VISUAL SERVOING TECHNIQUES FOR CONTINUOUS NAVIGATION OF A MOBILE ROBOT

N. García, O. Reinoso, J.M. Azorín
Dept. Ingeniería de Sistemas Industriales. Universidad Miguel Hernández.
Avd. de la Universidad s/n. Edif. Torreblanca. 03202 Elche, Spain

E. Malis
Institut National de Recherche en Informatique et Automatique
2004, route des Lucioles - B.P. 93, 06902 Sophia Antipolis Cedex, France.
R. Aracil
Departamento de Automtica, Electrnica e Informtica Industrial Universidad Politcnica de Madrid.
ETSII c/ Jos Gutirrez Abascal, 2 E-28006 Madrid, Spain.

Keywords: Visual servoing, mobile robots, navigation, control.

Abstract: A new method to control the navigation of a mobile robot which is based on visual servoing techniques is presented. The new contribution of this paper could be divided in two aspects: the first one is the solution of the problem which takes place in the control law when features appear or disappear from the image plane during the navigation; and the second one is the way of providing to the control system the reference path that must be followed by the mobile robot. The visual servoing techniques used to carry out the navigation are the image-based and the intrinsic-free approaches. Both are independent of calibration errors which is very useful since it is so difficult to get a good calibration in this kind of systems. Also, the second technique allows us to control the camera in spite of the variation of its intrinsic parameters. So, it is possible to modify the zoom of the camera, for instance to get more details, and drive the camera to its reference position at the same time. An exhaustive number of experiments using virtual reality worlds to simulate a typical indoor environment have been carried out.

1 INTRODUCTION

The framework for robot navigation is based on pre-recorded image features obtained during a training walk. Then, we want that the mobile robot repeat the same walk by means of image-based and invariant visual servoing techniques in a *continuous way*. In this point, the question is *what does a continuous way mean?*. Well, the answer to this question will be developed in the following sections of the paper but in advance the concept *continuous way* is referred to assure the continuity of the control law with the appearance/disappearance of image features during the control task.

There are some references about using pre-recorded images to control the navigation of a mobile robot but all of these applications use them in a different way. For instance, in (Matsumoto et al., 1996) the control system proposed for the navigation of a mobile robot is based on a visual representation of the route called *"View-Sequenced Route Representation(VSRR)"*. After the *VSRR* was acquired, the robot repeat the walk using an appearance-based method to control it by comparing the current view with pre-recorded one (*VSRR*).

There are some applications using visual servoing techniques for navigation of a mobile robot. In (R. Swain-Oropeza, 1997), the application of a visual servoing approach to a mobile robot which must execute coordinate motions in a known indoor environment is presented. The main contribution of this paper was the execution of a path expressed as sequence of several basic motions(like Go to an object, Follow a wall, Turn around a corner,...):each one is a visually-guide movement. In our case, the approximation proposed to the navigation is totally different in the way of dealing with the features which go in/out of the image plane during the path and similar to some references, commented above, in the way of specifying the path to be followed by the robot.

2 AUTONOMOUS NAVIGATION USING VISUAL SERVOING TECHNIQUES

The key idea of this method is to divide the autonomous navigation in two stages: the first one is the *training step* and the second one is the *autonomous*

García N., Reinoso O., M. Azorín J., Malis E. and Aracil R. (2004).
VISUAL SERVOING TECHNIQUES FOR CONTINUOUS NAVIGATION OF A MOBILE ROBOT.
In *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*, pages 343-348
DOI: 10.5220/0001134903430348
Copyright © SciTePress

navigation step. During the *training step*, the robot is human commanded via radio link or whatever interface and every sample time the robot acquires an image, computes the features and stores them in memory. Then, from near its initial position, the robot repeat the same walk using the reference features acquired during the *training step*. The current features and a visual servoing approach is used to minimize the error between current and reference features.

2.1 Control law

To carry out the autonomous navigation of robot, the image-based and invariant visual servoing approaches was used (Hutchinson et al., 1996)(Malis and Cipolla, 2000). Both approaches are based on the selection of a set $\mathbf{s}$ of visual features or a set $\mathbf{q}$ of invariant features that has to reach a desired value $\mathbf{s}^*$ or $\mathbf{q}^*$. Usually, $\mathbf{s}$ is composed of the image coordinates of several points belonging to the considered target and $\mathbf{q}$ is computed as the projection of $\mathbf{s}$ in the invariant space calculated previously. In the case of our navigation method, $\mathbf{s}^*$ or $\mathbf{q}^*$ is variable with time since in each sample time the reference features is updated with the desired trajectory of $\mathbf{s}$ or $\mathbf{q}$ stored in the robot memory in order to indicate the path to be followed by the robot.

