

Visual feedback in camera motion generation: Experimental results

F. Berry

Student Member IEEE

P. Martinet

J. Gallice

Laboratoire des Sciences et Matériaux pour l'Electronique, et d'Automatique.
Université Blaise Pascal de Clermont-Ferrand,
U.R.A. 1793 du C.N.R.S., F-63177 Aubière Cedex, France
E-Mail : Martinet, Berry, Gallice@lasmea.univ-bpclermont.fr

Abstract

In this paper, we propose several results about trajectory generation by visual servoing. The approach consists in defining a specific task function which allows us to take into account the time varying aspect of the reference feature and we synthesize a control law in the sensor space. This control law ensures the trajectory control in the image space and reduces the tracking error. Under specific conditions, the trajectory of the camera can be ensured in the robot workspace. The main goal of this work is to demonstrate the effectiveness of this approach through experimental results. During experiments, we have used a cartesian robot and a real time vision system. A CCD camera is mounted on the end effector of the robot. We present two types of trajectory. The first is an helical trajectory parallel to a cube side. The second experiments consists in passing around a cube. This last trajectory is built by linking several elementary trajectories (rotation and translation).

1 Introduction

This paper shows the use of a visual servoing in order to control the trajectory of a camera. The originality of this work is based on the trajectory tracking in image space. The “visual servoing” approach allows to introduce the sensor information directly in the control loop [3] [4]. Samson *et al* in [11] developed the formalism of the *task function* where the control is directly specified in terms of regulation in the sensor space. The robotic task to achieve is a positioning task in relation with a fixed or a moving target. Many works were done using a moving target object [2]. In this case, the use of a predictive filter (i.e Kalman filter) is absolutely essential to ensure

an accurate object tracking. For motionless objects, in visual servoing approach, we find many works [3], [12], [6] and [8]. The use of the *task function* concept defined in sensor space allows to introduce the notion of hybrid task. This task is made of a combination of a primary task, which realizes the visual servoing, and a secondary task. The secondary task can be considered as a minimization of a cost function. The main applications of a secondary task are visual tracking [10], singularities and joints limits avoidance [7]. The secondary task uses the unconstrained d.o.f. The velocities used in the secondary task are in open loop, under the constraint of a perfect achievement of the primary task.

For the last couple of years, researches in the field of motion control with a visual sensor proposed different approaches. Tsuji *et al.* in [14], use a stereo vision system to navigate in an indoor environment. A stereo image analyzer determines vertical projections of edge points on the floor in the image and a path planner guides the robot with local maps. Recently Swain in [13] proposed a method based on visual servoing and on potential field. These results concern a linkage of four visual servoing tasks applied to a mobile robot.

To solve the problem of camera motion generation, a first approach consists in expressing the control law in the workspace. Some papers [15] [5] [9] have presented different 3D approaches using a monocular camera as the main sensor. The problem in these approaches is to implement the reconstruction algorithms. To avoid these difficulties, we propose to control the camera motion directly in image space.

In the first part of this paper, we remind of the fundamental basis of the trajectory tracking developed in [1]. In a second part, several kind of trajectories controlled by visual servoing are generated and suc-

cessfully implemented with our experimental platform. The problem of linkage between the trajectories is tackled.

2 Theoretical basis

In a previous work [1], we have proposed the theoretical development in order to perform a trajectory tracking. The original contribution of this approach is to control the trajectory directly in the visual sensor space. Two main points have been considered: the control law which ensures the tracking along the trajectory and the generation of the reference trajectory. Next, we just remind the main results concerning both points.

Control law For the control law, we have used the task function formalism [11] and we have defined the following task function:

$$\underline{e}(\underline{s}(\underline{r}, t), t) = C [\underline{s}(\underline{r}, t) - \underline{s}^*(t)] \quad (1)$$

where $\underline{s}(\underline{r}, t)$ is the vector of measured features in image space, $\underline{s}^*(t)$ is the desired trajectory expressed in image space and C is a constant matrix which allows to take into account more visual information than the d.o.f. to control. Visual information $\underline{s}(\underline{r}, t)$ depend on the situation (noted \underline{r}) between the sensor (camera) and the observed scene. They may also depend on the time, if the target object is in motion. In our case, we consider a motionless environment so we have $\frac{\partial s}{\partial t} = 0$ and the variations of $\underline{s}(\underline{r}, t)$ are given by the following differential relation:

$$\frac{d\underline{s}(\underline{r}(t), t)}{dt} = \frac{\partial \underline{s}}{\partial \underline{r}} \cdot \frac{d\underline{r}}{dt} \quad (2)$$

where $\frac{d\underline{r}}{dt} = T = (\vec{V}, \vec{\Omega})$ is the kinematic screw.

