S. Benhimane
E. Malis
INRIA
2004, route des Lucioles – B.P. 93,
06902 Sophia Antipolis Cedex, France

Homography-based 2D Visual Tracking and Servoing

Abstract

The objective of this paper is to propose a new homography-based approach to image-based visual tracking and servoing. The visual tracking algorithm proposed in the paper is based on a new efficient second-order minimization method. Theoretical analysis and comparative experiments with other tracking approaches show that the proposed method has a higher convergence rate than standard first-order minimization techniques. Therefore, it is well adapted to real-time robotic applications. The output of the visual tracking is a homography linking the current and the reference image of a planar target. Using the homography, a task function isomorphic to the camera pose has been designed. A new image-based control law is proposed which does not need any measure of the 3D structure of the observed target (e.g. the normal to the plane). The theoretical proof of the existence of the isomorphism between the task function and the camera pose and the theoretical proof of the stability of the control law are provided. The experimental results, obtained with a 6 d.o.f. robot, show the advantages of the proposed method with respect to the existing approaches.

KEY WORDS—visual tracking, visual servoing, efficient second-order minimization, homography-based control law

1. Introduction

Vision-based control offers a wide spectrum of application possibilities entailing the use of computer vision and control theories : manipulation, medical robotics, automatic driving, observation and surveillance by aerial robots, etc. The achievement of such complex applications needs the integration of visual tracking and visual servoing techniques. In this paper, we describe our contributions to template-based visual tracking algorithms and model-free vision-based control techniques. These techniques are integrated in a unifying framework leading to generic, flexible and robust systems that can be used for a variety of robotic applications.

The International Journal of Robotics Research
Vol. 26, No. 7, July 2007, pp. 661–676
DOI: 10.1177/0278364907080252

1.1. Visual tracking

Visual tracking is the core of a vision-based control system in robotics (Hutchinson et al. 1996). When considering real-time robotic applications, the main requirements of a tracking algorithm are efficiency, accuracy and robustness. Visual tracking methods can be classified into two main groups. The first group is composed of methods that track local features such as line segments, edges or contours across the sequence (Isard and Blake 1996; Torr anbd Zisserman 1999; Drummond and Cipolla 1999). These techniques are sensitive to feature detection and cannot be applied to complex images that do not contain special sets of features to track. The second group is composed of methods that only make use of image intensity information. These methods estimate the movement, the deformation or the illumination parameters of a reference template between two frames by minimizing an error measure based on image brightness. Many approaches have been proposed to find the relationship between the measured error and the parameters variation. Some methods compute (learn) this relationship in an off-line processing stage: difference decomposition (Gleicher 1997; Jurie and Dhome 2002), active blobs (Sclaroff and Isidoro 1998), active appearance models (Cootes et al. 1998). Although these methods are a possible solution to the problem, they cannot be used in some real-time robotic applications where the learning step cannot be processed online. For example, consider a robot moving in an unknown environment that needs to instantaneously track an object suddenly appearing in its field of view. Alternatively, there are methods that minimize the sum-of-squared-differences (SSD) between the reference template and the current image using parametric models (Lucas and Kanade 1981; Hager and Behumeur 1998; Shum ans Szeliski 2000; Baker and Matthews 2001). Many minimization algorithms could be used to estimate the transformation parameters. Theoretically, the Newton method has the highest local convergence rate since it is based on a second-order Taylor series of the SSD. However, the Hessian computation in the Newton method is time consuming. In addition, if the Hessian is not positive definite, convergence problems can occur. In this paper, we propose to use an efficient second-order minimization method (ESM) (Malis

2004) to solve the problem. The ESM method has a high convergence rate like the Newton method, but the ESM does not need to compute the Hessian. Due to its generality, the ESM algorithm has been successfully used to build an efficient visual tracking algorithm in Benhimane and Malis (2004). Theoretical analysis and comparative simulations with other tracking approaches show that the method has a higher convergence rate than other minimization techniques. Consequently, the ESM algorithm tracks with higher inter-frame movements and it is well-adapted to real-time visual servoing applications.

1.2. Visual Servoing

Visual servoing uses the visual information tracked by one or multiple cameras (Hashimoto 1993; Hutchinson et al. 1996) in order to control a robot with respect to a target. This robotic task can be considered as the regulation of a task function $\mathbf{e}$ that depends on the robot configuration and the time (Samson et al. 1991). In this paper, we consider eye-in-hand visual servoing approaches that use the minimum amount of 3D information about the observed target. Our objective is to design a visual servoing method that does not need any measure of the 3D structure of the target and that only needs the reference image and the current image to compute the task function. In the literature, the visual servoing methods are generally classified as follows:

- 3D visual servoing: the task function is expressed in the Cartesian space, i.e. the visual information acquired from the two images (the reference and the current images) is used to explicitly reconstruct the pose (the translation and the rotation in Cartesian space) of the camera (see, for example, Wilson et al. 1996; Martinet et al. 1997; Basri et al. 1998; Taylor et al. 2000; Malis and Chaumette 2002). The camera translation (up to a scale factor) and the camera rotation can be estimated through the Essential matrix (Longuet-Higgins 1981; Hartley 1992; Faugeras 1993). However, the Essential matrix cannot be estimated when the target is planar or when the motion performed by the camera between the reference and the current pose is a pure rotation. For these reasons, it is better to estimate the camera translation (up to a scale factor) and the camera rotation using a homography matrix (Malis et al. 2000).
- 2D visual servoing: the task function is expressed directly in the image, i.e. these visual servoing methods do not need the explicit estimation of the pose error in the Cartesian space (see, for example, Espiau et al. 1992; Chaumette 2004). A task function isomorphic to the camera pose is built. As far as we know, except for some special "ad hoc" target (Cowan and Chang 2002), the isomorphism is generally supposed true without any formal proof. The real existence of the isomorphism avoids situations where the task function is null and the camera is not well positioned (Chaumette 1998). In general, the task function is built using simple image features such as the coordinates of interest points. Since the control takes place in the image, the target has much more chance to remain visible in the image.
- 2D 1/2 visual servoing: the task function is expressed of remaining both in the Cartesian space and in the image, i.e. the rotation error is estimated explicitly and the translation error is expressed in the image (see, for example, Malis et al. 1999; Deguchi 1998). These visual servoing approaches make it possible not only to perform the control in the image but also it is possible to demonstrate the stability and the robustness of the control law (Malis and Chaumette 2002).

We notice that, for any of the previous methods, we need a measure (on-line or off-line) of some 3D information concerning the observed target. In the 2D 1/2 visual servoing and 3D visual servoing, the pose reconstruction using the homography estimation is not unique (two different solutions are possible). In order to choose the good solution, it is necessary to have an estimate of the normal vector to the target plane. If the estimate is very poor we could choose the wrong solution. In the 2D visual servoing, when considering for example points as features, the corresponding depths are necessary to have a stable control law (Malis and Rives 2003). The 3D information can be obtained on-line. However, the price to pay is a time consuming estimation step. For example, when the target is planar, many images are needed to obtain a precise estimation of the normal to the plane.

In this paper, we present a new 2D visual servoing method that makes it possible to control the robot by building a task function isomorphic to the camera pose in the Cartesian space. We have demonstrated that isomorphism exists between a task function $\mathbf{e}$ (measured using the homography that matches the reference target plane image and the current one) and the camera pose in the Cartesian space, i.e. the task function $\mathbf{e}$ is null if and only if the camera is back to the reference pose. Contrary to the standard 2D visual servoing, we have demonstrated that we do not need to measure any 3D information in order to guarantee the stability of the control. The computation of the control law is quite simple (we do not need either the estimation of an interaction matrix or the decomposition of the homography) and, similarly to the task function, the control law does not need any measure of 3D information on the observed target.

For simplicity, in order to introduce our approach, in this paper we consider planar targets with unknown 3D information (i.e. the normal vector to the target plane is unknown). The generalization of the new approach to non-planar targets is straightforward since a homography related to a virtual plane can also be measured if the target is non-planar (Malis et al. 2000).

2. Modeling and Notation

As already mentioned in the introduction, we consider eye-in-hand visual servoing methods. In other words, the robot is controlled in order to position the current camera frame $\mathcal{F}$ to the reference camera frame $\mathcal{F}^*$. We suppose that the only available information are an image $\mathcal{I}^*$ of the scene at the reference pose and a current image $\mathcal{I}$ of the observed scene (acquired in real time).

2.1. Perspective Projection

Let $\mathcal{P}$ be a point in the 3D space. Its 3D coordinates are $\boldsymbol{\mathcal{X}}^* = [X^* \; Y^* \; Z^*]^\top$ in the reference frame $\mathcal{F}^*$. Using a perspective projection model, the point projects on a virtual plane perpendicular to the optical axis and distant one meter from the projection center in the point $\mathbf{m}^* = [x^* \; y^* \; 1]^\top$ verifying:

$$\mathbf{m}^* = \frac{1}{Z^*}\boldsymbol{\mathcal{X}}^*. \tag{1}$$

We call $\mathcal{I}_m^*$ the reference image in normalized coordinates. A pinhole camera performs a perspective projection of the point $\mathcal{P}$ on the image plane $\mathcal{I}^*$ [11]. The image coordinates $\mathbf{p}^* = [u^* \; v^* \; 1]^\top$ can be obtained from the normalized coordinates with an affine transformation:

$$\mathbf{p}^* = \mathbf{K}\mathbf{m}^* \tag{2}$$

where the camera intrinsic parameters matrix $\mathbf{K}$ can be written as follows:

$$\mathbf{K} = \begin{bmatrix} f & fs & u_0 \\ 0 & fr & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where f is the focal length in pixels, s represents the default of orthogonality between the image frame axis, r is the aspect ratio and $[u_0 \; v_0]$ are the coordinates of the principal point (in pixels).