To simplify in this section, the formulation presented is only referred to image-based visual servoing. All the formulation of this section can be applied directly to the invariant visual servoing approach changing $\mathbf{s}$ by $\mathbf{q}$. The visual task function(Samson et al., 1991) is defined as the regulation of an error function:

$$\mathbf{e} = \mathbf{C}(\mathbf{s} - \mathbf{s}^*(t)) \tag{1}$$

The derivative of the task function,considering $\mathbf{C}$ constant, will be:

$$\dot{\mathbf{e}} = \mathbf{CLv} - \mathbf{C}\,\dot{\mathbf{s}}^* \tag{2}$$

where $\mathbf{v} = (\mathbf{V}^\mathbf{T}, \omega^\mathbf{T})$ is the camera velocity screw and $\mathbf{L}$ is the interaction matrix.

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

$$\dot{\mathbf{e}} = -\lambda\,\mathbf{e} \qquad \text{so} \qquad \mathbf{CLv} = -\lambda\,\mathbf{e} + \mathbf{C}\,\dot{\mathbf{s}}^* \tag{3}$$

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

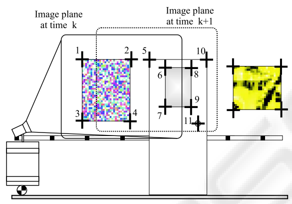
$$\mathbf{v} = -\lambda\,(\mathbf{CL})^{-1}\mathbf{e} + (\mathbf{CL})^{-1}\mathbf{C}\,\dot{\mathbf{s}}^* \tag{4}$$

if $\mathbf{C}$ is setting to $\mathbf{L}^+$, then $(\mathbf{CL}) > 0$ and the task function converge to zero and, in the absence of local minima and singularities, so does the error $s - s^*$. Finally substituting $\mathbf{C}$ by $\mathbf{L}^+$ in equation (8), we obtain the expression of the camera velocity that is sent to the robot controller:

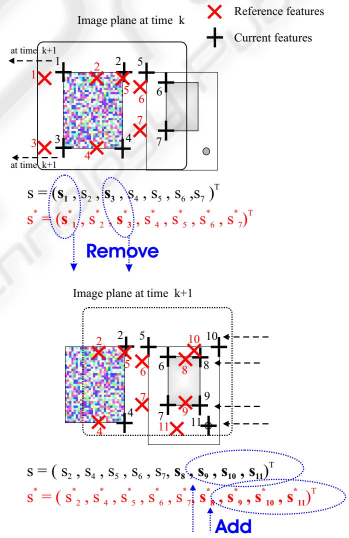
$$\mathbf{v} = -\lambda\,\mathbf{L}^+\,(\mathbf{s} - \mathbf{s}^*(t)) + \mathbf{L}^+\,\dot{\mathbf{s}}^* \tag{5}$$

3 DISCONTINUITIES IN VISUAL NAVIGATION

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. The navigation of a mobile robot is a typical example where this kind of problems are produced.



(a) Croquis of the autonomous navigation of the robot



(b) Appearance/Disappearance of image features from time k to k+1

Figure 1: Navigation of a mobile robot controlled by visual servoing techniques.

During autonomous navigation of the robot, some features appear or disappear from the image plane so they will must be added to or removed from the visual error vector (Figure 1). This change in the error vector produces 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 pseudoinverse of interaction matrix.

In the case of using the invariant visual servoing approach to control the robot, the effect produced by the appearance/disappearance of features could be more important since the invariant space $\mathcal{Q}$ used to compute the current and the reference invariant points($\mathbf{q}, \mathbf{q}^*$) changes with features(Malis, 2002c).

4 CONTINUOUS CONTROL LAW FOR NAVIGATION

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 based on weighted features is presented.

4.1 Weighted features

The question is *What is a weighted feature?*. The answer is obvious *it's a feature which is pre-multiplied by a factor called weight.* This weight is going to be used in order to anticipate in some way the possible discontinuities produced in the control law by the appearance/disappearance of the 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:

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

where $0 < \alpha < 0.5$ and $0 < \beta < 0.5$. if the conditions (6)are verified then the following conditions are truth too:

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

where $0 < \alpha < 0.5$ and $0 < \beta < 0.5$. 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 $\gamma_y(x)$ ($\gamma_u^i = \gamma_y(u_i)$ and $\gamma_v^i = \gamma_y(v_i)$ respectively). 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).