T represents the relative velocity between the camera and its environment. The term $\frac{\partial s}{\partial \underline{r}} = L^T$, called interaction matrix or image jacobian, characterizes the interaction between the sensor and its environment.

In considering an exponential decay of $\underline{e}(\underline{r}, t)$:

$$\dot{\underline{e}}(\underline{r}, t) = -\lambda \underline{e}(\underline{r}, t) \quad (3)$$

(λ is a positive scalar constant) and with the choice of $C = \hat{L}^{T+}$ (estimate of L^T compute for $\underline{s} = \underline{s}^*$) the expression of the control law becomes:

$$T = -\lambda \hat{L}^{T+} \cdot (\underline{s}(\underline{r}, t) - \underline{s}^*(t)) + \hat{L}^{T+} \frac{d\underline{s}^*(t)}{dt} \quad (4)$$

The first term ensures the visual servoing to maintain a rigid link between sensor and target. The second term expresses the influence of the trajectory generation. It allows to compensate the tracking error with a high efficiency.

Trajectory generation The visual reference feature $\underline{s}^*(t)$ corresponding to the desired trajectory can be built from two transformations presented on the figure 1.

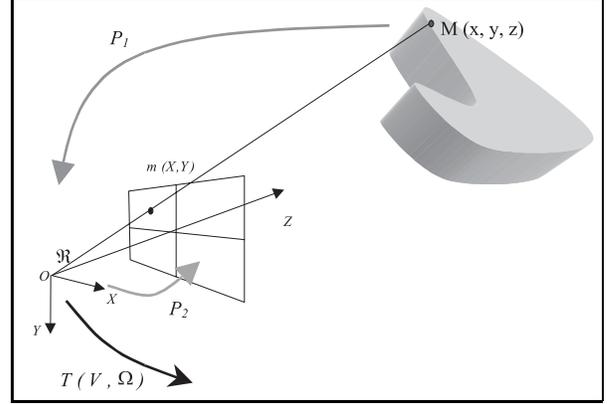


Figure 1: Simplified model of our configuration

The first transformation noted P_1 , expresses the coordinates of a point M linked rigidly to a fixed object with coordinates (x, y, z) in the moving effector frame \mathfrak{R} . The velocity screw $T = (\underline{V}, \underline{\Omega})$ is applied to \mathfrak{R} . If we use the matrix notation (i.e $\underline{p} = (x, y, z)^T$, $\underline{V} = (Vx, Vy, Vz)^T$, $\tilde{r} \equiv \vec{r} \wedge \cdot$), the evolution of this point is given by the well known kinematic equation:

$$\frac{d}{dt} \underline{p} = -V - \omega \tilde{r} \underline{p} \quad (5)$$

where ω is the rotation velocity, and \tilde{r} is the unitary anti symmetric matrix of the rotation axis. As long as \underline{V} and $\omega \tilde{r}$, expressed in the camera frame \mathfrak{R} , are constant, the general solution of equation 5 is given by:

$$\underline{p}(t) = \left(I_3 - \sin(\omega \Delta t) \tilde{r} + (1 - \cos(\omega \Delta t)) \tilde{r}^2 \right) \underline{p}(t_0) - \left(I_3 - \frac{1 - \cos(\omega \Delta t)}{\omega \Delta t} \tilde{r} + \frac{\omega \Delta t - \sin(\omega \Delta t)}{\omega \Delta t} \tilde{r}^2 \right) \underline{V} \Delta t \quad (6)$$

This relation expresses the position of the point M ($\underline{p} = \overrightarrow{OM} = (x, y, z)^T$) at time t with the knowledge of the position and the orientation of the point M at

time t_o , the velocity screw $T = (\vec{V}, \vec{\Omega})$ and the period sampling $\Delta t = t - t_o$.

