Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems

Image-based positioning with respect to a non-structured scene using 2D image motion

[1,2]Armel Crétual* [1]François Chaumette [2]Giulio Sandini

[1] IRISA / INRIA Rennes
Campus de Beaulieu
35042 Rennes cedex, France

[2] LIRA-Lab, Università di Genova
Viale Francesco Causa, 15
16145 Genova, Italy

E-mail: armel@ltsi.univ-rennes1.fr, chaumett@irisa.fr, giulio@lira.dist.unige.it

Abstract

Visual servoing based upon geometrical features such as image points coordinates is now well set on. Nevertheless, this approach has the drawback that it usually needs visual marks on the observed object to retrieve geometric features. The idea developed here is that these features can be retrieved by integrating dynamic ones, which can be estimated without any a priori knowledge of the scene. Thus, more realistic scenes can be used to achieve vision-based control such as positioning tasks. We show that an affine model is insufficient to ensure convergence and that a quadratic one is needed. Finally, results are presented related to the positioning with respect to a complex scene.

1 Introduction

The aim of visual servoing, as presented in [1, 2], is to control the robot displacements using visual features. One of the method used to complete such control laws is to apply the task function approach [3] to visual sensors and is based on the linear relation existing between image features variation and camera motion [4]. Geometric primitives have most often been used until now to complete robotic tasks such as positioning with respect to a given object. For example, visual marks are used in [4] to extract geometric features from the image. Convergence is usually ensured and stable, at least when the initial position is in the 3D neighborhood of the desired one, and most of these applications run at video rate. The major problem encountered is that an a priori knowledge of the geometric features is needed.

To avoid the use of marked objects, a first improve was to track points of interest (p.o.i). A necessary condition for the success of the vision part, is the fact that these points must be significant, meaning corners where the spatial gradient is high in two different directions. This technique has been used in [5]. The main limitation is the very high sensitivity to occlusion of the point of interest. In [6], the visual features used are parameters of a model of the object contour. In that case, a partial and limited occlusion of the object is admitted. A last improving is to add an estimation of the contour position using the motion in the image. This allows to robustify the previous approach in the case of moving objects. It has been done in [7] and [8], where, in both of these articles, a CAD model of the 3D object is used.

Another solution to avoid the problem of using marked objects is to use the motion in the image. Indeed, such a 2D motion is independent of the scene content. Therefore, as several algorithms are now able to perform the estimation of a model of motion in real-time (meaning fast enough to be implemented in robotic loop), such as the one presented in [9], it is possible to use such an information in a visual servoing scheme. This idea has been exploited in [10, 11, 12].

Two other studies, close one to the other and presented in [13, 14], have been done using the deformations of the object of interest in the image in order to position a camera with respect to this object. The authors use the 6 parameters of the affine deformation of a planar object in order to reduce the complexity of its representation, compared to a B-spline or a snake one. In their case, this model is sufficient, but their exist several limitations, such as the needing of 3D features concerning the orientation and the depth of the object in the first article, and a singularity occurs when the object is parallel to the camera plane in both methods.

Our idea is to recover the position of points in the image, only by integrating their estimated speed. Such an estimation is performed applying a model of motion to the current positions of p.o.i. This technique has already been used in [15] for the pan and tilt tracking of a moving target. In this article, only 2 d.o.f. of the

*Armel Crétual's current address is LTSI, Université de Rennes I, Campus de Beaulieu, 35042 Rennes cedex, France

camera were constrained since only one point was used. Thus, we propose to generalize this approach, by controlling the whole position of the camera (3 translation and 3 rotations) using several points. On the contrary of the methods based on the tracking of points in the image, multiple occlusions are possible since the positions are estimated from a global measurement of the scene motion.

The paper is structured as following. The principle of our approach, incuding the retrieval of position from speed, the control law, is exposed in Section 2. The choice of the p.o.i. is discussed and we show the need to use a 8-parameters quadratic model of motion instead of a 6-parameters affine one. In Section 3, results are displayed concerning the positioning of the camera with respect to a complex scene. That is first, the comparison of the method accuracy when using an affine or a quadratic model of motion, in the case of simple initial errors. Then, complete results are given in the case of a complex initial error. Finally, conclusions and future works are presented in Section 4.