4.2 Smooth Task function

Suppose that n matched points are available in the current image and in the reference features stored. Everyone of these points(current and reference) will have a weight γ_{uv}^i which can be computed as it's shown in the previous subsection 4.1. With them and their weights, a task function can be built (Samson et al., 1991):

$$\mathbf{e} = \mathbf{CW}\,(\mathbf{s} - \mathbf{s}^*(t)) \tag{6}$$

where $\mathbf{W}$ is a $(2n \times 2n)$ diagonal matrix where its elements are the weights γ_{uv}^i of the current features multiplied by the weights of the reference features.

The derivate of the task function,considering $\mathbf{C}$ constant, will be:

$$\dot{\mathbf{e}} = \mathbf{CW}\,\mathbf{Lv} - \mathbf{CW}\dot{\mathbf{s}}^* + \mathbf{C}\dot{\mathbf{W}}\,(\mathbf{s} - \mathbf{s}^*) \tag{7}$$

A simple control law can be obtained by imposing the exponential convergence of the task function to zero ($\dot{\mathbf{e}} = -\lambda\,\mathbf{e}$):

$$\mathbf{CWLv} - \mathbf{CW}\dot{\mathbf{s}}^* + \mathbf{C}\dot{\mathbf{W}}(\mathbf{s} - \mathbf{s}^*) = -\lambda\mathbf{e} \tag{8}$$

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:

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

Setting $\mathbf{C} = (\mathbf{WL})^+$, if $(\mathbf{CWL}) > 0$ then the task function converges to zero and, in the absence of local minima and singularities, so does the error $\mathbf{s} - \mathbf{s}^*$. Finally substituting $\mathbf{C}$ by $(\mathbf{WL})^+$ in equation (9), we obtain the expression of the camera velocity that is sent to the robot controller:

$$\mathbf{v} = -\lambda\,(\mathbf{WL})^+\mathbf{W}(\mathbf{s} - \mathbf{s}^*(t)) + (\mathbf{WL})^+\mathbf{W}\dot{\mathbf{s}}^* \tag{10}$$

4.3 Invariant visual servoing with weighted features

The theoretical background about invariant visual servoing can be extensively found in (Malis, 2002b)(Malis, 2002c). 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:

$$\boldsymbol{\gamma_i} = \sqrt{\frac{n}{\sum_{i=1}^{n} (\boldsymbol{\gamma_{uv}^i})^2}} \cdot \boldsymbol{\gamma_{uv}^i} \tag{11}$$

The weights γ_{uv}^i defined in the previous subsection are *redistributed* in order to have $\sum \gamma_\mathbf{i}^{\mathbf{2}} = n$. 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 $\mathbf{p}_i^{\gamma_i} = \gamma_i\ \mathbf{p}_i$. Using all the weighted points we can compute the following symmetric (3×3) matrix:

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

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:

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

where $\mathbf{S}_m^{\gamma_i}$ is a symmetric matrix which is does not directly depend on the camera parameters. 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 (13) 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 $\mathcal{Q}^{\gamma_i}$:

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

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 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:

$$\mathbf{q}_i^* = \mathbf{T}_{p^*}^{\gamma_i\,-1}\mathbf{p}_i^* = \mathbf{T}_{m^*}^{\gamma_i\,-1}\mathbf{m}_i^* \tag{17}$$

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 converse is true even if the intrinsic parameters change during the servoing.

4.3.1 Control in the weighted invariant space

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:

$$\dot{\boldsymbol{s}} = \boldsymbol{L}\ \boldsymbol{v} \tag{18}$$

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 (Malis, 2002a) but the term $\boldsymbol{C}_{\boldsymbol{i}}^{\boldsymbol{\gamma_i}}$ 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 (10) 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.

5 EXPERIMENTS IN A VIRTUAL INDOOR ENVIRONMENT

Exhaustive experiments have been carried out using a virtual reality tool for modelling an indoor environment. To make more realistic simulation, errors in intrinsic and extrinsic parameters of the camera mounted in the robot have been considered. An estimation $\widehat{\mathbf{K}}$ of the real matrix $\mathbf{K}$ has been used with an error of 25% in focal length and a deviation of 50 pixels in the position of the optical center. Also an estimation $\widehat{\mathbf{T}}_{RC}$ of the camera pose respect to the robot frame has been computed with a rotation error of $u\theta = [3.75\ 3.75\ 3.75]^T\ degrees$ and translation error of $t = [2\ 2\ 0]^T\ cm$. We are only going to present the results of one experiment where image based and invariant visual servoing and the new formulation with weighted features are used to control the autonomous navigation of an holonomic mobile robot. In Figure 3, the control signals sent to the robot controller using the image-based and invariant visual servoing approaches are shown. In Figure 3 (b,d,f,h), details of the control law, where the discontinuities can be clearly appreciated, are presented. To show the improvements of the new formulation presented in this paper, the control law using the image-based and invariant visual servoing with weighted features can be seen in Figure 4. The same details of the control law shown in Figure 3 (b,d,f,h) are presented in Figure 4 (b,d,f,h). Observing both figures, the improvements respect to the continuity of the control law are self-evident.