The second transformation noted P_2 , allows to express the projection $m(X,Y)$ of the point $M(x,y,z)$ (Pinhole Camera Model) in the centered image frame:

$$X(t) = \frac{x(t)}{z(t)}.F_x \quad \text{and} \quad Y(t) = \frac{y(t)}{z(t)}.F_y \quad (7)$$

where F_x and F_y are the focal lengths of the camera.

3 Experimental results

Our robotic platform is a cartesian robot with 6 degrees of freedom (built by the firm AFMA Robot). It is composed by 3 axis of translation and 3 axis of rotation. A CCD camera is mounted on the end effector of the robot and is connected to the vision system WINDIS. This whole platform is controlled by a VME system, and can be programmed in C language under VxWorks real time operating system.

3.1 Experimental context

The target is composed of a cube (the length of the edge is 25cm) and 8 LED (Fig.2). Two LED locate each edge and we choose the object frame centred in the cube. In this frame, the 8 LED have the following coordinates: $x = \pm 12.5cm$, $y = \pm 7.5cm$, $z = \pm 12.5cm$.

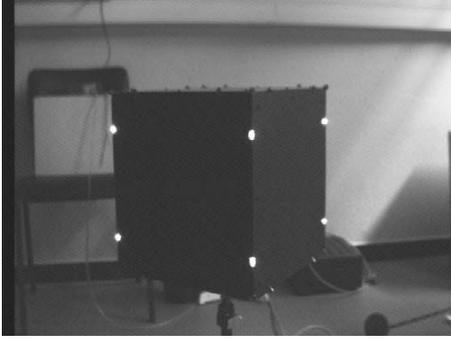


Figure 2: Camera view of the target

The application runs at video rate (40 ms), but the vision system introduces a data flow latency of 80 ms. The vision process is based on the extraction of the center of gravity of the illuminated points and a specific algorithm allows to sort the features used at each step of the trajectory. A coarsely calibrated vision system¹ has been used for all the experimentations and

¹Only F_x , F_y and u_0 , v_0 are known.

the estimation of the depth of the 8 LED used in visual servoing is not accurate.

3.2 Circular helix parallel to a cube side

In this experimentation, we perform a circular helix parallel to a cube side using the sensor signal provided by the camera. In a first step, we perform a positioning task with 67.5 cm between the side and the effector, and then we execute the trajectory. It is composed by a circular translation around the z axis and a translation along the same axis (Fig. 3). The velocities in the robot workspace are: $V_x = -\omega R \sin(\omega t)$, $V_y = \omega R \cos(\omega t)$, V_z .

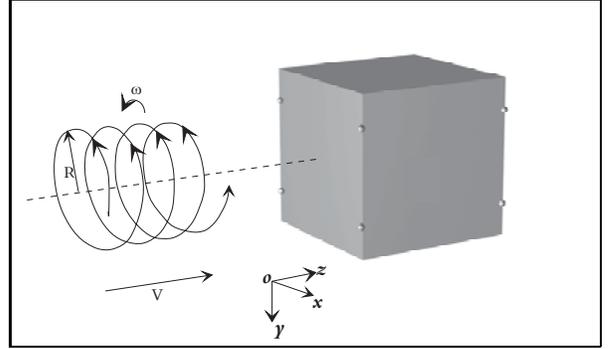


Figure 3: Circular helix trajectory

We want to keep the orientation of the effector, so we have $\omega=0$. The relation 6 allows us to calculate the first transformation P_1 which expresses the coordinate of the reference feature in the effector frame:

$$\underline{p}(t) = \underline{p}(t_o) - \int_{\alpha=t_o}^{\alpha=t} \underline{V}(\alpha) d\alpha \quad (8)$$

and we obtain on each axis:

$$\begin{cases} x(t + \Delta t) = x(t) - R \cos(\omega t) \Delta t \\ y(t + \Delta t) = y(t) - R \sin(\omega t) \Delta t \\ z(t + \Delta t) = z(t) - V_z \Delta t \end{cases}$$