2 Principle

Our aim is to control the robot motion by classical image-based techniques, but without any a priori knowledge on the image content. The proposed solution is to retrieve geometric features by integrating dynamic measurements over time. In practical, this has been performed using the robust multi-resolution algorithm (RMR) presented in [9]. Its robustness is based on the rejection of outliers and allows to reach a weakly noisy measurment of motion. Therefore, potential drift problems due to integration can be avoided.

2.1 Retrieval of the 2D features

Let us denote $s = (x, y)^T$, the 2D projection at time t of a 3D point M, and $\dot{s}$ its apparent speed in the image. s can obviously be recovered knowing its projection position s_0 at time 0 and the evolution of $\dot{s}$ over time, by:

$$s = s_0 + \sum_{i=1}^{k} \dot{s}_i \, \delta t_i \qquad \text{(in discrete form)} \qquad (1)$$

with $\dot{s}_i$ being the i^{th} measurement of $\dot{s}$ and δt_i, the time duration between (i-1)th and ith measurements, provided by the computer clock.

The motion model used to approximate speed in the image can be for example a quadratic one with 8 parameters as the following one (see [16]):

$$\begin{cases} \dot{x} &= c_1 + a_1 x + a_2 y + q_1 x^2 + q_2 xy \\ \dot{y} &= c_2 + a_3 x + a_4 y + q_3 y^2 + q_4 xy \end{cases} \qquad (2)$$

with:

$$\begin{cases} c_1 = -\frac{T_x}{Z_p} - \Omega_y & c_2 = -\frac{T_y}{Z_p} + \Omega_x \\ a_1 = \gamma_1 \frac{T_x}{Z_p} + \frac{T_z}{Z_p} & a_2 = \gamma_2 \frac{T_x}{Z_p} + \Omega_z \\ a_3 = \gamma_1 \frac{T_y}{Z_p} - \Omega_z & a_4 = \gamma_2 \frac{T_y}{Z_p} + \frac{T_z}{Z_p} \\ q_1 = q_4 = -\gamma_1 \frac{T_x}{Z_p} - \Omega_y & q_2 = q_3 = -\gamma_2 \frac{T_x}{Z_p} + \Omega_x \end{cases}$$

where $T = (T_x, T_y, T_z)$ and $\Omega = (\Omega_x, \Omega_y, \Omega_z)$ represent the translation and the rotation speeds between the camera and the object frames. $Z = Z_p + \gamma_1 X + \gamma_2 Y$ is the equation of the planar approximation of the object surface around the considered point, expressed in the camera frame.

Of course, other models, e.g. constant (the restriction of the presented one to terms c_i) or affine (the restriction to terms c_i and a_i) could be used. In fact, there is a necessary compromise to find between accuracy of the estimation and computation load.

Finally, s can be, not only a single point, but a set of n points of the image. In the following, it will always be the case. Their real 2D-coordinates on the i-th image will be denoted s_i and their estimations σ_i, with:

$$\begin{cases} s_i = & (x_{1,i}, \ldots, x_{n,i}, y_{1,i}, \ldots, y_{n,i})^T \\ \sigma_i = & (\xi_{1,i}, \ldots, \xi_{n,i}, \psi_{1,i}, \ldots, \psi_{n,i})^T \end{cases}$$

2.2 Control law

Having an estimation of the position of several points of the image, it is possible to define a control law, in order to bring them to a desired position. To do so, we use the principle of the virtual rigid link from [4]. A priori, a set of three points is sufficient to ensure such a link where 6 d.o.f. are constrained (3 rotations, and 3 translations). Nevertheless, it has been shown in [17] that the system can reach singularities, in particular when the optical center is on the cylinder defined by the circle circumscribed to these points, with a direction orthogonal to the plane of the triangle. Moreover, for one position of three points in the image corresponds four positions of the camera in the 3D frame. Therefore, a redundant information based on four points is used, even if in that case, local minima can be reached, if the initial position is far from the desired one [18].

Vector s is linked to the camera motions by the interaction relation:

$$\dot{s} = L_s \; (T_x, T_y, T_z, \Omega_x, \Omega_y, \Omega_z)^t$$

where (T, Ω) is the interaction screw of the camera.

Let s^* be the desired positions of the points. The task vector (or error vector) is defined as $e = C(s - s^*)$, where C is a constant chosen matrix. An exponential decay of this error leads to the control law:

$$(T, \Omega)^T = -\lambda \left(C \, \widehat{L} \right)^{-1} \widehat{e} \qquad (3)$$

where $\widehat{e}$ is a measure of e and $\widehat{L}$ is an approximation of L_s, the interaction matrix related to s, which corresponds to its value at convergence using a "manual" estimation $\widehat{Z_i}$ of the depth Z_i of each point.