Some experiments using a filter to avoid the effects of the discontinuities in the control law have been carried out too. A 8th order butterworth lowpass digital filter has been designed to overpass the discontinuities produced by the appearance/disappearance of image features. In Figure 2 (a,b,c), the continuity of the control law using a filter or weighted features and the discontinuity with the classical image-based approach can be seen. However the trajectory of the mobile robot is not the same using weighted features than filtering the control law (Figure 2 (d,e)). Furthermore the trajectory error when the control law is filtered needs more time to stabilize than when the formulation with weighted features is used.

6 CONCLUSION

In this paper the originally formulation of two visual servoing approaches, which avoids the discontinuities in the control law when features go in/out of the image plane during the control task, is presented and tested by several experiments in a virtual indoor environment. The results presented corroborate that the new approach to the problem works better than a simple filter of the control signals. The validation of this results with a real robot is on the way by using a B21r mobile robot from iRobot company. As a future work, it would be interesting to test another kind of weighted functions.

REFERENCES

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

Malis, E. (2002a). Stability analysis of invariant visual servoing and robustness to parametric uncertainties. In *Second Joint CSS/RAS International Workshop on Control Problems in Robotics and Automation*, Las Vegas, Nevada.

Malis, E. (2002b). A unified approach to model-based and model-free visual servoing. In *European Conference on Computer Vision*, volume 2, pages 433–447, Copenhagen, Denmark.

Malis, E. (2002c). Vision-based control invariant to camera intrinsic parameters: stability analysis and path tracking. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 217–222, Washington, USA.

Malis, E. and Cipolla, R. (2000). Self-calibration of zooming cameras observing an unknown planar structure. In *Int. Conf. on Pattern Recognition*, volume 1, pages 85–88, Barcelona, Spain.

Matsumoto, Y., Inaba, M., and Inoue, H. (1996). Visual navigation using view-sequenced route representation. In *IEEE International Conference on Robotics and Automation (ICRA'96)*, volume 1, pages 83–88, Minneapolis, USA.

R. Swain-Oropeza, M. D. (1997). Visually-guided navigation of a mobile robot in a structured environment. In *International Symposium on Intelligent Robotic Systems (SIRS'97)*, Stockholm, Sweden.

Samson, C., Le Borgne, M., and Espiau, B. (1991). *Robot Control: the Task Function Approach.* volume 22 of Oxford Engineering Science Series.Clarendon Press., Oxford, UK, 1st edition.

