12]:

$$\mathbf{e} = \mathbf{CW}\ (\mathbf{s} - \mathbf{s}^*(t)) \tag{4}$$

where $\mathbf{W}$ is a $(2n \times 2n)$ diagonal matrix where its elements are the weights γ_{uv}^i. The derivative of the task function,considering $\mathbf{C}$ constant, will be:

$$\dot{\mathbf{e}} = \mathbf{CW}\ (\dot{\mathbf{s}} - \dot{\mathbf{s}}^*) + \mathbf{C\dot{W}}\ (\mathbf{s} - \mathbf{s}^*) \tag{5}$$

Substituting $\dot{\mathbf{s}} = \mathbf{L}\ \mathbf{v}$ in (5), we obtain:

$$\dot{\mathbf{e}} = \mathbf{CW}\ \mathbf{Lv} - \mathbf{CW}\dot{\mathbf{s}}^* + \mathbf{C\dot{W}}\ (\mathbf{s} - \mathbf{s}^*) \tag{6}$$

A simple control law can be obtained by imposing the exponential convergence of the task function to zero:

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \tag{7}$$

where λ is a positive scalar factor which tunes the speed of convergence:

$$\mathbf{CWLv} - \mathbf{CW}\dot{\mathbf{s}}^* + \mathbf{C\dot{W}}(\mathbf{s} - \mathbf{s}^*) = -\lambda \mathbf{e}$$

$$\begin{aligned} \mathbf{v} &= -\lambda\ (\mathbf{CWL})^{-1}\mathbf{e} + (\mathbf{CWL})^{-1}\mathbf{CW}\dot{\mathbf{s}}^* + \\ &+ \ (\mathbf{CWL})^{-1}\ \mathbf{C\dot{W}}\ (\mathbf{s} - \mathbf{s}^*) \end{aligned} \tag{8}$$

Let us suppose that these weights γ_{uv}^i are varying slowly, then $\dot{\mathbf{W}}$ can be considered nearly equal to zero ($\dot{\mathbf{W}} \approx 0$). Considering this assumption, the equation (8) can be rewritten as:

$$\mathbf{v} = -\lambda\ (\mathbf{CWL})^{-1}\mathbf{e} + (\mathbf{CWL})^{-1}\mathbf{CW}\dot{\mathbf{s}}^* \tag{9}$$

Setting $\mathbf{C} = (\mathbf{W}^*\mathbf{L}^*)^+$ (i.e. the matrices computed with the references values), if $(\mathbf{CWL}) > 0$ then the task function converges to zero and, in the absence of local minima and singularities, so does the error $\mathbf{s} - \mathbf{s}^*$.

IV. Application to Invariant Visual Servoing

The theoretical background about invariant visual servoing can be extensively found in [7][8][9]. In this section, we modify the approach in order to take into account weighted features. Suppose that n image points are available. The weights γ_i used in the weighted invariant visual servoing are obtained as follows:

$$\boldsymbol{\gamma_i} = \sqrt{\frac{n}{\sum_{i=1}^{n} (\boldsymbol{\gamma_{uv}^i})^2}} \cdot \boldsymbol{\gamma_{uv}^i} \tag{10}$$

The weights γ_{uv}^i defined in the previous section are *redistributed* in order to have $\sum \boldsymbol{\gamma_i}^2 = n$. Every image point with projective coordinates $\mathbf{p_i} = (u_i, v_i, 1)$ is multiplied by its own weight γ_i in order to obtain a weighted point $\mathbf{p}_i^{\gamma_i} = \boldsymbol{\gamma_i}\ \mathbf{p}_i$. Using all the weighted points we can compute the following symmetric (3×3) matrix:

$$\mathbf{S}_p^{\gamma_i} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{p}_i^{\gamma_i}\ \mathbf{p}_i^{\gamma_i\top} \tag{11}$$