In practical, the same value is given to each of these $\widehat{Z_i}$. Finally, $\widehat{L}^+$ is the pseudo-inverse of this approximation and C is chosen equal to $\widehat{L}^+$. The control is thus reduced to:

$$(T,\Omega)^T = -\lambda\, \widehat{e}$$

Theoretically, the exponential decay will be ensured and will remain stable under the sufficient condition [4]:

$$\widehat{L}^+ L_s > 0$$

However, the error vector e is in fact not equal to $C(s - s^*)$ but to $C(\sigma - s^*)$. This difference has an influence on the true interaction relation between used measurements and the camera motion, and therefore on the stability. If we write $\dot{\sigma} = L_\sigma (T,\Omega)^T$, the stability condition is in fact:

$$\widehat{L}^+ L_\sigma > 0 \qquad (4)$$

and it will be seen in Section 2.4 that such condition explains why an affine model is inadequate in some cases.

Only static objects were considered in this paper because of computational load for each iteration. Target tracking method was previously developed using dedicated objects [20]. The same method could be used on real objects if the current time processing was not so important and incompatible with the real-time constraints of tracking.

2.3 Choice of the points of interest (p.o.i.)

There are only two conditions for the choice of p.o.i.: they should appear both in the desired and initial images and they must not be a set of three aligned points.

In practical, to ensure a good observation of the scene deformations, the points are chosen sufficiently far one from the other. Moreover, the initial matching between points on the desired and initial images is performed semi-automatically. This means that an extraction of several characteristic points is made in the two images. This extraction is performed using the classical Harris and Stephens method [19]. Then, the operator chooses four of them, satisfying the previous conditions. Let us denote here, that the precision of this extraction is only around one pixel.

Another thing to be noticed, is that the only use of these points is to define the initial error. As their position is then estimated using the global motion of the scene, there is no need to track them along the task. Therefore, they can be hidden without disturbing the control, even at convergence. They can even go out of the camera field of view during the servoing.

2.4 Motion model: affine vs. quadratic

As specified in section 2.1, to retrieve the position of each point, the speed is approximated using a polynomial model of motion. An important question is to define which model should be usedsince there is a compromise to find between the swiftness of the estimation and its accuracy. In our case, six d.o.f. of the robot are constrained. Therefore, a constant model of motion, that has been previously used in a tracking task constraining only two rotational d.o.f. [15], is here heavily insufficient.

The affine model includes six parameters. Nevertheless, it is not enough to ensure the correct positioning. Actually, using such a model to estimate the points positions does not allow to make the distinction between a translation along $\overrightarrow{x}$(resp. along $\overrightarrow{y}$) and a rotation around $\overrightarrow{y}$(resp. around $\overrightarrow{x}$). More precisely, in that case the link between the discrete variation of s (meaning the approximation of $\dot{s}$ by $(s_{k+1} - s_k)/\delta t$) is linked to the camera motion by the following matrix L_σ:

$$L_\sigma = \begin{bmatrix} -1/Z_1 & 0 & \xi_1 & 0 & -1 & \psi_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -1/Z_4 & 0 & \xi_4 & 0 & -1 & \psi_4 \\ 0 & -1/Z_1 & \psi_1 & 1 & 0 & -\xi_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & -1/Z_4 & \psi_4 & 1 & 0 & -\xi_4 \end{bmatrix}$$

It can never be ensured that the rank of this matrix would be 6. Therefore, if the rank is inferior to 6, the stability condition (4) cannot be respected, as the product $L_\sigma \widehat{L}^+$ must be strictly positive. In particular, when all the Z_i values are equal, meaning when the camera is parallel to the scene, the rank of this matrix is only 4. In such a case, combinations of T_x and Ω_y (resp. T_y and Ω_x) appear in its null space. This explains the singularity encountered in [13, 14].

On the contrary, if the considered model of motion is the simplified quadratic one, the corresponding matrix L_σ is equal to:

$$L_\sigma = \begin{bmatrix} -1/Z_1 & 0 & \xi_1 & \xi_1\psi_1 & -1-\xi_1^2 & \psi_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -1/Z_4 & 0 & \xi_4 & \xi_4\psi_4 & -1-\xi_4^2 & \psi_4 \\ 0 & -1/Z_1 & \psi_1 & 1+\psi_1^2 & -\xi_1\psi_1 & -\xi_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & -1/Z_4 & \psi_4 & 1+\psi_4^2 & -\xi_4\psi_4 & -\xi_4 \end{bmatrix}$$

The rank of this matrix is always 6. Therefore, we can conclude that it is necessary to use this model.