Let $\mathbf{R} \in \mathbb{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$ be respectively the rotation and the translation between the two frames $\mathcal{F}$ and $\mathcal{F}^*$. In the current frame $\mathcal{F}$, the point $\mathcal{P}$ has the following coordinates $\boldsymbol{\mathcal{X}} = [X \; Y \; Z]^\top$ and we have:

$$\boldsymbol{\mathcal{X}} = \mathbf{R}\boldsymbol{\mathcal{X}}^* + \mathbf{t}. \tag{4}$$

Let $\mathbf{u} = [u_x \; u_y \; u_z]^\top$ be the unit vector corresponding to the rotation axis and θ ($\theta \in]-\pi, \pi[$) be the rotation angle. Setting $\mathbf{r} = \theta\mathbf{u}$, we have:

$$\mathbf{R} = \exp([\mathbf{r}]_\times) \tag{5}$$

where exp is the matrix exponential function and where the skew matrix $[\mathbf{r}]_\times$ is defined as follows:

$$[\mathbf{r}]_\times = \begin{bmatrix} 0 & -r_z & +r_y \\ +r_z & 0 & -r_x \\ -r_y & +r_x & 0 \end{bmatrix}. \tag{6}$$

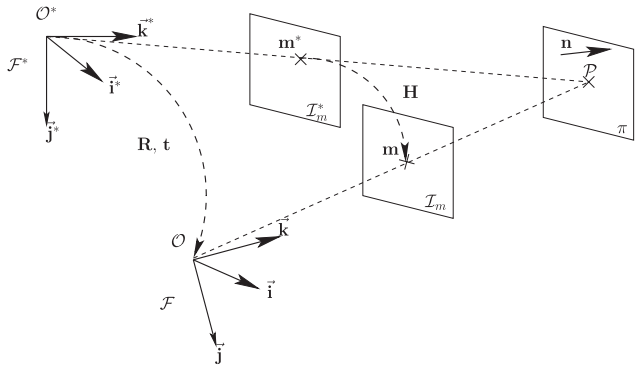


Fig. 1. Projection model and homography between two images of a plane

The point $\boldsymbol{\mathcal{X}}$ projects on the current normalized image $\mathcal{I}_m$ in $\mathbf{m} = [x \; y \; 1]^\top$ where:

$$\mathbf{m} = \frac{1}{Z}\boldsymbol{\mathcal{X}} \tag{7}$$

and projects on the current image $\mathcal{I}$ in $\mathbf{p} = [u \; v \; 1]^\top$ where:

$$\mathbf{p} = \mathbf{K}\mathbf{m}. \tag{8}$$

2.2. Homography Between Two Images of a Plane

Let us suppose that the point $\mathcal{P}$ belongs to a plane π. Let $\mathbf{n}^*$ be the normal vector to π expressed in the reference frame $\mathcal{F}^*$ and d^* is the distance (at the reference pose) between the plane π and the center of projection. If we choose $\mathbf{n}^*$ such that:

$$\left\|\mathbf{n}^*\right\| = \sqrt{\mathbf{n}^{*\top}\mathbf{n}^*} = \frac{1}{d^*} \tag{9}$$

then, we can write:

$$\mathbf{n}^{*\top}\boldsymbol{\mathcal{X}}^* = 1. \tag{10}$$

By using equations (1), (4), (7) and (10), we obtain the following relationship between $\mathbf{m}$ and $\mathbf{m}^*$:

$$\frac{Z}{Z^*}\mathbf{m} = \mathbf{H}\mathbf{m}^* \tag{11}$$

where the homography matrix $\mathbf{H}$ can be written as follows:

$$\mathbf{H} = \mathbf{R} + \mathbf{t}\mathbf{n}^{*\top}. \tag{12}$$

By using equations (2), (8) and (11), we obtain the following relationship between $\mathbf{p}$ and $\mathbf{p}^*$:

$$\frac{Z}{Z^*}\mathbf{p} = \mathbf{G}\mathbf{p}^* \tag{13}$$

where the matrix $\mathbf{G}$ can be written as follows:

$$\mathbf{G} = \mathbf{K}\mathbf{H}\mathbf{K}^{-1}. \tag{14}$$

Given two images $\mathcal{I}$ and $\mathcal{I}^*$ of a planar target, it is possible to compute the homography matrix $\mathbf{G}$ up to a scale factor.

We choose the scale factor of the matrices $\mathbf{G}$ and $\mathbf{H}$ such that the determinants of $\mathbf{H}$ and $\mathbf{G}$ are equal to 1. Then the matrices $\mathbf{H}$ and $\mathbf{G}$ belong to the Special Linear group $\mathbb{SL}(3)$ of dimension 3. This choice is well justified since $\det(\mathbf{H}) \leq 0$ (or $\det(\mathbf{G}) \leq 0$) happens only when the point $\mathcal{O}$ passes though the plane π.

Given the matrices $\mathbf{G}$ and $\mathbf{K}$, we compute the matrix $\mathbf{H}$ up to a scale factor. Decomposing the matrix $\mathbf{H}$ to obtain the rotation $\mathbf{R}$ and the translation $\mathbf{t}$ has more than one solution [11]. In general, given the matrix $\mathbf{K}$, four solutions $\{\mathbf{R}_i,\ \mathbf{t}_i,\ \mathbf{n}_i^*\}$, $i \in \{1,\ 2,\ 3,\ 4\}$ are possible but only two are physically admissible. An approximation of the real normal vector $\mathbf{n}^*$ to the target plane makes it possible to choose the good pose.

The matrix $\mathbf{G} = (g_{ij})$ defines a projective transformation in the image. A group action $\mathbf{w}$ can be defined from $\mathbb{SL}(3)$ on $\mathbb{P}^2$:

$$\mathbf{w} : \mathbb{SL}(3) \times \mathbb{P}^2 \rightarrow \mathbb{P}^2. \tag{15}$$

For all $\mathbf{G} \in \mathbb{SL}(3)$, $\mathbf{w}(\mathbf{G})$ is a $\mathbb{P}^2$ automorphism:

$$\begin{array}{rcl} \mathbf{w}(\mathbf{G}) \ : \ \mathbb{P}^2 & \rightarrow & \mathbb{P}^2 \\ \mathbf{p}^* & \mapsto & \mathbf{p} = \mathbf{w}(\mathbf{G})(\mathbf{p}^*) \end{array} \tag{16}$$

such that:

$$\mathbf{p} = \mathbf{w}(\mathbf{G})(\mathbf{p}^*) = \begin{bmatrix} \frac{g_{11}\,u^* + g_{12}\,v^* + g_{13}}{g_{31}\,u^* + g_{32}\,v^* + g_{33}} \\ \frac{g_{21}\,u^* + g_{22}\,v^* + g_{23}}{g_{31}\,u^* + g_{32}\,v^* + g_{33}} \\ 1 \end{bmatrix}.$$

Let $\mathbf{I}$ be the identity matrix. We have the following properties:

- $\forall \mathbf{p} \in \mathbb{P}^2$:
$$\mathbf{w}(\mathbf{I})(\mathbf{p}) = \mathbf{p} \tag{17}$$
- $\forall \mathbf{p}^* \in \mathbb{P}^2$ and $\forall \mathbf{G}_1,\ \mathbf{G}_2 \in \mathbb{SL}(3)$:
$$\mathbf{w}(\mathbf{G}_1)(\mathbf{w}(\mathbf{G}_2)(\mathbf{p}^*)) = \mathbf{w}(\mathbf{G}_1) \circ \mathbf{w}(\mathbf{G}_2)(\mathbf{p}^*) \tag{18}$$
$$= \mathbf{w}(\mathbf{G}_1\mathbf{G}_2)(\mathbf{p}^*) \tag{19}$$
- $\forall \mathbf{G} \in \mathbb{SL}3$:
$$(\mathbf{w}(\mathbf{G}))^{-1} = \mathbf{w}(\mathbf{G}^{-1}). \tag{20}$$

2.3. The Image Model

A $(n \times m)$ image $\mathcal{I}^*$ can be considered as a $(n \times m)$ matrix containing the pixel intensities. The entry $\mathcal{I}^*(u^*, v^*)$ is the intensity of the pixel located at the line u^* and the column v^*. We suppose there exists a regular function I^*:

$$\begin{array}{rcl} I^* \ : \ \mathbb{P}^2 & \rightarrow & \mathbb{R} \\ \mathbf{p}^* \ = \ [u^*\ v^*\ 1]^\top & \mapsto & I^*(u^*, v^*) \end{array} \tag{21}$$

that verifies $\forall (u^*, v^*) \in \{1, 2, ..., n\} \times \{1, 2, ..., m\}$, we have $I^*(\mathbf{p}^*) = \mathcal{I}^*(u^*, v^*)$. For the non-integer values of (u^*, v^*), $I^*(\mathbf{p}^*)$ is obtained by interpolating $\mathcal{I}^*(u^*, v^*)$ where (u^*, v^*) are integer. In this paper, we suppose that the "image constancy assumption" is verified, i.e., the two projections $\mathbf{p}^*$ and $\mathbf{p}$ of the same 3D point $\mathcal{P}$ in the images $\mathcal{I}^*$ and $\mathcal{I}$ have the same intensity:

$$\mathcal{I}^*(\mathbf{p}^*) = \mathcal{I}(\mathbf{p}). \tag{22}$$

3. ESM Homography-based Visual Tracking

3.1. Problem Statement

In this section, we suppose that the object we aim to track is planar. The object is projected in the reference image $\mathcal{I}^*$ in some region of q pixels. This region is called the reference template. Since the object is supposed to be planar, there is a homography $\overline{\mathbf{G}}$ that transforms each pixel $\mathbf{p}_i^*$ of the reference pattern into its corresponding pixel in the current image $\mathcal{I}$. Tracking the reference template in the current image $\mathcal{I}$ consists in finding the projective transformation $\overline{\mathbf{G}} \in \mathbb{SL}(3)$ that transforms each pixel $\mathbf{p}_i^*$ of the reference pattern into its corresponding pixel in the current image $\mathcal{I}$, i.e. finding the homography $\overline{\mathbf{G}}$ such that $\forall i \in \{1,\ 2,\ ...,\ q\}$:

$$\mathcal{I}\left(\mathbf{w}\left(\overline{\mathbf{G}}\right)(\mathbf{p}_i^*)\right) = \mathcal{I}^*(\mathbf{p}_i^*) \tag{23}$$

Suppose that we have an approximation $\hat{\mathbf{G}}$ of $\overline{\mathbf{G}}$, the problem consists in finding an incremental transformation $\mathbf{G}(\mathbf{x})$ (where the (8×1) vector $\mathbf{x}$ contains a local parameterization of $\mathbb{SL}(3)$) such that the difference between the region of the image $\mathcal{I}$ (transformed with the composition $\mathbf{w}(\hat{\mathbf{G}}) \circ \mathbf{w}(\mathbf{G}(\mathbf{x}))$) and the corresponding region in the image $\mathcal{I}^*$ is null. Tracking consists in finding the vector $\mathbf{x}$ such that $\forall i \in \{1,\ 2,\ ...,\ q\}$, we have:

$$y_i(\mathbf{x}) = \mathcal{I}\left(\mathbf{w}(\hat{\mathbf{G}}) \circ \mathbf{w}(\mathbf{G}(\mathbf{x}))(\mathbf{p}_i^*)\right) - \mathcal{I}^*(\mathbf{p}_i^*) = 0. \tag{24}$$

Let $\mathbf{y}(\mathbf{x})$ be the $(q \times 1)$ vector containing the image differences:

$$\mathbf{y}(\mathbf{x}) = \begin{bmatrix} y_1(\mathbf{x}) & y_2(\mathbf{x}) & ... & y_q(\mathbf{x}) \end{bmatrix}^\top. \tag{25}$$

Then, the problem consists in finding $\mathbf{x} = \mathbf{x}_0$ verifying:

$$\mathbf{y}(\mathbf{x}_0) = \mathbf{0}. \tag{26}$$

Since the matrix $\hat{\mathbf{G}} \in \mathbb{SL}(3)$, i.e. $\det(\hat{\mathbf{G}}) = 1$ by construction, it is evident that the solution to the problem verifies:

$$\mathbf{G}(\mathbf{x}_0) = \hat{\mathbf{G}}^{-1}\overline{\mathbf{G}}. \tag{27}$$

The system (26) is generally nonlinear and many methods could be used to solve the problem. However, due to real-time constraints the problem is often solved by using an iterative minimization after linearizing the image signal with respect to the transformation parameters.