The image points depends on the upper triangular matrix $\mathbf{K}$ containing the camera intrinsic parameter and on the normalized image coordinates $\mathbf{m}_i$: $\mathbf{p}_i = \mathbf{K}\mathbf{m}_i$. Thus, we have $\mathbf{p}_i^{\gamma_i} = \mathbf{K}(\gamma_i \mathbf{m}_i) = \mathbf{K}\mathbf{m}_i^{\gamma_i}$ and the matrix $\mathbf{S}_p^{\gamma_i}$ can be written as follows:

$$\mathbf{S}_p^{\gamma_i} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{p}_i^{\gamma_i} \mathbf{p}_i^{\gamma_i\top} = \mathbf{K}\ \mathbf{S}_m^{\gamma_i}\ \mathbf{K}^\top \tag{12}$$

where $\mathbf{S}_m^{\gamma_i}$ is a symmetric matrix which is does not directly depend on the camera parameters:

$$\mathbf{S}_m^{\gamma_i} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{m}_i^{\gamma_i} \mathbf{m}_i^{\gamma_i\top} \tag{13}$$

If the points are not collinear and $n > 3$ then $\mathbf{S}_\mathbf{p}^{\gamma_i}$ and $\mathbf{S}_\mathbf{m}^{\gamma_i}$ are positive definite matrices and they can be written, using a Cholesky decomposition, as:

$$\mathbf{S}_p^{\gamma_i} = \mathbf{T}_p^{\gamma_i}\ \mathbf{T}_p^{\gamma_i\top} \quad \text{and} \quad \mathbf{S}_m^{\gamma_i} = \mathbf{T}_m^{\gamma_i}\ \mathbf{T}_m^{\gamma_i\top} \tag{14}$$

From equations (12) and (14), the two transformation matrices, can be related by:

$$\boldsymbol{T}_p^{\gamma_i} = \boldsymbol{K}\ \boldsymbol{T}_m^{\gamma_i} \tag{15}$$

The matrix $\mathbf{T}_p^{\gamma_i}$ defines a projective transformation and can be used to define a point in a new projective space $\mathcal{Q}^{\gamma_i}$:

$$\mathbf{q}_i = \mathbf{T}_p^{\gamma_i\,-1}\mathbf{p}_i = \mathbf{T}_m^{\gamma_i\,-1}\mathbf{K}^{-1}\mathbf{p}_i = \mathbf{T}_m^{\gamma_i\,-1}\mathbf{m}_i \qquad (16)$$

The new projective space $\mathcal{Q}^{\gamma_i}$ does not depend directly on camera intrinsic parameters but it only depends on the weights γ_i and on the normalized points. The normalized points $\mathbf{m}_i$ depends on a (6×1) vector $\boldsymbol{\xi}$ containing global coordinates of an open subset $\mathcal{S} \subset \mathbb{R}^3 \times SO(3)$ (i.e. represents the position of the camera in the Cartesian space). Suppose that a reference image of the scene, corresponding to the reference position $\boldsymbol{\xi}^*$ has been stored and computed the reference points $\mathbf{p}_i^*$ in a previous learning step. The camera parameters $\mathbf{K}^*$ are eventually different from the current camera parameters. We use the same weights γ_i to compute the weighted reference $\mathbf{p}_i^{*\gamma_i} = \gamma_i \mathbf{p}_i^*$. Similarly to the current image, we can define a reference projective space:

$$\mathbf{q}_i^* = \mathbf{T}_{p^*}^{\gamma_i\,-1}\mathbf{p}_i^* == \mathbf{T}_m^{*\gamma_i\,-1}\mathbf{K}^{*-1}\mathbf{p}_i^* = \mathbf{T}_{m^*}^{\gamma_i\,-1}\mathbf{m}_i^* \quad (17)$$

Note that, since the weights in equations (16) and (17) are the same, if $\boldsymbol{\xi} = \boldsymbol{\xi}^*$ then $\mathbf{q}_i = \mathbf{q}_i^* \;\forall\, i \in \{1, \ldots, n\}$ and the convergence is true even if the intrinsic parameters change during the servoing. Thus, similarly to the standard invariant visual servoing, the control of the camera is achieved by stacking all the reference points of space $\mathcal{Q}^{\gamma_i}$ in a $(3n\times1)$ vector $\mathbf{s}^*(\boldsymbol{\xi}^*) = (\mathbf{q}_1^*(t), \mathbf{q}_2^*(t), \cdots, \mathbf{q}_n^*(t))$. Similarly, the points measured in the current camera frame are stacked in the $(3n\times1)$ vector $\boldsymbol{s}(\boldsymbol{\xi}) = (\boldsymbol{q}_1(t), \boldsymbol{q}_2(t), \cdots, \boldsymbol{q}_n(t))$. If $\boldsymbol{s}(\boldsymbol{\xi}) = \boldsymbol{s}^*(\boldsymbol{\xi}^*)$ then $\boldsymbol{\xi} = \boldsymbol{\xi}^*$ and the camera is back to the reference position whatever the camera intrinsic parameters. The derivative of vector $\boldsymbol{s}$ is:

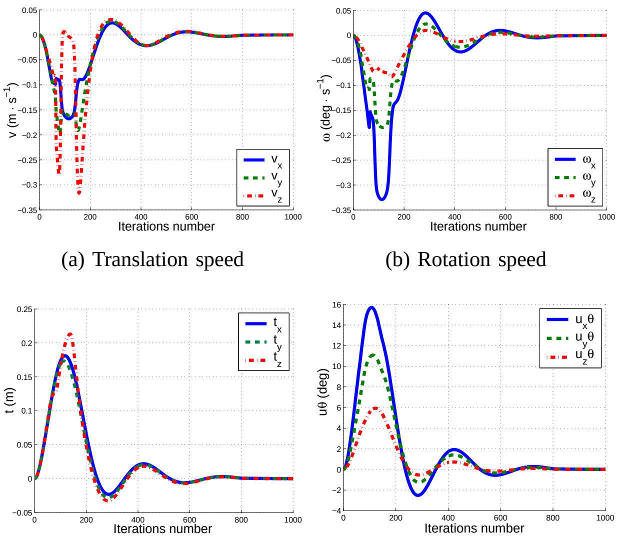
$$\dot{\boldsymbol{s}} = \boldsymbol{L}\,\boldsymbol{v} \qquad (18)$$

where the $(3n\times6)$ matrix $\boldsymbol{L}$ is called the interaction matrix and $\boldsymbol{v}$ is the velocity of the camera. The interaction matrix depends on current normalized points $\boldsymbol{m}_i(\boldsymbol{\xi}) \in \mathcal{M}$ ($\boldsymbol{m}_i$ can be computed from image points $\boldsymbol{m}_i = \boldsymbol{K}^{-1}\,\boldsymbol{p}_i$), on the invariant points $\boldsymbol{q}_i(\boldsymbol{\xi}) \in \mathcal{Q}^{\gamma}$, on the current depth distribution $\boldsymbol{z}(\boldsymbol{\xi}) = (Z_1, Z_2, ..., Z_n)$ and on the current redistributed weights γ_i. The interaction matrix in the weighted invariant space $(\boldsymbol{L}_{qi}^{\gamma_i} = \boldsymbol{T}_{mi}^{\gamma_i}\,(\boldsymbol{L}_{mi} - \boldsymbol{C}_i^{\gamma_i}))$is obtained like in [8] but the term $\boldsymbol{C}_i^{\gamma_i}$ must be recomputed in order to take into account the *r*edistributed weights γ_i.

In order to control the movement of the camera, we use the control law (9) where $\mathbf{W}$ depends on the weights previously defined and $\mathbf{L}$ is the interaction matrix recomputed for the invariant visual servoing with weighted features.

V. Simulation Results

In section II-A, the results of an experiment with simulated data is presented. The goal of this simulation is the control of a camera that is moving due to a perturbation and zooming in. In this case, the new formulation of invariant visual servoing with weighted features is used to control the same camera. Comparing Figures 2 and 4(a)(b), we can see the improvements of invariant visual servoing with weighted features in the continuity of the control law. The control law with this formulation is continuous (Figure 4(a)(b)) so the camera pose is also continuous without sharp movements (Figure 4(c)(d)).