3 Results

The task has been implemented on a 6 d.o.f. eye-in-hand system. The scene which was used is presented on figure 1 in the desired configuration. It is composed of a main plane with several 3D objects laying on it. It is a "real" one in the sense that no dedicated object is added. Images were acquired by a SunVideo board and all the computation was made on a 170 MHz UltraSparc station. The size of the images was 256×256 pixels which leads to 500 ms iteration for an affine model estimation and 800 ms ones for a quadratic one.

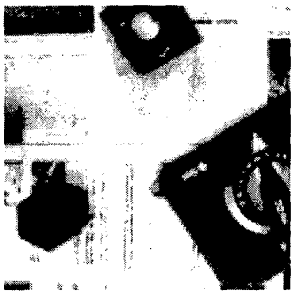

Figure 1: Observed scene at desired position

3.1 Affine vs. quadratic model

The aim of the first experiment was to prove the necessity of using the quadratic model of motion. Different initial positions have been considered corresponding to simple motions of the camera, meaning a displacement on only one d.o.f.. Results concerning these experiments are displayed on table 1. In the first column, the displacement between the initial position and the desired one is given in the fixed frame. The desired position corresponds to the position $\tau = (0, 0, 0)$ and the orientation $\omega = (0, 0, 0)$ in this same frame. The second and third column gives the final position and orientation of the camera using respectively the affine and the quadratic model of motion. Positions are given in mm and orientation in deg.

The first thing to notice is that using the quadratic model gives strongly better results than using the affine one. In the first case, the average error is around 10 mm in position and less than 1 deg in orientation. On the contrary, using the affine model, these errors reach more than 170 mm in position and 12 deg in rotation, and a divergence is even possible (for example in case of a rotation around $\vec{x}$). One can also notice that, when the affine model is used, an error in translation along $\vec{x}$ is always combined with an error in rotation around $\vec{y}$ and vice versa. This proves experimentally the necessity of using the quadratic model of motion. The small residual errors, when using this model, can easily be explained by the weak precision of the initial extraction of the points (about 1 pixel whereas it is about a tenth of a pixel with dedicated objects).

3.2 A result sequence

The aim of the following experiment is to show the accuracy of our method, even when the initial error is large. In that case, only the quadratic model has been used as it was shown previously that the affine one is inadequate. Several curves related to this experiment are displayed below on figure 2. First, one can notice that despite the quite important error at the beginning (nearly a hundred pixels), the convergence is obtained. It is performed without oscillations and it remains stable once convergence is reached.

One image upon ten of the sequence acquired is displayed on figure 3. The difference between the initial and desired positions of the camera is important: T = (300, 350, -150) (in mm), Ω = (25, -20, 25) (in deg). This can be noticed by the large disparity between initial and desired images. On each of image of the sequence, the estimated p.o.i. are designated by a target sign. It can be seen that they do not all correspond to corners with a high spatial gradient. Moreover, on several of these images, some of them have been hidden.

It can be noticed that convergence is reached despite the occlusion of several p.o.i.. In some cases (for example in the image at the upper right) three of the four points are hidden. This only leaves one point visible, which is highly insufficient for techniques based on the points tracking. Comparing the last image of the sequence, meaning the one obtained at convergence, to the desired one presented on figure 1, one can obviously notice that it is very close.

4 Conclusion

In this paper, we have presented a new method to position a camera in front of a complex scene. The idea exploited is that the position of a point can be retrieved by integration, as soon as its speed can be estimated at each iteration and its initial position is known. To constrain 6 d.o.f., 4 points are thus used.