3.2. System Linearization

Let the $(q \times 8)$ matrix $\mathbf{J}(\mathbf{x})$ be the Jacobian matrix, i.e it is the gradient of the vector $\mathbf{y}(\mathbf{x})$ with respect to the vector $\mathbf{x}$:

$$\mathbf{J}(\mathbf{x}) = \nabla_{\mathbf{x}}\mathbf{y}(\mathbf{x}). \tag{28}$$

Let the $(q \times 8)$ matrix $\mathbf{M}(\mathbf{x}_1, \mathbf{x}_2)$ defined as:

$$\mathbf{M}(\mathbf{x}_1, \mathbf{x}_2) = \nabla_{\mathbf{x}_1} \left(\mathbf{J}(\mathbf{x}_1)\mathbf{x}_2\right). \tag{29}$$

It is possible to linearize the vector $\mathbf{y}(\mathbf{x})$ about $\mathbf{x} = \mathbf{0}$ using the second-order Taylor series approximation:

$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\,\mathbf{x} + \frac{1}{2}\mathbf{M}(\mathbf{0}, \mathbf{x})\,\mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3) \tag{30}$$

where $\mathbf{O}(\|\mathbf{x}\|^i)$ is a remainder of order i. For $\mathbf{x} = \mathbf{x}_0$, the system (26) can be written:

$$\mathbf{y}(\mathbf{x}_0) \approx \mathbf{y}(\mathbf{0}) + \left(\mathbf{J}(\mathbf{0}) + \frac{1}{2}\mathbf{M}(\mathbf{0}, \mathbf{x}_0)\right)\mathbf{x}_0 = \mathbf{0}. \tag{31}$$

3.3. Iterative Minimization

In general, the system (31) is solved using a sum-of-squared differences minimization. It consists in solving iteratively the following function:

$$f(\mathbf{x}) = \frac{1}{2}\|\mathbf{y}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\mathbf{x} + \frac{1}{2}\mathbf{M}(\mathbf{0}, \mathbf{x})\mathbf{x}\|^2. \tag{32}$$

A necessary condition for a vector $\mathbf{x} = \mathbf{x}_0$ to be a local or a global minimum of the cost function f is that the derivative of f is null at $\mathbf{x} = \mathbf{x}_0$, i.e.:

$$\nabla_{\mathbf{x}} f(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_0} = \mathbf{0}. \tag{33}$$

Standard Newton minimization solves the system (33) iteratively. At each iteration, an incremental $\mathbf{x}_0$ is estimated:

$$\mathbf{x}_0 = -\mathbf{S}^{-1}\mathbf{J}(\mathbf{0})^\top \mathbf{y}(\mathbf{0}) \tag{34}$$

where the (8×8) matrix $\mathbf{S}$ depends on the Hessian matrices $\frac{\partial^2 y_i(\mathbf{x})}{\partial \mathbf{x}^2}$ and is supposed to be invertible:

$$\mathbf{S} = \mathbf{J}(\mathbf{0})^\top \mathbf{J}(\mathbf{0}) + \sum_{i=0}^{q} \left.\frac{\partial^2 y_i(\mathbf{x})}{\partial \mathbf{x}^2}\right|_{\mathbf{x}=\mathbf{0}} y_i(\mathbf{0}). \tag{35}$$

Once $\mathbf{x}_0$ estimated, the homography matrix $\hat{\mathbf{G}}$ is updated as follows:

$$\hat{\mathbf{G}} \longleftarrow \hat{\mathbf{G}}\,\mathbf{G}(\mathbf{x}_0) \tag{36}$$

where the arrow $\longleftarrow$ denotes the update assignment (the left and the right versions of $\hat{\mathbf{G}}$ are respectively the new and the old estimates).

The loop stops if the estimated value of $\mathbf{x}_0$ becomes too small.

The Newton minimization has a quadratic convergence in the neighborhood of $\mathbf{x}_0$. In addition, if the cost function $f(\mathbf{x})$ is convex quadratic, the global minimum can be reached in only one iteration. However, when the cost function $f(\mathbf{x})$ is not convex quadratic, convergence problems may happen if the matrix $\mathbf{S}$ is not definite positive. Furthermore, Newton method needs the computation of the Hessian matrices. For these reasons, many methods have been proposed to approximate the matrix $\mathbf{S}$ with a definite positive matrix $\hat{\mathbf{S}}$. Then, instead of being second-order approximations (as the Newton method does), these methods are first-order approximations (30). Among these methods, there are:

- Gradient descent:

$$\mathbf{S} \approx \hat{\mathbf{S}} = \alpha\,\mathbf{I} \quad \text{where} \quad \alpha > 0 \tag{37}$$

- Gauss–Newton:

$$\mathbf{S} \approx \hat{\mathbf{S}} = \mathbf{J}(\mathbf{0})^\top \mathbf{J}(\mathbf{0}) \tag{38}$$

- Levenberg–Marquardt:

$$\mathbf{S} \approx \hat{\mathbf{S}} = \mathbf{J}(\mathbf{0})^\top \mathbf{J}(\mathbf{0}) + \alpha\,\mathbf{I} \quad \text{where} \quad \alpha > 0. \tag{39}$$

In the literature, many template-based tracking algorithm use such approximations. For example, in Shum and Szeleski (2000), the authors use the Gauss–Newton approximation with a compositional homography update (as described in the equation (36)). In Lucas and Kanade (1981) and Shi and Tomasi (1994) the authors use also the Gauss–Newton approximation with an additional homography update.

There are also algorithms that approximate the current Jacobian $\mathbf{J}(\mathbf{0})$ (which varies from one iteration to another) by a constant Jacobian (Hager and Belhumeur 1998; Baker and Matthews 2001):

$$\mathbf{J}(\mathbf{0}) \approx \hat{\mathbf{J}}. \tag{40}$$

This makes the algorithm faster since the matrix $\mathbf{J}(\mathbf{0})$ and the inverse of the matrix $\hat{\mathbf{S}}$ are computed once for all. However, the price to pay is a smaller convergence region.

The second-order approximation of the cost function is used very little since it needs the computation of the Hessian matrices and convergence problems may happen if the matrix $\mathbf{S}$ is not definite positive.

3.4. The ESM Visual Tracking Algorithm

We present now an efficient algorithm that solves the second order approximation of the system (26), this method will be called "ESM visual tracking algorithm". The proposed method does not need the computation of the Hessian matrices.

3.4.1. The Lie Algebra $\mathfrak{sl}(3)$

The projective transformation matrix $\mathbf{G}(\mathbf{x})$ is in the group $\mathbb{SL}(3)$ which is a Lie group. The Lie algebra associated to this group is $\mathfrak{sl}(3)$. Matrices in this algebra are (3×3) with a null trace. The exponential map is a homeomorphism between a neighborhood of $\mathbf{I} \in \mathbb{SL}(3)$ and a neighborhood of the null matrix $\mathbf{0} \in \mathfrak{sl}(3)$.

Let $\{\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_8\}$ be a basis of the the Lie algebra $\mathfrak{sl}(3)$. A matrix $\mathbf{A}(\mathbf{x}) \in \mathfrak{sl}(3)$ can be written as follows:

$$\mathbf{A}(\mathbf{x}) = \sum_{i=1}^{8} x_i \mathbf{A}_i. \tag{41}$$

A projective transformation $\mathbf{G}(\mathbf{x}) \in \mathbb{SL}(3)$ in the neighborhood of $\mathbf{I}$ can be parameterized as follows:

$$\mathbf{G}(\mathbf{x}) = \exp(\mathbf{A}(\mathbf{x})) = \sum_{i=0}^{\infty} \frac{1}{i!} (\mathbf{A}(\mathbf{x}))^i. \tag{42}$$

We use the following $\mathfrak{sl}(3)$ basis matrices:

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{A}_4 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}_5 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{A}_6 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_7 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \mathbf{A}_8 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

3.4.2. The ESM Iterative Minimization

We use the homography Lie algebra parameterization described above. In the second-order Taylor series approximation of the vector $\mathbf{y}(\mathbf{x})$ about $\mathbf{x} = \mathbf{0}$ (30), the computation of the matrix $\mathbf{M}(\mathbf{0}, \mathbf{x})$ needs the computation of the Hessian matrices of the vector $\mathbf{y}(\mathbf{x})$. However, by using the first-order Taylor series approximation of the vector $\mathbf{J}(\mathbf{x})$ about $\mathbf{x} = \mathbf{0}$:

$$\mathbf{J}(\mathbf{x}) = \mathbf{J}(\mathbf{0}) + \mathbf{M}(\mathbf{0}, \mathbf{x}) + \mathbf{O}(\|\mathbf{x}\|^2) \tag{43}$$

the equation (30) can be written without computing the Hessian matrices of $\mathbf{y}(\mathbf{x})$:

$$\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})) \mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3). \tag{44}$$

It is a second-order approximation of $\mathbf{y}(\mathbf{x})$ about $\mathbf{x} = \mathbf{0}$. For $\mathbf{x} = \mathbf{x}_0$, we have:

$$\mathbf{y}(\mathbf{x}_0) \approx \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}_0)) \mathbf{x}_0. \tag{45}$$

Given the expressions for $\mathbf{J}(\mathbf{0})$ and $\mathbf{J}(\mathbf{x}_0)$ in Appendix A the sum of the Jacobians can be written as follows:

$$\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}_0) = \mathbf{J}_{\mathcal{I}} \mathbf{J}_{\mathbf{w}} \mathbf{J}_{\mathbf{G}} + \mathbf{J}_{\mathcal{I}^*} \mathbf{J}_{\mathbf{w}} \mathbf{J}_{\mathbf{G}_0}. \tag{46}$$

Using the formula (72), the equation (45) can be written as follows:

$$\mathbf{y}(\mathbf{x}_0) \approx \mathbf{y}(\mathbf{0}) + \frac{1}{2} (\mathbf{J}_{\mathcal{I}} + \mathbf{J}_{\mathcal{I}^*}) \ \mathbf{J}_{\mathbf{w}} \ \mathbf{J}_{\mathbf{G}} \ \mathbf{x}_0. \tag{47}$$

Using this approximation, the system (26) can be solved iteratively using the least-square method. Let $\mathbf{J}_{esm}$ be the following matrix:

$$\mathbf{J}_{esm} = \frac{1}{2} (\mathbf{J}_{\mathcal{I}} + \mathbf{J}_{\mathcal{I}^*}) \ \mathbf{J}_{\mathbf{w}} \ \mathbf{J}_{\mathbf{G}}. \tag{48}$$

The cost function to be minimized can be written as follows:

$$f(\mathbf{x}) = \frac{1}{2} \|\mathbf{y}(\mathbf{0}) + \mathbf{J}_{esm} \mathbf{x}\|^2. \tag{49}$$

This cost function has a local or a global minimum in $\mathbf{x} = \mathbf{x}_0$ verifying:

$$\mathbf{x}_0 = \mathbf{J}_{esm}^{+} \mathbf{y}(\mathbf{0}) \tag{50}$$

where $\mathbf{J}_{esm}^{+}$ is the pseudo-inverse of $\mathbf{J}_{esm}$. Iteratively, we estimate $\mathbf{x}_0$, then we update $\hat{\mathbf{G}}$ (36). For each $\hat{\mathbf{G}}$, $\mathbf{y}(\mathbf{0})$ and $\mathbf{J}_{\mathcal{I}}$ are computed. The loop stops when $\mathbf{x}_0$ becomes too small.

Given $\mathbf{G}$, using an approximation of the matrix $\mathbf{K}$, we compute the matrix $\mathbf{H} = \mathbf{K}^{-1}\mathbf{G}\mathbf{K}$. Then, the matrix $\mathbf{H}$ is used for computing the visual servoing control law described in the next section.