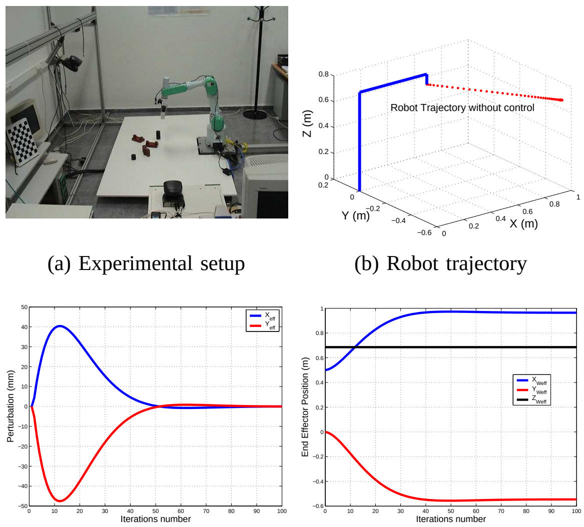
(a) Translation speed (b) Rotation speed

(c) Translation error (d) Rotation error

Fig. 4: Experiment with simulated data: Invariant visual servoing with weighted features.

VI. Experimental Results

Experimental results has been obtained using a 7 axis redundant Mitsubishi PA-10 manipulator (only 6 of its 7 dof have been considered). The experimental setup used in this work also include one camera (JAI CM 536) rigidly mounted in robot end-effector, some experimental objects and two computers one of them with a Matrox Genesis vision board and the other with the PA-10 controller board. The goal of these experiments is keeping the robot in a reference position in spite of the fact that a perturbation is applied to the end-effector position of the robot (Figure 5). The amplitude and duration of this perturbation was chosen to produce that some image features go out of the image plane during the control.



(a) Experimental setup (b) Robot trajectory

(c) Perturbation (d) End-Effector Position

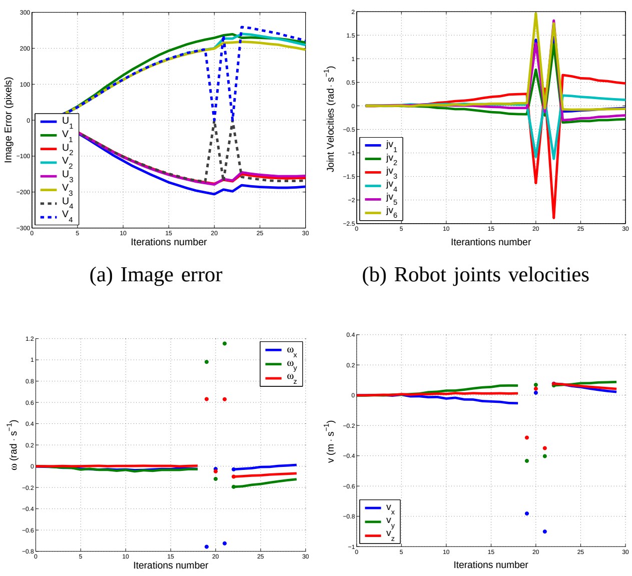
Fig. 5: Perturbation apply to the PA-10

A. *Invariant visual servoing approach*

We compare the weighted and un-weighted invariant visual servoing approaches using both, a constant and a varying interaction matrix.

1) *Constant Interaction Matrix:*

In this section, the interaction matrix is assumed constant and determined during off-line step using the desired value of the visual features and an approximation of the points depth at the reference camera pose. Then a perturbation is applied to the end-effector position of the robot (Figure 5). The goal of the control is to keep the robot in the reference position using the invariant visual servoing approach. During the experiment, one feature go out of the image plane (feature with (U_4V_4) in Figure 6a). Due to the disappearance of this feature, a discontinuity is produced in the control law (Figure 6c-d). In Figure 6b, the robot end-effector velocities can be seen. As it is shown in it, the system becomes unstable due to the lost of a feature during the control. In the second experiment, the new formulation of invariant visual servoing with weighted features was used. In Figure 7f, the weights of every feature can be seen. Some features go out or are near the border of the image plane (features 2, 3, 4). As it is shown in Figure 7(b), the system is stable even though one feature goes out of the image plane. The control law in this case is continuous (Figure 7c-d). Note that you can not appreciate the same noise in the Figure 7 than in the Figure 6 due to the difference between limits of the figures.