The main advantage of our method is that the correspondence between the current points positions and the desired one has to be performed only once, at the beginning. Therefore, there is no need to track the p.o.i. in the image at each iteration. Moreover, our method is not sensitive to occlusion. An important thing to underline in our approach is the necessity to estimate 8 parameters of motion, even if only 6 d.o.f. are constrained. This is due to the insufficiency of the 2D-affine

Initial motion	Final position (affine model)	Final position (quadratic model)
T_x = + 100 mm	τ = (-36, +133, +20) ω = (+9.2, +2.1, 0.0)	τ = (-9, +15, +7) ω = (+1.3, +1.0, +0.2)
Ω_x = + 10 deg	divergence (position when a joint limit is reached) τ = (+64, -358, +276) ω = (-25.6, -4.6, +1.0)	τ = (-5, +12, +5) ω = (+0.9, +0.4, -0.1)
T_z = - 150 mm	τ = (-98, -50, +11) ω = (-3.7, +6.8, +0.6)	τ = (+13, -8, 0) ω = (-0.6, +0.8, +0.4)
Ω_z = + 30 deg	τ = (-84, +178, +34) ω = (+12.5, +5.8, -0.1)	τ = (-22, +6, -2) ω = (0.0, +1.4, +0.5)

Table 1: Comparison of positions at convergence when using the affine or the quadratic model of motion

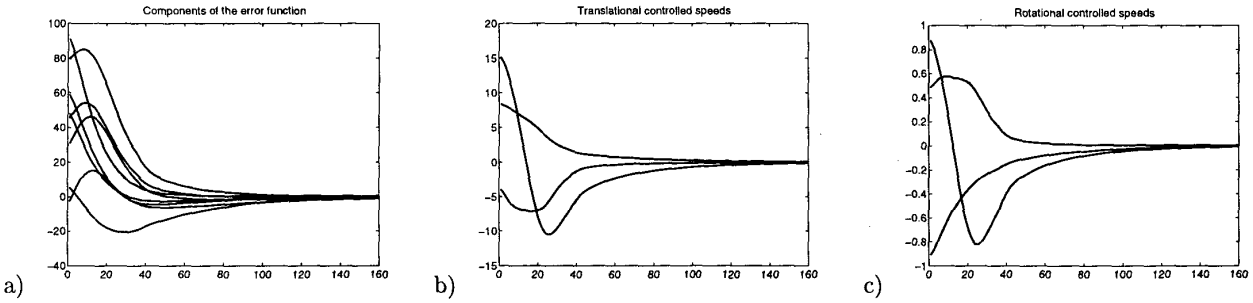

Figure 2: Positionning result: a) $s - s^*$ (in pixel), b) Tx, Ty, Tz (in mm/s), c) $\Omega_x, \Omega_y, \Omega_z$ (in deg/s)

model of motion to express all the links between 3D motions of the camera and the corresponding 2D ones.

Future works will be held on two levels. First, on the practical stage, the initial vision process of points matching will be improved in an automatic way. Finally, to provide more robustness to our approach, we will study the case of less planar scenes.

Acknowledgments

Part of the work presented here has been done while Armel Crétual was post-doctoral fellow in LIRA-lab at Genoa University, Italy, thanks to Giulio Sandini.

References

[1] K. Hashimoto, ed., *Visual servoing. Real-time control of robot manipulators based on visual sensory feedback.* World scientific series in robotics and automated systems, World scientific, 1993.

[2] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. Robotics Automation*, vol. 12, pp. 651–670, Oct. 1996.

[3] C. Samson, M. Le Borgne, and B. Espiau, *Robot control : the task function approach.* Oxford University Press, 1991.

[4] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robotics Automation*, vol. 8, pp. 313–326, June 1992.

[5] G. Hager, "A modular system for robust hand-eye coordination using feedback from stereo vision," *IEEE Trans. Robotics Automation*, vol. 13, pp. 582–595, Aug. 1997.

[6] E. Coste-Manière, P. Couvignon, and P. Khosla, "Visual servoing in the task-function framework: a contour following task," *Journal of Intelligent Robotic Systems*, vol. 12, pp. 1–22, Jan. 1995.

[7] T. Drummond and R. Cipolla, "Real-time tracking of complex structures with on-line camera calibration," in *BMVC'99*, pp. 574–583, Sept. 1999.

[8] F. Keçeci, M. Tonko, H. Nagel, and V. Gengenbach, "Improving visually servoed disasembly operations by automatic camera placement," in *ICRA'98*, (Leuven, Belgium), pp. 2947–2952, May 1998.

[9] J. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *Journal of Visual Communication and Image Representation*, vol. 6, pp. 348–365, Dec. 1995.