4. Homography-based 2D Visual Servoing

In this section, we present a new visual servoing method that does not need any measure of the structure of the observed target. In order to do that, we have to define an isomorphism between the camera pose and the visual information extracted from the reference image and the current image only. Given this isomorphism, we compute a stable control law which also relies on visual information only.

4.1. Isomorphism Between Task Function and Camera Pose

The two frames $\mathcal{F}$ and $\mathcal{F}^*$ coincide, if and only if, the matrix $\mathbf{H}$ is equal to the identity matrix $\mathbf{I}$. Using the homography matrix $\mathbf{H}$, we build a task function $\mathbf{e} \in \mathbb{R}^6$ locally isomorphic to the camera pose (since we have restricted $\theta \neq \pm\pi$). The task function $\mathbf{e}$ is null, if and only if the camera is back to the reference pose.

Theorem 1 *Task function isomorphism.*

Let $\mathbf{R}$ *be the rotation matrix and* $\mathbf{t}$ *be the translation vector between* $\mathcal{F}^*$ *et* $\mathcal{F}$*, where* $\mathbf{R} = \exp\left(\theta\,[\mathbf{u}]_\times\right)$*,* $\theta \in]-\pi, \pi[$ *and let* $\boldsymbol{\mathcal{X}}^* = [X^*\ Y^*\ Z^*]$ *be the coordinates of a certain point* $\mathcal{P} \in \pi$ *in the reference frame* $\mathcal{F}^*$*. We define the task function* $\mathbf{e}$ *as follows:*

$$\mathbf{e} = \begin{bmatrix} \mathbf{e}_v \\ \mathbf{e}_\omega \end{bmatrix} = \begin{bmatrix} (\mathbf{t} + (\mathbf{R} - \mathbf{I})\boldsymbol{\mathcal{X}}^*)/Z^* \\ 2\sin(\theta)\mathbf{u} + [\mathbf{n}^*]_\times \mathbf{t} \end{bmatrix} \tag{51}$$

where $\mathbf{n}^*$ *is the normal vector to the plane* π *expressed in the reference frame* $\mathcal{F}^*$*. The function* $\mathbf{e}$ *is isomorphic to the camera pose, i.e.* $\mathbf{e} = \mathbf{0}$*, if and only if,* $\theta = 0$ *et* $\mathbf{t} = \mathbf{0}$*.*

The proof of the theorem is given in Appendix B. We can demonstrate also that the task function $\mathbf{e}$ can be computed using the two images $\mathcal{I}$ and $\mathcal{I}^*$ only, i.e. without directly measuring the 3D structure of the target ($\mathbf{n}^*$ et Z^*). Given the homography matrix $\mathbf{H}$, we can write:

$$\mathbf{e}_v = (\mathbf{H} - \mathbf{I})\mathbf{m}^* \tag{52}$$

$$[\mathbf{e}_\omega]_\times = \mathbf{H} - \mathbf{H}^\top. \tag{53}$$

See the Appendix B for the proof of these equations. If we have $\mathbf{e}_v = \mathbf{0}$, then the two projections $\boldsymbol{\mathcal{X}}^*$ and $\boldsymbol{\mathcal{X}}$ of the same 3D point $\mathcal{P}$ coincide. And if we have $\mathbf{e}_\omega = \mathbf{0}$, then the homography matrix $\mathbf{H}$ is symmetric.

In this paper, for simplicity, we consider only this isomorphism. However, there exists a group of isomorphisms that can be built using the homography matrix $\mathbf{H}$. For example, we can choose the task function $\mathbf{e}$ as follows:

$$\mathbf{e}_v = \frac{\overline{\mathbf{m}}^\top \mathbf{H} \overline{\mathbf{m}}^*}{\overline{\mathbf{m}}^\top \overline{\mathbf{m}}} \overline{\mathbf{m}} - \overline{\mathbf{m}}^*$$

$$[\mathbf{e}_\omega]_\times = \mathbf{H} - \mathbf{H}^\top$$

where $\overline{\mathbf{m}} = \frac{1}{n}\sum_{i=1}^n \mathbf{m}_i$ and $\overline{\mathbf{m}}^* = \frac{1}{n}\sum_{i=1}^n \mathbf{m}_i^*$ (i.e. the center of gravity of a cloud of points), and where $\mathbf{m}_i^*$ and $\mathbf{m}_i$ are corresponding points. We can demonstrate also that this function is isomorphic to the camera pose.

4.2. The Control Law

The derivative of the task function with respect to time $\dot{\mathbf{e}}$ can be written as follows:

$$\dot{\mathbf{e}} = \mathbf{L} \begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\omega} \end{bmatrix} \tag{54}$$

where $\boldsymbol{\nu}$ is the camera translation velocity, $\boldsymbol{\omega}$ is the camera rotation velocity and $\mathbf{L}$ is the (6×6) interaction matrix. The matrix $\mathbf{L}$ can be written as follows:

$$\mathbf{L} = \begin{bmatrix} 1/Z^* & -[\mathbf{e}_v + \mathbf{m}^*]_\times \\ [\mathbf{n}^*]_\times & -[\mathbf{n}^*]_\times [\mathbf{t}]_\times + 2\mathbf{L}_\omega \end{bmatrix} \tag{55}$$

where the (3×3) matrix $\mathbf{L}_\omega$ can be written as follows:

$$\mathbf{L}_\omega = \mathbf{I} - \frac{\sin(\theta)}{2}[\mathbf{u}]_\times - \sin^2\left(\frac{\theta}{2}\right)(2\mathbf{I} + [\mathbf{u}]_\times^2). \tag{56}$$

The interaction matrix $\mathbf{L}$ does not need to be estimated. It is only useful to analytically prove the following theorem on the stability of the control law:

Theorem 2 *Local stability.*

The control law:

$$\begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\omega} \end{bmatrix} = -\begin{bmatrix} \lambda_v \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \lambda_\omega \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{e}_v \\ \mathbf{e}_\omega \end{bmatrix} \tag{57}$$

where $\lambda_v > 0$ *and* $\lambda_\omega > 0$ *is locally stable.*

See the Appendix B for the proof. This control law only depends on the task function. Consequently, it can be computed using the two images $\mathcal{I}$ and $\mathcal{I}^*$. With such a control law, the task function $\mathbf{e}$ converges exponentially to $\mathbf{0}$. The local stability of the control law is guaranteed for all $\mathbf{n}^*$ and for all $\boldsymbol{\mathcal{X}}^*$. By choosing $\lambda_v > 0$ and $\lambda_\omega > 0$ such that $\lambda_v \neq \lambda_\omega$, one can make $\mathbf{e}_v$ and $\mathbf{e}_\omega$ converge at different speeds.

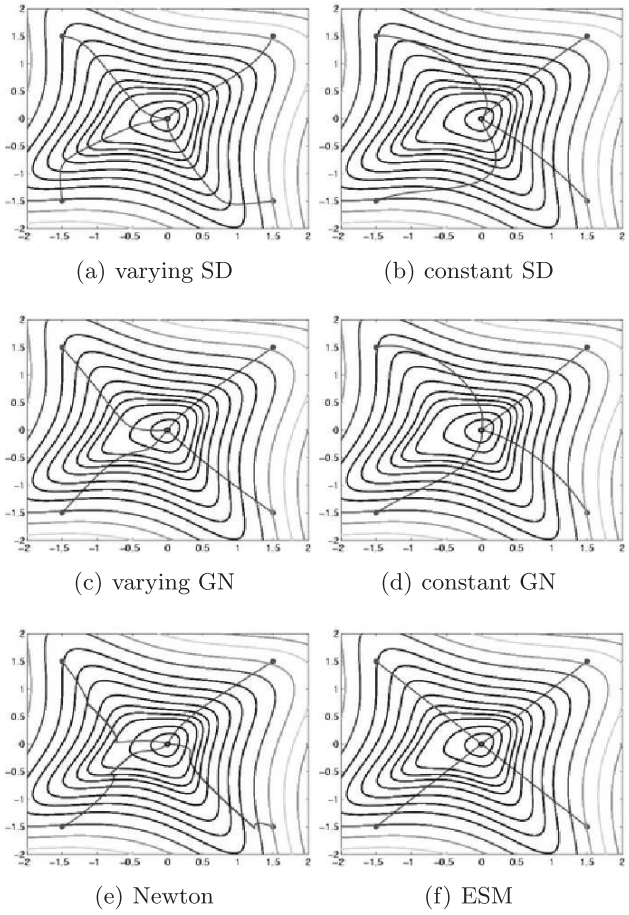
5. Simulation Results

5.1. Advantages of the ESM Algorithm

The main advantage of having a second-order approximation is the high convergence rate. Another advantage is the avoidance of local minima close to the global one (i.e. when the second-order approximation is valid). Here, we show these advantages with the help of two simple examples.

5.1.1. High Convergence Rate

Consider a (4 × 1) vector function $\mathbf{y}(\mathbf{x})$ quadratic in a (2 × 1) parameter vector $\mathbf{x}$. The simulation is repeated 4 times with different starting points: $\mathbf{x}_0 \in \{(\pm 1.5, \pm 1.5)\}$. Suppose we can measure the constant Jacobian $\mathbf{J}(\mathbf{0})$ and the varying Jacobian $\mathbf{J}(\mathbf{x}_0)$. The results for six different minimization methods are given in Figure 2. The contours represent isolines of the SSD (i.e. the cost function has the same value for each point of the contour) while the other lines represent the paths for each starting point. Obviously, the ideal path (i.e. the shortest one) would be a straight line from $\mathbf{x}_0$ to $\mathbf{0}$. Figure 2(a) shows that the varying Steepest Descent method always moves in a direction perpendicular to the isolines. For this reason, it has a slow convergence rate and cannot reach the minimum following a straight line. The paths for the constant Steepest Descent method are even longer (see the path lengths in Figure 2(b)). The constant (Figure 2(d)) and the varying (Figure 2(c)) Gauss–Newton methods perform better than the constant and the varying Steepest Descent methods respectively. In fact, the constant and the varying Gauss–Newton methods use a rough approximation of the Hessian. An ill conditioned and indefinite Hessian matrix causes oscillations of the Newton method in Figure 2(e). Finally, the ESM method gives the best solution since the paths in Figure 2(f) are straight lines. Indeed, when the function $\mathbf{y}(\mathbf{x})$ is exactly quadratic we can correctly estimate the displacement in only one step and thus the correct descent direction regardless of the shape of the isolines.



(a) varying SD (b) constant SD

(c) varying GN (d) constant GN

(e) Newton (f) ESM

Fig. 2. Comparing the behavior of 6 different minimization methods.

5.1.2. Avoiding Local Minima

In the second simulation, we choose a different quadratic function $\mathbf{y}(\mathbf{x})$ such that the corresponding SSD cost function has a local minimum very close to the global minimum. The Newton method and all methods with varying Jacobian fall into the local minimum when the starting point is close to it (see Figures 3(a), 3(c) and 3(e)). In this case, methods with constant Jacobian could diverge (see Figures 3(b) and 3(d)). Indeed, the constant Jacobian approximation is valid only in a neighborhood of the true solution. On the other hand, the ESM method follows the shortest path (see Figure 3(f)). Thus, if $\mathbf{y}(\mathbf{x})$ is locally quadratic the ESM method is able to avoid local minima. Obviously, if the local minimum is far from the true minimum the second-order approximation is not valid any more.