We apply the transformation P_2 in using the pinhole camera model, and we have:

$$X^*(t) = \frac{x(t) - R \sin(\omega t) \Delta t}{z(t) - V_z \Delta t}.F_x$$

$$Y^*(t) = \frac{y(t) - R \cos(\omega t) \Delta t}{z(t) - V_z \Delta t}.F_y$$

The parameters of this experiment are:

- Radius of the helix $R=3$ cm
- Translation velocity $V_z=2$ cm.s⁻¹
- Pulsation rate $\omega=\frac{4\pi}{5}$ rad.s⁻¹

- Gain of the control loop $\lambda=0.2$

On the graph 4 we present a view of the measured trajectory and reference trajectory. For this experimentation, we use a coarsely calibrated vision system what explains the little deformations.

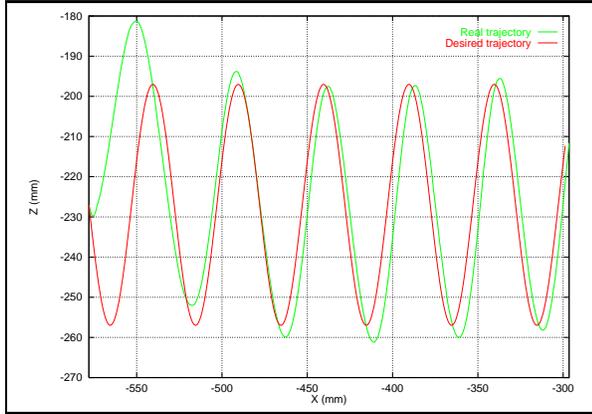


Figure 4: Right View of the trajectory

In Fig. 5, the translation velocities applied to the effector are presented. These velocities are not sensitive to the noise on the measurements, the comparison between the desired parameters et the mesured values shows that the trajectory tracking is well performed.

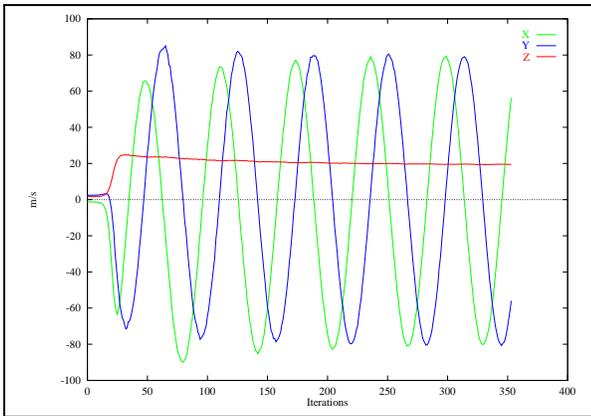


Figure 5: Translation velocity of the effector

In the Table 1, we compare the desired value of each velocity with the average error. This error is computed as follows:

$$E = \frac{1}{n} \sum_{i=1}^n \sqrt{(T_i - T_i^*)^2}$$

where $T_i = \{V_x, V_y, V_z, R_x, R_y, R_z\}$ is the i^{th} value of the measured velocity and T_i^* is the i^{th} value of the desired velocity.

	$V_{x_{max}}$ (mm/s)	$V_{y_{max}}$ (mm/s)	V_z (mm/s)	R_x (deg/s)	R_y (deg/s)	R_z (deg/s)
Desired Value	75.4	75.4	20	0	0	0
Average Error	5.51	7.20	0.38	0.31	0.14	0.006

Table 1: Error on Velocity

In this table, the error along the y axis is twice as big than x axis. This difference is due to the focal factor (≈ 1.5). More, the extraction of features (points) are realized in one frame (512x256 pixels), so the sub-sampling on the y axis involves that the center of gravity is more noisy in the y direction. This explains why the average error on V_y (resp. R_x) is bigger than the one on V_x (resp. R_y).

3.3 Complex trajectory

3.3.1 Description of the trajectory

We perform a complex trajectory by linking three single trajectories. This task consists in passing around an edge of a cube (Fig. 6).