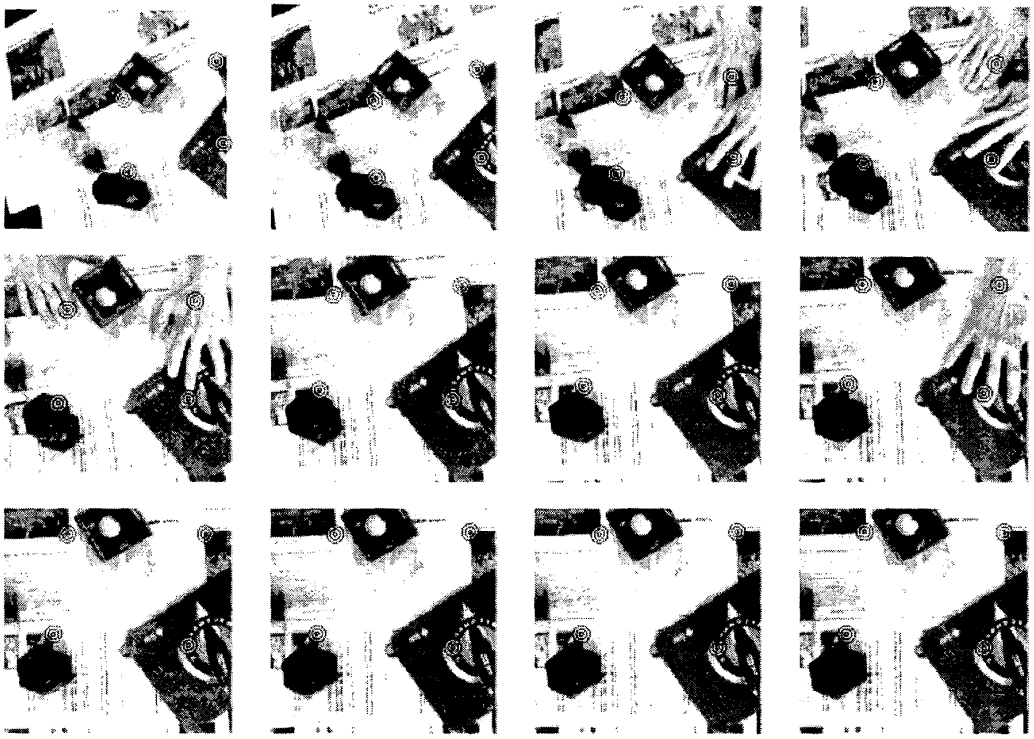

Figure 3: One image upon ten of the sequence with the estimated position of p.o.i. marked by a target symbol

[10] A. Crétual and F. Chaumette, "Positionning a camera parallel to a plane using dynamic visual servoing," in *IROS'97*, vol. 1, (Grenoble, France), pp. 43–48, Sept. 1997.

[11] P. Questa, E. Grossmann, and G. Sandini, "Camera self orientation and docking maneuver using normal flow," in *SPIE AeroSense'95*, (Orlando, Florida), Apr. 1995.

[12] V. Sundareswaran, P. Bouthemy, and F. Chaumette, "Exploiting image motion for active vision in a visual servoing framework," *IJRR*, vol. 15, pp. 629–645, Dec. 1996.

[13] C. Colombo and B. Allotta, "Image-based robot task planning and control using a compact visual representation," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 29, pp. 92–99, Jan. 1999.

[14] T. Drummond and R. Cipolla, "Visual tracking and control using Lie algebra," in *CVPR*, vol. 2, (Fort Collins (CO)), pp. 652–657, June 1999.

[15] A. Crétual and F. Chaumette, "Image-based visual servoing by integration of dynamic measurements," in *ICRA'98*, vol. 3, (Leuven, Belgique), pp. 1994–2001, May 1998.

[16] M. Subarrao and A. Waxman, "Closed-form solutions to image equations for planar surface in motion," *Computer Vision, Graphics, and Image Processings*, vol. 36, pp. 208–228, Nov. 1986.

[17] H. Michel and P. Rives, "Singularities in the determination of the situation of a robot effector from the perspective view of 3 points," Rapport de Recherche 1850, INRIA, Feb. 1993.

[18] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The Confluence of Vision and Control* (G. Hager, D. Kriegman, and A. Morse, eds.), LNCIS Series, Springer-Verlag, 1998.

[19] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Conference*, pp. 189–192, 1988.

[20] F. Chaumette and A. Santos, "Tracking a moving object by visual servoing," in *IFAC-WC'98*, vol. 9, (Sydney, Australia), pp. 409–414, July 1993.