5.2. Comparison with Standard Tracking Methods

We compared the ESM method with the constant Gauss Newton method (CGN) proposed in Baker and Matthews (2001) and with the varying Gauss Newton method (VGN) proposed in Shum and Szeliski (2000). We have used the Matlab software available on the web page of Dr Simon Baker at the Robotics Institute of the Carnegie Mellon University. Thus, the performance of the algorithms were compared with the same experimental setup. In order to have a ground truth, the ESM algorithm was tested by warping the image shown in Figure 4(a). The (124 × 124) template illustrated in Figure 4(b) was selected in the center of the image. The computational complexity of the ESM algorithm is equivalent to the VGN method, which is higher than the CGN method. In order to have the same execution time per iteration, we can use a smaller subset (25 %) of the template for computing the Jacobians and the estimated displacement. The template was warped 1000 times using different random homographies. Similarly to Baker and Matthews (2001), the homography was computed by adding a Gaussian noise to the coordinates of the four corners of the template. The standard deviation σ of the Gaussian noise was increased from 1 to 12. Figure 4(c) plots the frequencies of convergence (% over 1000 tests). As

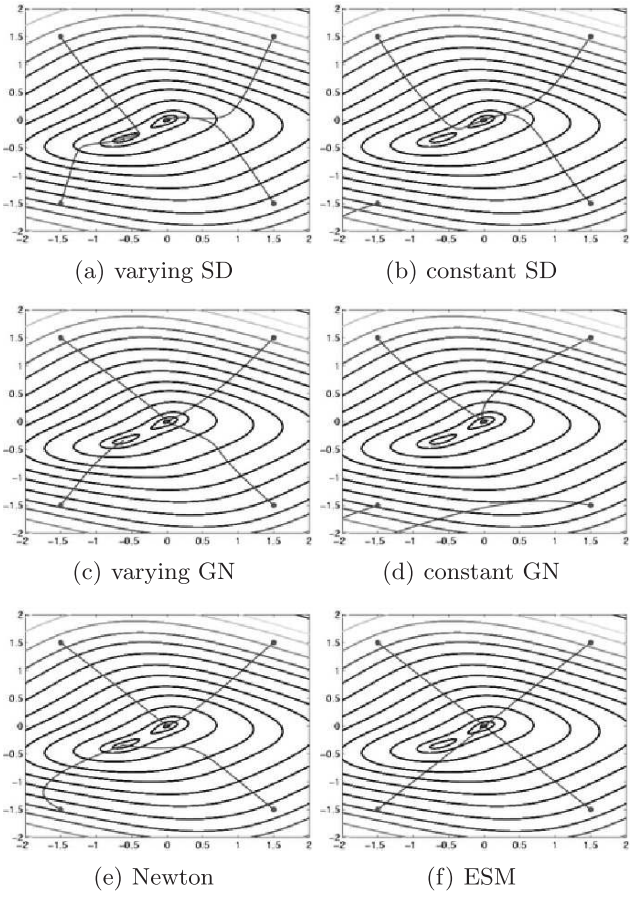


Fig. 3. Comparing the behavior of 6 different minimization methods.

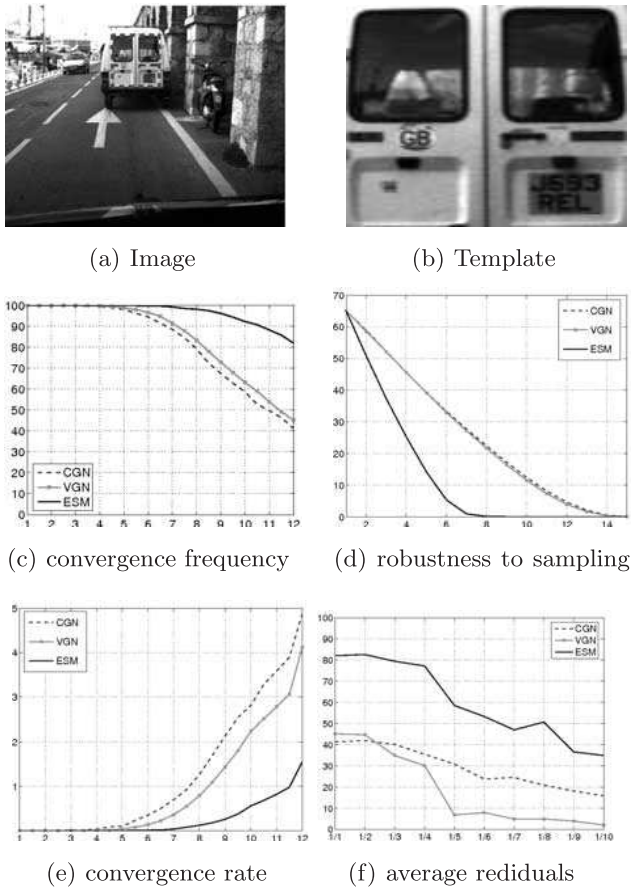


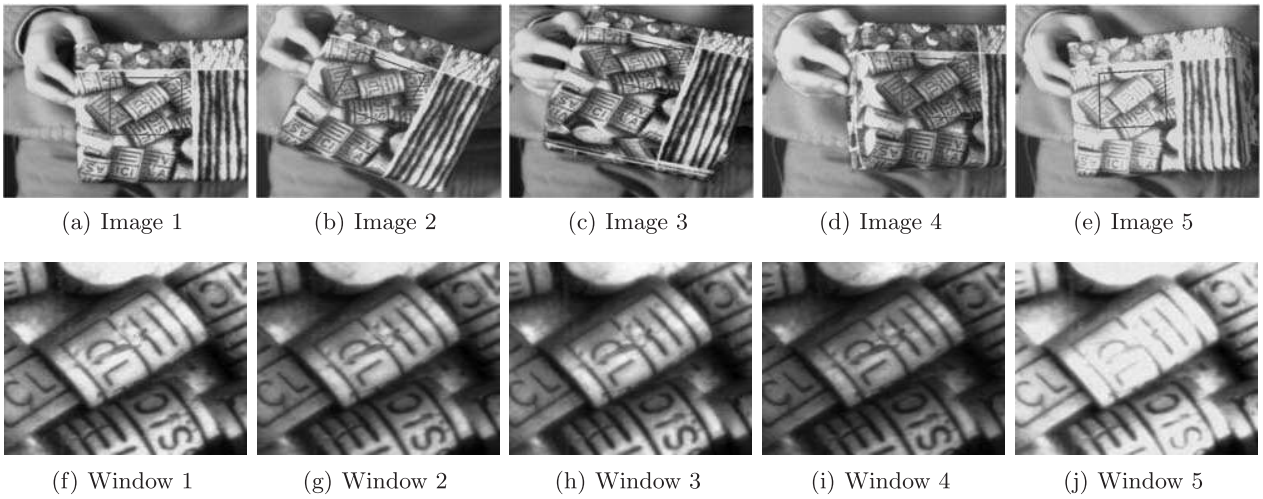
Fig. 4. Comparison between ESM and standard methods.

σ increases, the frequency of convergence of the CGN and the VGN methods decay quicker than the frequency of convergence of the ESM method. At the final $\sigma = 12$, the frequency of convergence of the CGN and the VGN methods are only 40% while the frequency of convergence of the ESM method is 80%. Figure 4(e) shows the average convergence rate (over the converged tests) of the algorithms for $\sigma = 12$. The initial value of the SSD is the same for the three algorithms but the speed of convergence of the ESM method is much higher. This means that we can perform real-time tracking at higher rates. Since our objective is to track objects in real time, it is very important to measure the residuals after each minimization. Indeed, since the number of iterations is fixed by the frame rate, the error will cumulate. Figure 4(f) plots the average residual over all the tests for which the algorithms did not diverge (we consider that the algorithm diverges when the final SSD is bigger than the initial SSD). Obviously the SSD increases with the amplitude of the initial displacement. However, the ESM method performs much better than the CGN method and the VGN method. Finally, we tested the robustness of the algorithms to sampling. Figure 4(d) plots the frequency of convergence for $\sigma = 12$ against the sampling rate r between the size of the subset used in the algorithm and the size of the template, e.g. for 1/1 we use all the template while for 1/10 we use 1 % of the template (1 pixel used every 10 pixels of the image per line and per column). The ESM algorithm is more robust to sampling. For $r = 1/10$, the frequency of convergence of the ESM method is almost the same as the two other methods without sampling. Thus, we can obtain the same frequency of convergence with a faster algorithm.

6. Experimental Results

6.1. Visual Tracking

The second-order tracking was tested on sequences with moving planar objects. Five images were extracted from each sequence and they are shown in the first row of Figure 5 and



(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4 (e) Image 5

(f) Window 1 (g) Window 2 (h) Window 3 (i) Window 4 (j) Window 5

Fig. 5. Tracking a template on a planar object.

(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4 (e) Image 5

(f) Window 1 (g) Window 2 (h) Window 3 (i) Window 4 (j) Window 5

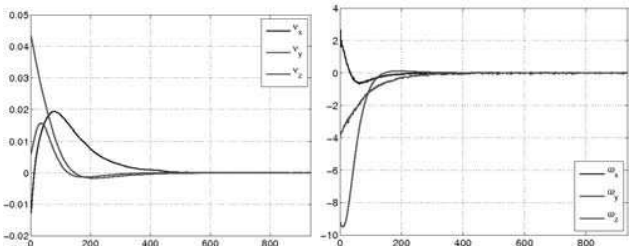
Fig. 6. Tracking a template on the back of a car.

Figure 6. In the first experiment, the template to track was a (150 × 150) window shown in Figure 5(f). The red windows in the first row of Figure 5 are warped back and shown in the second row of Figure 5. Despite illumination changes and image noise, the warped windows are very close to the reference template proving that the tracking is accurately performed. During the sequence, a generic projective motion and several light variations were observed. For example, Figures 5(b) and 5(c) show translation and rotation around the $\vec{z}$ and $\vec{x}$ axis respectively, while Figure 5(d) and 5(e) show a rotation around the $\vec{y}$ and varying illumination (the image becomes darker, the image becomes lighter). In the second experiment, we tracked a (43 × 43) template on the back of a car with a camera mounted on another car (see Figure 6(a) to (e)). Again, the tracking is accurately performed in spite of the template changes due to people movement that we can see through the window of the car (see Figure 6(f) to (j)).