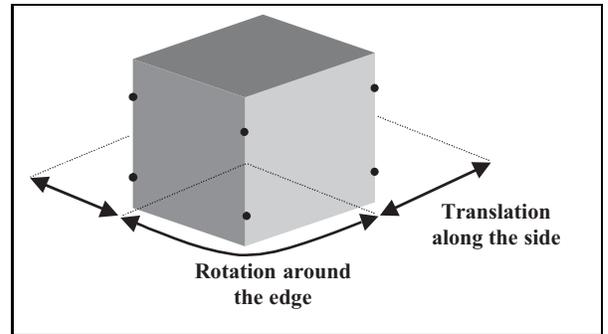


Figure 6: Overview of the trajectory

For the transition step between the trajectories j and $j+1$, we used the following equation to combine both the corresponding kinematic screws T_j and T_{j+1} :

$$T = \gamma(t)T_j + (1 - \gamma(t))T_{j+1} \quad (9)$$

where $\gamma(t)$ is a sigmoid function such $\gamma(0) = 1$ and $\gamma(\tau) = 0$, (τ sets the transition speed between the both trajectories ($0 < t < \tau$)).

The three single trajectories are a translation along the side 1, a rotation around the edge between the side 1 and 2 and a translation along the side 2. In the next parts, we present how to compute the time varying image feature for each element of the trajectory.

Translation

This task consists in sliding along the cube sides. For

this trajectory, we use the relation 6 which can be rewritten as:

$$\underline{p}(t) = \underline{p}(t_o) - \underline{V}(t - t_o) \quad (10)$$

We consider $V_y=V_z=0$, so the first transformation P_1 which expresses the coordinates of the reference feature in the effector frame is:

$$\begin{cases} x(t + \Delta t) = x(t) - V_x \Delta t \\ y(t + \Delta t) = y(t) \\ z(t + \Delta t) = z(t) \end{cases}$$

We apply the transformation P_2 by using the pinhole camera model and we obtain:

$$X^*(t) = \frac{x(t) - V_x \Delta t}{z(t)} . F_x$$

$$Y^*(t) = \frac{y(t)}{z(t)} . F_y$$

Rotation

This task consists in rotating around the cube edge between both sides.

In this case, if the radius of the rotation \underline{d} is not equal to zero, then our trajectory is composed by a combination of a translation and a rotation and we have $\underline{V} = \omega \tilde{r} \underline{d}$. If \underline{d} is constant in the frame \mathfrak{R} (case of angular motion), then from 6 we get:

$$\underline{p}(t) = \underline{p}(t_o) + \left(-\sin(\omega(t - t_o)) \tilde{r} + (1 - \cos(\omega(t - t_o))) \tilde{r}^2 \right) (\underline{p}(t_o) - \underline{d}) \quad (11)$$

The vector \vec{r} defining the rotational axis is parallel to the y axis, so $\vec{r} = [0, r_y, 0]^T = [0, 1, 0]^T$.

From 11, we get:

$$\begin{cases} x(t + \Delta t) = x(t) \cos(\omega \Delta t) - (z(t) - d) \sin(\omega \Delta t) \\ y(t + \Delta t) = y(t) \\ z(t + \Delta t) = x(t) \sin(\omega \Delta t) + (z(t) - d) \cos(\omega \Delta t) + d \end{cases}$$

We apply the transformation P_2 by using the pinhole camera model and we obtain:

$$X^*(t) = \frac{x(t) \cos(\omega \Delta t) - (z(t) - d) \sin(\omega \Delta t)}{x(t) \sin(\omega \Delta t) + (z(t) - d) \cos(\omega \Delta t) + d} . F_x$$

$$Y^*(t) = \frac{y(t)}{x(t) \sin(\omega \Delta t) + (z(t) - d) \cos(\omega \Delta t) + d} . F_y$$

3.3.2 Results

In this part, we present the experimental results. We use the following parameters:

Translation	Rotation
Trans. vel. $V_x = \pm 2 \text{cm.s}^{-1}$	Rot. vel. $\omega = \pi/5 \text{ rad.s}^{-1}$
Dist. side/camera $d=67.5 \text{cm}$	Rad. of the rot. $d=67.5 \text{cm}$
Gain $\lambda=0.1$	Gain $\lambda=0.1$

For the sigmoid parameter, we choose $\tau=40$.