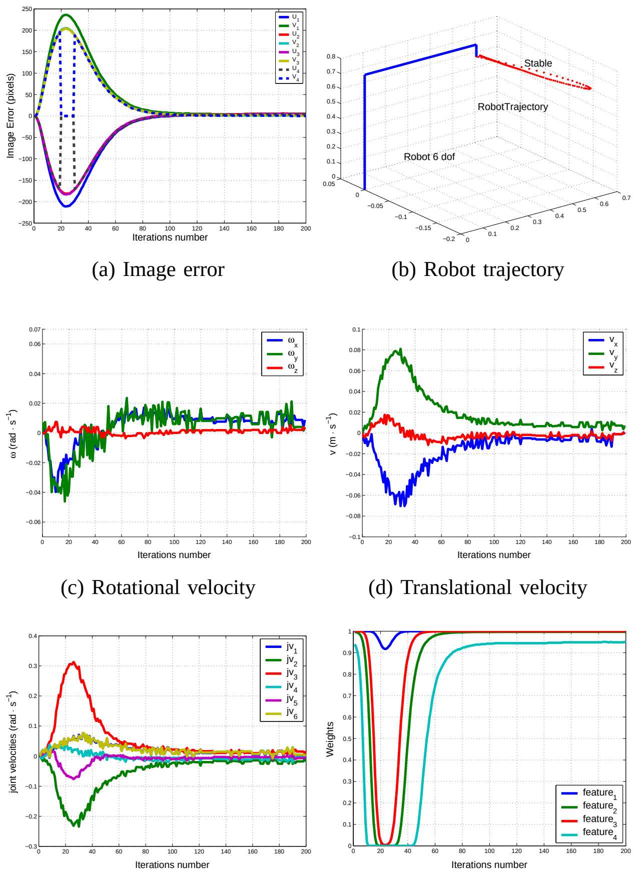
(a) Image error (b) Robot joints velocities

(c) Rotational velocity (d) Translational velocity

Fig. 6: Experimental results: Robot with an eye-in-hand camera perturbated with respect to its reference pose. The interaction matrix is constant.

2) *Variable Interaction Matrix:*

In this section, the interaction matrix is now updated at each iteration of the control law using the current measurement of the visual features and an estimation of the depth of each considered point. The depth can be obtained from the knowledge of a 3D model of the object [1]. Then a perturbation is applied to the end-effector position of the robot (Figure 5). The goal of the control is to keep the robot in the reference position. During the experiment, one feature go out of the image plane (feature with (U_4V_4) in Figure 8). Due to the disappearance of this feature, a discontinuity is produced in the control law (Figure 8b-c). This discontinuity is not enough to produce that the system becomes unstable. In The second experiment, the new formulation of invariant visual servoing with weighted features was used. In Figure 9(d), the weights of every feature can be seen. Some features go out or are near the border of the image plane (feature 3, 4). The system is stable even though one feature goes out of the image plane. The control law in this case is continuous (Figure 9(b)(c)).

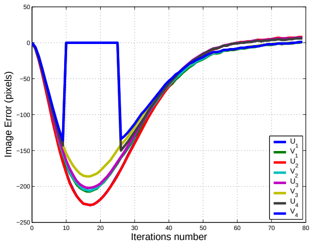


(a) Image error (b) Robot trajectory

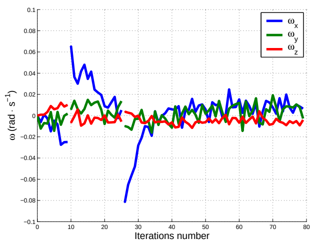
(c) Rotational velocity (d) Translational velocity

(e) Joint velocities (f) Weights

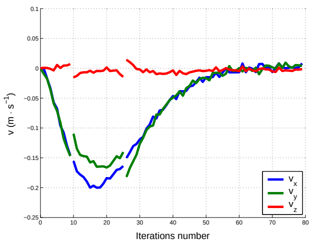
Fig. 7: Experimental results: Robot with an eye-in-hand camera perturbated with respect to its reference pose. The interaction matrix is constant.