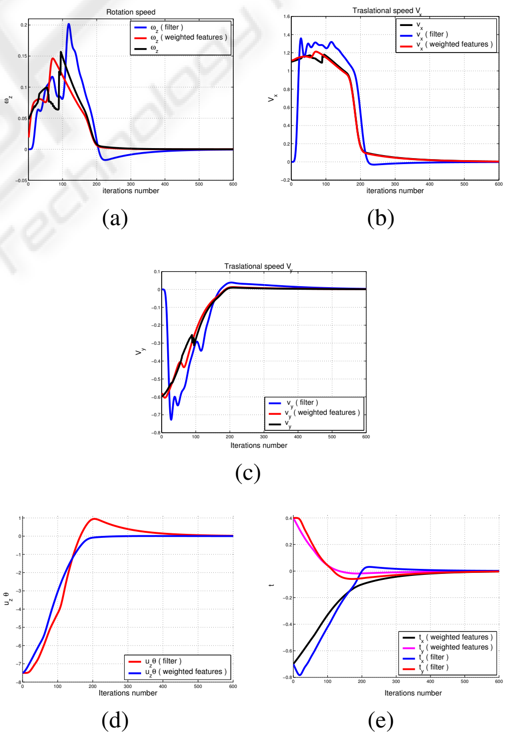
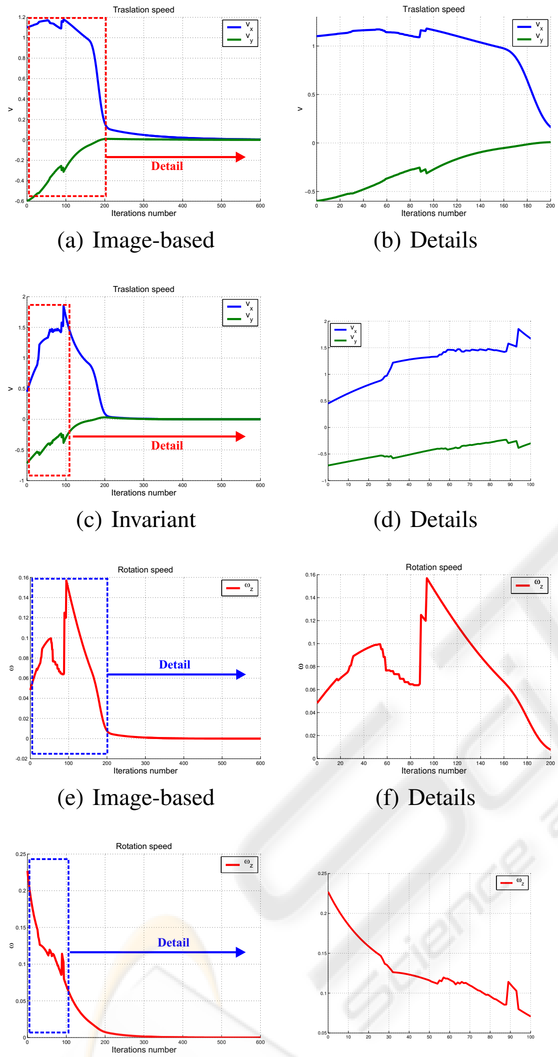
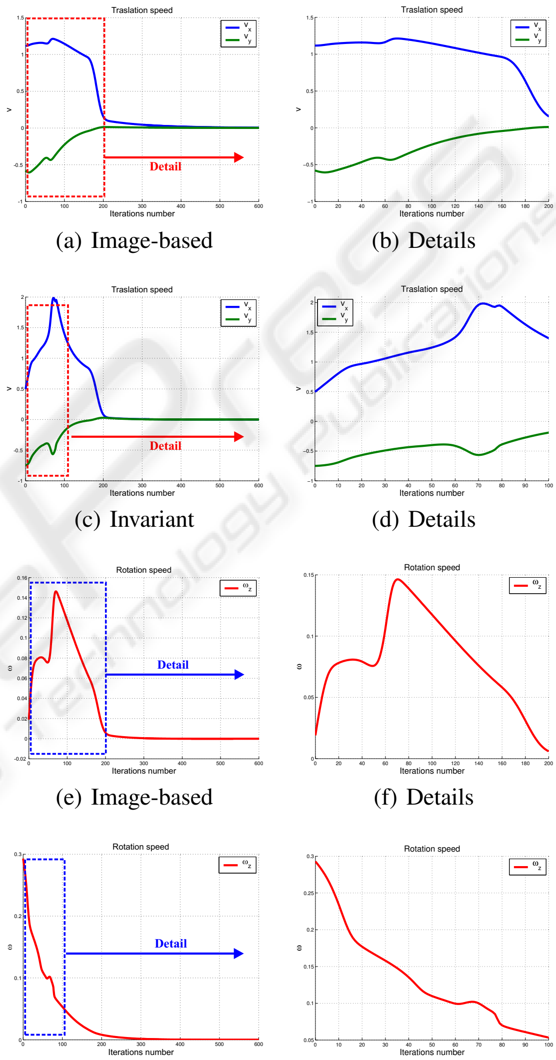


Figure 2: Image-based visual servoing approach: classical, with weighted features and using a filter. The translation and rotation errors are measured respectively in m and deg, while speeds are measured respectively in $\frac{m}{s}$ and $\frac{deg}{s}$.



(a) Image-based (b) Details (c) Invariant (d) Details (e) Image-based (f) Details (g) Invariant (h) Details

Figure 3: Control law: Image-based(a,b,e,f) and invariant(c,d,g,h) visual servoing. The translation and rotation speeds are measured respectively in $\frac{m}{s}$ and $\frac{deg}{s}$



(a) Image-based (b) Details (c) Invariant (d) Details (e) Image-based (f) Details (g) Invariant (h) Details

Figure 4: Control law: Image-based(a,b,e,f) and invariant(c,d,g,h) visual servoing with weighted features. The translation and rotation speeds are measured respectively in $\frac{m}{s}$ and $\frac{deg}{s}$