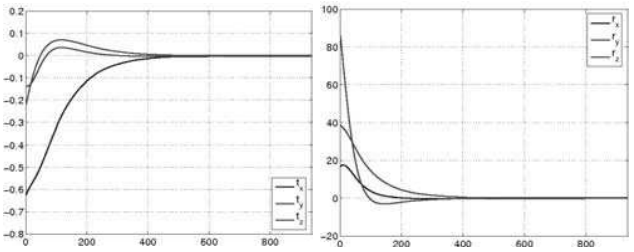
6.2. Visual Tracking and Servoing

We tested the proposed visual servoing on the 6 d.o.f. robot of the LAGADIC research team at INRIA Rennes. The robot is accurately calibrated and it provides a ground truth for measuring the accuracy of the positioning task. A calibrated camera was mounted on the end-effector of the robot and a reference

(a) Initial image (b) Final image

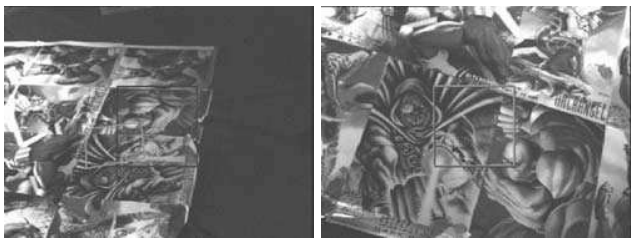


(c) Translation velocity (d) Rotation velocity

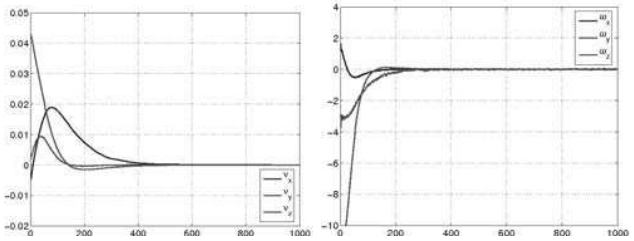


(e) Translation error function (f) Rotation error

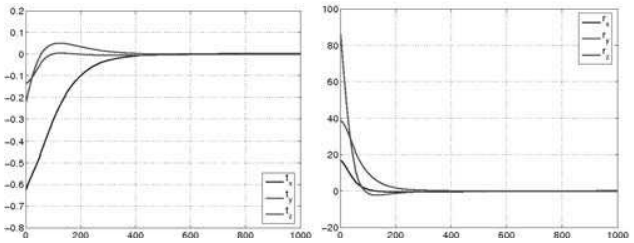
Fig. 7. Experiment 1: Camera positioning with respect to a planar object without approximating the normal vector to the object plane.



(a) Initial image (b) Final image



(c) Translation velocity (d) Rotation velocity



(e) Translation error function (f) Rotation error

Fig. 8. Experiment 2: Camera positioning with an uncalibrated camera without approximating the normal vector to the object plane.

image was captured at the reference pose. The positioning a planar target was used for the positioning task. We started from another pose (the initial pose) which allowed us to see the object from a different angle. The robot was controlled using the control law (57) with $\lambda_v = \lambda_\omega = 0.1$ in order to return to the reference pose. At the initial pose (the translation displacement is 0.68 m and the rotation displacement is 96 degrees), we can see the projective transformation of the area of interest (the rectangle in the center in Figures 7(a) and 7(b) corresponds to the desired position). We used the ESM[1] visual tracking algorithm (Benhimane and Malis 2004) to track the area of interest and at the same time to estimate the homography matrix $\mathbf{H}$. Given the matrix $\mathbf{H}$, the control law was computed. As the control point ($\mathbf{m}^*$ in the equation (52)) we used the center of gravity of the area. At the convergence, the robot is back to its reference pose and the visual information coincides with the visual information of the reference pose (see figure 7(b)). The control law is stable: the translation figures 7(c) and the rotation 7(d) velocities converge to zero. As shown in Figures 7(e) and 7(f), the camera displacement converge to zero very accurately (less than 1 mm error for the translation and less that 0.1 degree for the rotation).

A second experiment was performed under similar conditions (the same initial camera displacement, an unknown normal vector to the plane, an unknown camera/object distance...). In contrast to the previous experiment, the positioning task was performed with respect to a different target (see Figure 8(a)). We also used a very poor estimation of the camera parameters: $\widehat{f} = 800$, $\widehat{r} = 0.5$, $\widehat{u}_0 = 100$, $\widehat{v}_0 = 200$ (the calibrated parameters were $f = 592$, $r = 0.96$, $u_0 = 198$, $v_0 = 140$). Figures 8(c) and 8(d) show that the control law is robust to camera calibration errors: the translation and rotation velocities converge to zero. At the convergence, the visual information coincides with the visual information of the reference

1. The ESM visual tracking software can be downloaded from the following web-page: http://www-sop.inria.fr/icare/personnel/malis/software/ESM.html

image (see Figure 8(b)). Again, Figures 8(e) and 8(f) show that the camera displacement converges to zero (as in the previous experiment an error of approximately 1 mm error for the translation 0.1 degrees for the rotation).

7. Conclusions

In this paper, we have described two contributions to vision-based robot control. First, we have proposed a real-time algorithm for tracking images of planar targets. We performed an efficient second-order approximation of the image error using only first-order derivatives (the ESM algorithm). This avoids the computation of the Hessian of the cost function. At the same time, the second-order approximation allows the tracking algorithm to achieve a high convergence rate. This is very important if we want to track objects in real time. Secondly, this is the first time that a homography-based 2D approach to visual servoing that do not need any measure of the 3D structure of the observed target has been proposed. We have designed a simple and stable control law that directly uses the output of the ESM visual tracking (i.e. the homography). We think that this approach can open new research directions in the field of vision-based robot control. Indeed, as far as we know, none of the existing methods are able to position a robot with respect to an object without measuring, on-line or off-line, some information on its 3D structure.

Many improvements in the proposed methods should be studied. The ESM algorithm could be extended in order to take into account illumination changes or could be transformed into a robust algorithm in order to take into account partial occlusions. The control law could be improved using a trajectory planning in order to have a larger stability region and to take into account visibility constraints.

Appendix A

A.1. Jacobians Computation

The objective of this paragraph is to compute the Jacobian matrix $\mathbf{J}(\mathbf{x})$ corresponding to the derivative of $\mathbf{y}(\mathbf{x})$ at $\mathbf{x} = \mathbf{0}$ and at $\mathbf{x} = \mathbf{x}_0$ ($\mathbf{x}_0$ verifies the equation (27)).

A.1.1. The Current Jacobian

The ith line of the matrix $\mathbf{J}(\mathbf{0})$, called the current Jacobian, can be written as follows:

$$\nabla_{\mathbf{x}} y_i(\mathbf{x})|_{\mathbf{x}=\mathbf{0}} = \nabla_{\mathbf{x}} \mathcal{I}\left(\mathbf{w}(\hat{\mathbf{G}}\mathbf{G}(\mathbf{x}))(\mathbf{p}_i^*)\right)\Big|_{\mathbf{x}=\mathbf{0}}. \quad (58)$$

Thanks to the property (19), we have:

$$\nabla_{\mathbf{x}} y_i(\mathbf{x})|_{\mathbf{x}=\mathbf{0}} = \nabla_{\mathbf{x}} \mathcal{I}\left(\mathbf{w}(\hat{\mathbf{G}})(\mathbf{w}(\mathbf{G}(\mathbf{x}))(\mathbf{p}_i^*))\right)\Big|_{\mathbf{x}=\mathbf{0}}. \quad (59)$$

The ith line of the Jacobian $\mathbf{J}(\mathbf{0})$ can be written as the product of three Jacobians:

$$\nabla_{\mathbf{x}} y_i(\mathbf{x})|_{\mathbf{x}=\mathbf{0}} = \mathbf{J}_{\mathcal{I}}\, \mathbf{J}_{\mathbf{w}}\, \mathbf{J}_{\mathbf{G}}. \quad (60)$$

1. $\mathbf{J}_{\mathcal{I}}$ is a (1×3) matrix corresponding to the spatial derivative of the current image warped using the projective transformation $\mathbf{w}(\hat{\mathbf{G}})$:

$$\mathbf{J}_{\mathcal{I}} = \nabla_{\mathbf{z}} \mathcal{I}\left(\mathbf{w}(\hat{\mathbf{G}})(\mathbf{z})\right)\Big|_{\mathbf{z}=\mathbf{p}_i^*}. \quad (61)$$

   Since $\mathbf{p}_i^*$ is in $\mathbb{P}^2$ it is a (3×1) vector with the third entry equal to 1. The third entry of $\mathbf{J}_{\mathcal{I}}$ is equal to zero.

2. The Jacobian $\mathbf{J}_{\mathbf{w}}$ is a (3×9) matrix:

$$\mathbf{J}_{\mathbf{w}} = \nabla_{\mathbf{Z}} \mathbf{w}(\mathbf{Z})(\mathbf{p}_i^*)\big|_{\mathbf{Z}=\mathbf{G}(\mathbf{0})=\mathbf{I}}. \quad (62)$$

   For $\mathbf{p}_i^* = \begin{bmatrix} u_i^* & v_i^* & 1 \end{bmatrix}^\top$, this Jacobian can be written as follows:

$$\mathbf{J}_{\mathbf{w}} = \begin{bmatrix} \mathbf{p}_i^{*\top} & \mathbf{0} & -u_i^* \mathbf{p}_i^{*\top} \\ \mathbf{0} & \mathbf{p}_i^{*\top} & -v_i^* \mathbf{p}_i^{*\top} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (63)$$

3. The Jacobian $\mathbf{J}_{\mathbf{G}}$ is a (9×8) matrix that can be written as follows:

$$\mathbf{J}_{\mathbf{G}} = \nabla_{\mathbf{x}} \mathbf{G}(\mathbf{x})|_{\mathbf{x}=\mathbf{0}} \quad (64)$$

   Using (41) and (42), this Jacobian can be written as:

$$\mathbf{J}_{\mathbf{G}} = \begin{bmatrix} [\mathbf{A}_1]_v & [\mathbf{A}_2]_v & \dots & [\mathbf{A}_8]_v \end{bmatrix} \quad (65)$$

   where $[\mathbf{A}_i]_v$ is the matrix $\mathbf{A}_i$ reshaped as a vector (the entries are picked line per line).

The two Jacobians $\mathbf{J}_{\mathbf{w}}$ and $\mathbf{J}_{\mathbf{G}}$ are constants. Thus, they can be computed once and for all. The Jacobian $\mathbf{J}_{\mathcal{I}}$ has to be computed at each iteration since it depends on the updated value $\mathbf{w}(\hat{\mathbf{G}})$.

A.1.2. The Reference Jacobian

Using equations (19), (20) and (23), $y_i(\mathbf{x})$ can be written as follows:

$$y_i(\mathbf{x}) = \mathcal{I}^*\left(\mathbf{w}(\overline{\mathbf{G}}^{-1}\hat{\mathbf{G}}\mathbf{G}(\mathbf{x}))(\mathbf{p}_i^*)\right) - \mathcal{I}^*(\mathbf{p}_i^*). \quad (66)$$

Using equation (27), the ith line of the Jacobian $\mathbf{J}(\mathbf{x}_0)$, called the reference Jacobian, can be written as follows:

$$\nabla_{\mathbf{x}} y_i(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_0} = \nabla_{\mathbf{x}} \mathcal{I}^*\left(\mathbf{w}(\mathbf{G}(\mathbf{x}_0)^{-1}\mathbf{G}(\mathbf{x}))(\mathbf{p}_i^*)\right)\big|_{\mathbf{x}=\mathbf{x}_0}. \quad (67)$$

The ith line of the Jacobian can be written as the product of three Jacobians:

$$\nabla_{\mathbf{x}} y_i(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_0} = \mathbf{J}_{\mathcal{I}^*}\, \mathbf{J}_{\mathbf{w}_0}\, \mathbf{J}_{\mathbf{G}_0}. \quad (68)$$

1. $\mathbf{J}_{\mathcal{I}^*}$ is a (1×3) matrix corresponding to the spatial derivative of the reference image:

$$\mathbf{J}_{\mathcal{I}^*} = \nabla_{\mathbf{z}} \mathcal{I}^*(\mathbf{z})\big|_{\mathbf{z}=\mathbf{p}_i^*}. \tag{69}$$

   As for $\mathbf{J}_{\mathcal{I}}$, the third entry of $\mathbf{J}_{\mathcal{I}^*}$ is equal to zero.