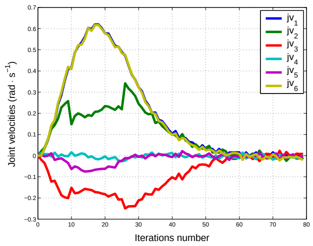
(a) Image error



(b) Rotational velocity

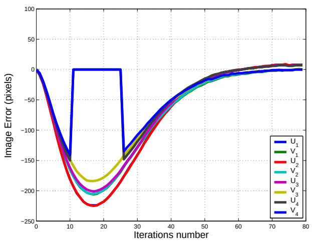


(c) Translational velocity

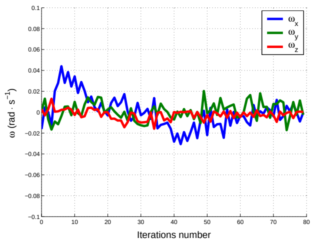


(d) Robot joints velocities

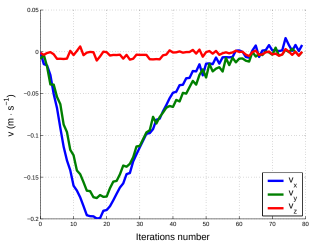
Fig. 8: Experimental results: Robot with an eye-in-hand camera perturbated with respect to its reference pose. The interaction matrix is update at each iteration of the control.



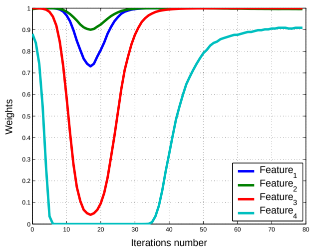
(a) Image error



(b) Rotational velocity



(c) Translational velocity



(d) Robot joint velocities

Fig. 9: Experimental results: Robot with an eye-in-hand camera perturbated with respect to its reference pose. The interaction matrix is update at each iteration of the control.

VII. Conclusions

In this paper, a possible solution for the singularities produced in vision-based control by the appearance/disappearance of image features has been presented and tested. The new approach can be used to improve both standard image-based and intrinsic-free visual servoing approaches. Contrary to the approaches based on constrained movements of the camera to ensure the visibility of features during the control and on zooming to keep it, the solution presented does not constrain the camera movements and allows us to control the zoom with another purpose. The camera trajectory seems to be closer to the optimal one than the one produced by the other approaches.

References

[1] D. Dementhon and L. Davis Modelbased object pose in 25 lines of code. In *Int. Journal of Computer Vision*, 15(1/2):123–141, June 1995.

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

[3] S. Benhimane and E. Malis Vision-based control with respect to planar and non-planar objects using a zooming camera In *IEEE International Conference on Advanced Robotics*, Coimbra, Portugal, July 2003.

[4] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In D. Kriegman, G. Hager, and A. Morse, editors, *The confluence of vision and control*, volume 237 of *LNCIS Series*, pages 66–78. Springer Verlag, 1998.

[5] K. Hashimoto. *Visual Servoing: Real Time Control of Robot manipulators based on visual sensory feedback*, volume 7 of *World Scientific Series in Robotics and Automated Systems*. World Scientifi c Press, Singapore, 1993.

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

[7] E. Malis. Visual servoing invariant to changes in camera intrinsic parameters. In *International Conference on Computer Vision*, volume 1, pages 704–709, Vancouver, Canada, July 2001.

[8] E. Malis Stability Analysis of Invariant Visual Servoing and Robustness to Parametric Uncertainties. *Second Joint CSS/RAS International Workshop on Control Problems in Robotics and Automation*, Las Vegas, Nevada, December, 2002.

[9] E. Malis. Vision-based control invariant to camera intrinsic parameters: stability analysis and path tracking. In *IEEE International Conference on Robotics and Automation*, volume 1, Washington, D.C., USA, May 2002.

[10] Y. Mezouar, F. Chaumette Path Planning For Robust Image-based Control In *IEEE Trans. on Robotics and Automation*, 18(4):534-549, August 2002.

[11] Y. Mezouar, F. Chaumette Optimal Camera Trajectory with Image-Based Control *International Journal of Robotics Research*, 22(10-11):781-803, October-November 2003.

[12] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control: the Task Function Approach*, volume 22 of *Oxford Engineering Science Series*. Clarendon Press, Oxford, UK, 1991.