2. The Jacobian $\mathbf{J}_{\mathbf{w}_0}$ is a (3×9) matrix:

$$\mathbf{J}_{\mathbf{w}_0} = \nabla_{\mathbf{Z}} \mathbf{w}(\mathbf{Z})(\mathbf{p}_i^*)\big|_{\mathbf{Z}=\mathbf{G}(\mathbf{x}_0)^{-1}\mathbf{G}(\mathbf{x}_0)=\mathbf{I}} = \mathbf{J}_{\mathbf{w}}. \tag{70}$$

3. The third Jacobian $\mathbf{J}_{\mathbf{G}_0}$ is a (9×8) matrix:

$$\mathbf{J}_{\mathbf{G}_0} = \nabla_{\mathbf{x}} \mathbf{G}(\mathbf{x}_0)^{-1}\mathbf{G}(\mathbf{x})\big|_{\mathbf{x}=\mathbf{x}_0}. \tag{71}$$

The Jacobians $\mathbf{J}_{\mathcal{I}^*}$ and $\mathbf{J}_{\mathbf{w}_0}$ are constants and can be computed once and for all. The Jacobian $\mathbf{J}_{\mathbf{G}_0}$ is complicated and generally depends $\mathbf{x}_0$. Using the Lie algebra and the exponential map properties, we can show that:

$$\mathbf{J}_{\mathbf{G}_0}\, \mathbf{x}_0 = \mathbf{J}_{\mathbf{G}}\, \mathbf{x}_0. \tag{72}$$

Appendix B

B.1. The Task Function is a Function of Image Measures Only

Using the equation (51), the vector $\mathbf{e}_v$ can be written as follows:

$$\mathbf{e}_v = (\mathbf{t} + (\mathbf{R} - \mathbf{I})\boldsymbol{\mathcal{X}}^*)/Z^* = (\mathbf{R}\boldsymbol{\mathcal{X}}^* + \mathbf{t} - \boldsymbol{\mathcal{X}}^*)/Z^*.$$

Using the equation (4), $\mathbf{e}_v$ becomes:

$$\mathbf{e}_v = (\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}^*)/Z^*.$$

Plugging equations (1) and (7) gives:

$$\mathbf{e}_v = \frac{Z}{Z^*}\mathbf{m} - \mathbf{m}^*.$$

Thanks to (11), $\mathbf{e}_v$ can be written using $\mathbf{H}$ and $\mathbf{m}^*$ only:

$$\mathbf{e}_v = \mathbf{H}\mathbf{m}^* - \mathbf{m}^* = (\mathbf{H} - \mathbf{I})\mathbf{m}^*.$$

Thanks to equation (12), we have:

$$\mathbf{H} - \mathbf{H}^\top = \mathbf{R} + \mathbf{t}\mathbf{n}^{*\top} - \mathbf{R}^\top - \mathbf{n}^*\mathbf{t}^\top.$$

Using the Rodriguez formula for the rotation matrix $\mathbf{R}$:

$$\mathbf{R} = \mathbf{I} + \sin(\theta)\,[\mathbf{u}]_\times + 2\cos^2\left(\frac{\theta}{2}\right)[\mathbf{u}]_\times^2$$

we can write:

$$\mathbf{R} - \mathbf{R}^\top = 2\sin(\theta)\,[\mathbf{u}]_\times.$$

Given the following property:

$$\mathbf{t}\mathbf{n}^{*\top} - \mathbf{n}^*\mathbf{t}^\top = \left[\left[\mathbf{n}^*\right]_\times \mathbf{t}\right]_\times.$$

The antisymmetric part of the matrix $\mathbf{H}$ can be written as:

$$\mathbf{H} - \mathbf{H}^\top = \left[2\sin(\theta)\mathbf{u} + \left[\mathbf{n}^*\right]_\times \mathbf{t}\right]_\times.$$

Consequently, given equation (51), we have:

$$\mathbf{H} - \mathbf{H}^\top = [\mathbf{e}_\omega]_\times.$$

B.2. The Task Function is Isomorphic to the Camera Pose

In order to simplify the proof of Theorem (1), we prove three simpler propositions.

Proposition 1 The matrix $\mathbf{H}\mathbf{H}^\top$ has one eigenvalue equal to 1. The eigenvector corresponding to the eigenvalue is $\mathbf{v} = \left[\mathbf{R}\mathbf{n}^*\right]_\times \mathbf{t}$.

Proof of proposition 1 *Using equation (12), we have:*

$$\mathbf{H}\mathbf{H}^\top = (\mathbf{R} + \mathbf{t}\mathbf{n}^{*\top})(\mathbf{R}^\top + \mathbf{n}^*\mathbf{t}^\top).$$

Since we have $\mathbf{R} \in \mathbb{SO}(3)$ *then* $\mathbf{R}\mathbf{R}^\top = \mathbf{I}$. *Thus, we have:*

$$\mathbf{H}\mathbf{H}^\top = \mathbf{I} + \mathbf{t}(\mathbf{R}\mathbf{n}^*)^\top + (\mathbf{R}\mathbf{n}^* + \|\mathbf{n}^*\|^2\mathbf{t})\mathbf{t}^\top.$$

The matrix $\mathbf{H}\mathbf{H}^\top$ *is the sum of* $\mathbf{I}$ *and a rank 2 matrix. Thus, one eigenvalue of* $\mathbf{H}\mathbf{H}^\top$ *is equal to 1. Setting* $\mathbf{v} = \left[\mathbf{R}\mathbf{n}^*\right]_\times \mathbf{t}$, *we have:*

$$(\mathbf{R}\mathbf{n}^*)^\top\mathbf{v} = 0 \quad \textit{and} \quad \mathbf{t}^\top\mathbf{v} = 0$$

showing that $\mathbf{v}$ *is an eigenvector of* $\mathbf{H}\mathbf{H}^\top$*:*

$$\mathbf{H}\mathbf{H}^\top\mathbf{v} = \mathbf{v}.$$

Proposition 2 If $\mathbf{H} = \mathbf{H}^\top$ and $\sin(\theta) \neq 0$, then $\mathbf{n}^{*\top}\,\mathbf{u} = 0$, $\mathbf{t}^\top\,\mathbf{u} = 0$ and $\mathbf{n}^{*\top}\,\mathbf{v} = 0$ (where $\mathbf{v} = \left[\mathbf{R}\mathbf{n}^*\right]_\times \mathbf{t}$).

Proof of proposition 2 *If we have* $\mathbf{H} = \mathbf{H}^\top$, *then we have:*

$$2\sin(\theta)\mathbf{u} + \left[\mathbf{n}^*\right]_\times \mathbf{t} = \mathbf{0}. \tag{73}$$

By multiplying each side of the equation (73) by $\mathbf{n}^{*\top}$, *we obtain:*

$$2\sin(\theta)\mathbf{n}^{*\top}\mathbf{u} = \mathbf{0}.$$

Since we have supposed that $\sin(\theta) \neq 0$, *we have:*

$$\mathbf{n}^{*\top}\mathbf{u} = 0.$$

Similarly, by multiplying each side of the equation (73) by $\mathbf{t}^\top$, *we obtain:*

$$\mathbf{t}^\top\mathbf{u} = 0.$$

Finally, using the Rodriguez formula for the rotation matrix, we have:

$$\begin{aligned}\mathbf{R}\mathbf{n}^* &= \left(\mathbf{I} + \sin(\theta)\,[\mathbf{u}]_\times + 2\cos^2\left(\frac{\theta}{2}\right)[\mathbf{u}]_\times^2\right)\mathbf{n}^* \\ &= \mathbf{n}^* + \sin(\theta)\,[\mathbf{u}]_\times\,\mathbf{n}^* + 2\cos^2\left(\frac{\theta}{2}\right)[\mathbf{u}]_\times^2\,\mathbf{n}^* \\ &= \mathbf{n}^* + \sin(\theta)\,[\mathbf{u}]_\times\,\mathbf{n}^* + 2\cos^2\left(\frac{\theta}{2}\right)\left(\mathbf{u}\mathbf{u}^\top - \mathbf{I}\right)\mathbf{n}^*.\end{aligned}$$

If we have $\mathbf{n}^{*\top}\mathbf{u} = 0$*, then we have:*

$$\mathbf{R}\mathbf{n}^* = \mathbf{n}^* + \sin(\theta)\,[\mathbf{u}]_\times\,\mathbf{n}^* - 2\cos^2\left(\frac{\theta}{2}\right)\mathbf{n}^*. \tag{74}$$

The antisymmetric matrix associated to the vector $\mathbf{R}\mathbf{n}^*$ *is:*

$$\left[\mathbf{R}\mathbf{n}^*\right]_\times = \left[\mathbf{n}^*\right]_\times + \sin(\theta)\left[[\mathbf{u}]_\times\,\mathbf{n}^*\right]_\times - 2\cos^2\left(\frac{\theta}{2}\right)\left[\mathbf{n}^*\right]_\times]$$

and since $\left[[\mathbf{u}]_\times\,\mathbf{n}^*\right]_\times = \mathbf{n}^*\mathbf{u}^\top - \mathbf{u}\mathbf{n}^{*\top}$*, we can write:*

$$\begin{aligned}\left[\mathbf{R}\mathbf{n}^*\right]_\times &= \left[\mathbf{n}^*\right]_\times + \sin(\theta)\left(\mathbf{n}^*\mathbf{u}^\top - \mathbf{u}\mathbf{n}^{*\top}\right) \\ &- 2\cos^2\left(\frac{\theta}{2}\right)\left[\mathbf{n}^*\right]_\times].\end{aligned}$$

By multiplying both sides of the equation by $\mathbf{n}^{*\top}$*, we obtain:*

$$\mathbf{n}^{*\top}\left[\mathbf{R}\mathbf{n}^*\right]_\times = \|\mathbf{n}^*\|^2\sin(\theta)\mathbf{u}^\top. \tag{75}$$

By multiplying both sides of the equation by $\mathbf{t}$*, we obtain:*

$$\mathbf{n}^{*\top}\left[\mathbf{R}\mathbf{n}^*\right]_\times\mathbf{t} = \left\|\mathbf{n}^*\right\|^2\sin(\theta)\mathbf{u}^\top\mathbf{t}.$$

Since $\mathbf{u}^\top\mathbf{t} = 0$*, then we prove that:*

$$\mathbf{n}^{*\top}\mathbf{v} = 0.$$

Proposition 3 If $\mathbf{H} = \mathbf{H}^\top$, $\mathbf{v} = \left[\mathbf{R}\mathbf{n}^*\right]_\times\mathbf{t} = \mathbf{0}$ and $\sin(\theta) \neq 0$ then $\det(\mathbf{H}) = -1$.

Proof of proposition 3 *If* $\mathbf{v} = \left[\mathbf{R}\mathbf{n}^*\right]_\times\mathbf{t} = \mathbf{0}$ *then it exists* $\alpha > 0$ *such that:*

$$\mathbf{t} = \alpha\mathbf{R}\mathbf{n}^*.$$

From equation (75), we obtain:

$$\left[\mathbf{n}^*\right]_\times\mathbf{R}\mathbf{n}^* = \left(\mathbf{n}^{*\top}\left[\mathbf{R}\mathbf{n}^*\right]_\times\right)^\top = \left\|\mathbf{n}^*\right\|^2\sin(\theta)\mathbf{u}. \tag{76}$$

Then, from equation (73) and equation (76), we obtain:

$$2\sin(\theta)\mathbf{u} = -\left[\mathbf{n}^*\right]_\times\mathbf{t} = -\alpha\left[\mathbf{n}^*\right]_\times\mathbf{R}\mathbf{n}^* = -\alpha\left\|\mathbf{n}^*\right\|^2\sin(\theta)\mathbf{u}.$$

By multiplying both sides of this equation by $\mathbf{u}^\top$*, we obtain:*

$$2\sin(\theta) = -\alpha\sin(\theta)\|\mathbf{n}^*\|^2.$$

Since we supposed $\sin(\theta) \neq 0$*, then we can write:*

$$\alpha = -\frac{2}{\|\mathbf{n}^*\|^2}$$

and finally the determinant of the matrix $\mathbf{H}$ *verifies:*

$$\det(\mathbf{H}) = 1 + \mathbf{n}^{*\top}\mathbf{R}^\top\mathbf{t} = 1 + \alpha\|\mathbf{n}^*\|^2 = -1. \tag{77}$$

Having a matrix $\mathbf{H}$ *with negative determinant means that the current frame* $\mathcal{F}$ *is on the opposite side of the target plane. This is impossible since it means that we cannot see the target in the image any more. This is the reason why we can always suppose* $\det(\mathbf{H}) > 0$*.*

Proof of theorem 1 *It is evident that if* $\theta = 0$ *and* $\mathbf{t} = \mathbf{0}$ *then* $\mathbf{e} = \mathbf{0}$*. We must prove now that if* $\mathbf{e} = \mathbf{0}$*, then* $\theta = 0$ *and* $\mathbf{t} = \mathbf{0}$*. Let us suppose that* $\mathbf{e} = \mathbf{0}$*. It is evident that if* $\theta = 0$ *then* $\mathbf{t} = \mathbf{0}$ *and if* $\mathbf{t} = \mathbf{0}$ *then* $\theta = 0$*. Now, let us suppose that* $\mathbf{e} = \mathbf{0}$ *and* $\mathbf{t} \neq \mathbf{0}$ *and* $\theta \neq 0$*. If* $\mathbf{e}_v = \mathbf{0}$ *then* $\mathbf{H}\mathbf{m}^* = \mathbf{m}^*$*. Thus,* $\mathbf{H}$ *has an eigenvalue equal to 1 and the vector* $\mathbf{m}^*$ *is the corresponding eigenvector. The vector* $\mathbf{m}^*$ *is also the eigenvector corresponding to the eigenvalue 1 of the matrix* $\mathbf{H}^2$*. Since* $\mathbf{e}_\omega = \mathbf{0}$ *then* $\mathbf{H} = \mathbf{H}^\top$ *and* $\mathbf{H}^2 = \mathbf{H}\mathbf{H}^\top$*. Given Proposition 2,* $\mathbf{m}^*$ *is then collinear to the vector* $\mathbf{v} = \left[\mathbf{R}\mathbf{n}^*\right]_\times\mathbf{t}$*. Since* $\det(\mathbf{H}) > 0$*, this vector is different from zeros (see Proposition 3). On the other hand, Proposition 2 shows that in this case* $\mathbf{n}^{*\top}\mathbf{m}^* = Z^* = 0$*. This is impossible since by definition* $Z^* > 0$*. Thus, it is impossible that* $\mathbf{e} = \mathbf{0}$ *and* $\mathbf{t} \neq \mathbf{0}$*,* $\theta \neq 0$*.*

B.3. The Interaction Matrix

Using equation (51), we have:

$$\begin{aligned}\dot{\mathbf{e}}_v &= (\dot{\mathbf{t}} + \dot{\mathbf{R}}\boldsymbol{\mathcal{X}}^*)/Z^* \\ &= (\boldsymbol{\nu} + [\boldsymbol{\omega}]_\times\,\mathbf{t} + [\boldsymbol{\omega}]_\times\,\mathbf{R}\boldsymbol{\mathcal{X}}^*)/Z^* \\ &= \boldsymbol{\nu}/Z^* + [\boldsymbol{\omega}]_\times\,(\mathbf{t} + \mathbf{R}\boldsymbol{\mathcal{X}}^*)/Z^* \\ &= \boldsymbol{\nu}/Z^* + [\boldsymbol{\omega}]_\times\left(\mathbf{e}_v + \mathbf{m}^*\right) \\ &= \boldsymbol{\nu}/Z^* - \left[\mathbf{e}_v + \mathbf{m}^*\right]_\times\boldsymbol{\omega}\end{aligned}$$

and:

$$\begin{aligned}\dot{\mathbf{e}}_\omega &= 2\frac{d\sin(\theta)\mathbf{u}}{dt} + [\mathbf{n}]_\times\,\dot{\mathbf{t}} \\ &= 2\mathbf{L}_\omega + [\mathbf{n}]_\times\,(\boldsymbol{\nu} + [\boldsymbol{\omega}]_\times\,\mathbf{t}) \\ &= [\mathbf{n}]_\times\,\boldsymbol{\nu} + (2\mathbf{L}_\omega - [\mathbf{n}]_\times\,[\mathbf{t}]_\times)\boldsymbol{\omega}\end{aligned}$$

Finally, we obtain equation (54) and the interaction matrix in equation (55).

B.4. Proof of the Local Stability of the Control Law

Proof of theorem 2 *After linearizing equation (54) about* $\mathbf{e} = \mathbf{0}$*, we obtain the following linear system:*

$$\dot{\mathbf{e}} = -\begin{bmatrix} \lambda_t/Z^* & -\lambda_r\,[\mathbf{m}^*]_\times \\ \lambda_t\,[\mathbf{n}^*]_\times & 2\lambda_r\mathbf{I} \end{bmatrix}\mathbf{e} = -\mathbf{L}_0\mathbf{e}.$$

The eigenvectors of the constant matrix $\mathbf{L}_0$ *are:* 2λ*,* $4Z^*$*,* $2Z^*+\lambda+\sqrt{\lambda^2+4Z^{*2}}$ *(twice),* $2Z^*+\lambda-\sqrt{\lambda^2+4Z^{*2}}$ *(twice), where* $\lambda = \lambda_v/\lambda_\omega$*. Since* $\lambda > 0$ *and* $Z^* > 0$*, the eigenvalues of matrix* $\mathbf{L}_0$ *are always positives. Consequently, the control law defined in equation (57) is always locally stable for any* $\mathbf{n}^*$ *and any* $\mathbf{m}^*$*.*

References

Baker, S. and Matthews, I. (2001). Equivalence and efficiency of image alignment algorithms. *IEEE Conference on Computer Vision and Pattern Recognition*, **1**: 1090–1097.

Basri, R., Rivlin, E., and Shimshoni, I. (1998). Visual homing: Surfing on the epipoles. *IEEE International Conference on Computer Vision*, pp. 863–869.

Benhimane, S. and Malis, E. (2004). Real-time image-based tracking of planes using efficient second-order minimization. *IEEE/RSJ International Conference on Intelligent Robots Systems*, pp. 943–948.

Chaumette, F. (1998). Potential problems of stability and convergence in image-based and position-based visual servoing. In *The Confluence of Vision and Control* (eds D. Kriegman, G. Hager and A. Morse), pp. 66–78, Vol. 237 of *LNCIS Series*, Springer Verlag.

Chaumette, F. (2004). Image moments: a general and useful set of features for visual servoing. *IEEE Transactions on Robotics*, **20**(4): 713–723.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1998). Active appearence models. *European Conference on Computer Vision*, Vol. 2, pp. 484–498.

Cowan, N. J. and Chang, D. E. (2002). Toward geometric visual servoing. In *Control Problems in Robotics* (eds A. Bicchi, H. Christensen and D. Prattichizzo), pp. 233–248, Vol, 4 *STAR, Springer Tracks in Advanced Robotics*, Springer Verlag.

Deguchi, K. (1998). Optimal motion control for image-based visual servoing by decoupling translation and rotation. *IEEE International Conference on Intelligent Robots and Systems*, Vol. 2, pp. 705–711.

Drummond, T. and Cipolla, R. (1999). Visual tracking and control using Lie algebras. *IEEE International Conference on Computer Vision and Pattern Recognition*, Fort Collins, CO, June, Vol. 2, pp. 652–657.

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

Faugeras, O. (1993). *Three-dimensionnal Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA.

Gleicher, M. (1997). Projective registration with difference decomposition. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 331–337.

Hager, G. D. and Belhumeur, P. N. (1998). Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(10): 1025–1039.

Hartley, R. (1992). Estimation of relative camera positions for uncalibrated cameras. In *European Conference on Computer Vision* (ed. G. Sandini), pp. 579–587, Vol. 588 of *Lecture Notes in Computer Science*, Springer-Verlag.

Hashimoto, K. (1993). *Visual Servoing: Real Time Control of Robot Manipulators based on Visual Sensory Feedback*, Vol. 7 of *World Scientific Series in Robotics and Automated Systems*. World Scientific, Singapore.

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

Isard, M. and Blake, A. (1996). Contour tracking by stochastic propagation of conditional density. *European Conference on Computer Vision*, pp. 343–356.

Jurie, F. and Dhome, M. (2002). Hyperplane approximation for template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(7): 996–1000.

Longuet-Higgins, H. C. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature*, **293**: 133–135.

Lucas, B. and Kanade, T. (1981). An iterative image registration technique with application to stereo vision. *International Joint Conference on Artificial Intelligence*, pp. 674–679.

Malis, E. (2004). Improving vision-based control using efficient second-order minimization techniques. *IEEE International Conference on Robotics and Automation*, New Orleans, LA, April.

Malis, E. and Chaumette, F. (2002). Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods. *IEEE Transactions on Robotics and Automation*, **18**(2): 176–186.

Malis, E., Chaumette, F., and Boudet, S. (1999). 2 1/2 d visual servoing. *IEEE Transactions on Robotics and Automation*, **15**(2): 234–246.

Malis, E., Chaumette, F., and Boudet, S. (2000). 2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *International Journal of Computer Vision*, **37**(1): 79–97.

Malis, E. and Rives, P. (2003). Robustness of image-based visual servoing with respect to depth distribution errors.

IEEE International Conference on Robotics and Automation, Taipei, pp. 1056–1061.

Martinet, P., Daucher, N., Gallice, J., and Dhome, M. (1997). Robot control using monocular pose estimation. *Workshop on New Trends In Image-Based Robot Servoing*, Grenoble, France, September, pp. 1–12.

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

Sclaroff, S. and Isidoro, J. (1998). Active blobs. *IEEE International Conference on Computer Vision*, pp. 1146–1153.

Shi, J. and Tomasi, C. (1994). Good features to track. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 593–600.

Shum, H. Y. and Szeliski, R. (2000). Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, **16**(1): 63–84.

Taylor, C. J., Ostrowski, J. P. and Jung, S. H. (2000). Robust vision-based pose control. *IEEE International Conference on Robotics and Automation*, pp. 2734–2740.

Torr, P. H. S. and Zisserman, A. (1999). Feature based methods for structure and motion estimation. In *International Workshop on Vision Algorithms* (eds W. Triggs, A. Zisserman and R. Szeliski), pp. 278–295.

Wilson, W. J., Hulls, C. C. W., and Bell, G. S. (1996). Relative end-effector control using cartesian position-based visual servoing. *IEEE Transactions on Robotics and Automation*, **12**(5): 684–696.