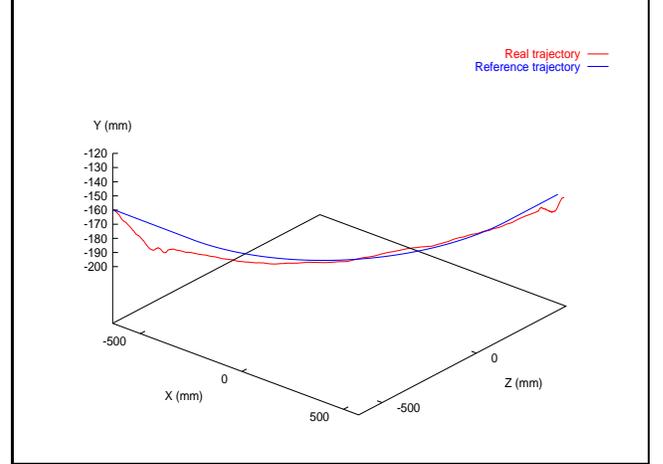


Figure 7: 3D trajectory of the camera.

On Figure 7, we present the real trajectory followed by the camera in the workspace and we can observe that the real and desired trajectories are very closed and smooth. The trajectory is realized with an accuracy of $\pm 1 \text{cm}$. To verify the control of the trajectories we compare the desired and measured features in image space. For instance, Figure 8 represents a comparison between features $X1$ and $X1^*$, $X4$ and $X4^*$. These features are respectively the upper right and lower left points. The visual servoing is well performed and the tracking error is reduced.

4 Discussion

In this paper, we have presented an approach to control the camera motion. In this way, we have expressed the trajectory directly in image space and designed a control law which takes into account the trajectory tracking. We have successfully implemented an helical trajectory and a complex trajectory composed by a sequence of elementary parts (Translation and Rotation). For a well performed tracking the assumptions are based on the existence of a rigid link between the camera and the target. The rigid link is guaranteed

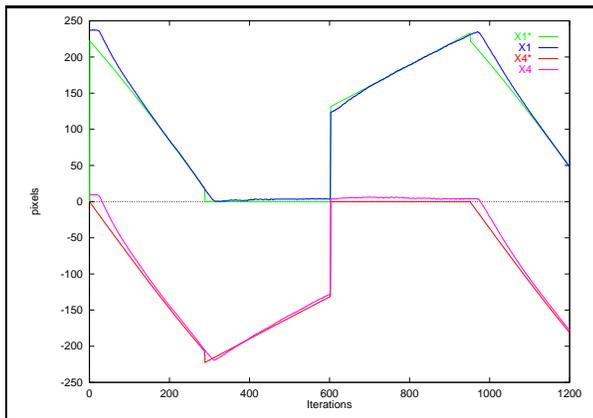


Figure 8: Reference features.

when the image jacobian L^{T+} is a full rank matrix. A necessary condition is the avoidance of visual singularities (for instance a plane is projected in a line,...). During experimentations, the estimation of the depth became required to ensure a full ranking of the image jacobian.

In the future, we plan to analyse the robustness in regard with the calibration and the depth estimation error for this kind of tasks. Another important point could be to study the singularity dues to the ambiguity of the sensor information.

References

- [1] F. Berry, P. Martinet, and J. Gallice. Trajectory generation by visual servoing. In *IEEE Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 1065–1071, Grenoble, France, 7-11 September 1997.
- [2] F. Chaumette and A. Santos. Tracking a moving object by visual servoing. In *IFAC 12th Triennial World Congress*, volume 9, pages 409–414, Sydney, Australia, july 1993.
- [3] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, june 1994.
- [4] J.T. Feddema and O.R. Mitchell. Vision guided with feature-based trajectory generation. *IEEE Trans. on Robotics and Automation*, 5(5):691–700, october 1989.
- [5] E. Grosso, M. Meta, A. Andrea, and G. Sandini. Robust visual servoing in 3d reaching tasks. *IEEE Transactions on Robotics and Automation*, 12(5):732–742, October 1996.
- [6] D. Khadraoui, G. Motyl, P. Martinet, J. Gallice, and F. Chaumette. Visual servoing in robotics scheme using a camera/laser stripe sensor. *IEEE Trans. on Robotics and Automation*, 12(5):743–749, October 1996.
- [7] E. Marchand, A. Rizzo, and F. Chaumette. Avoiding robot joint limits and kinematics singularities in visual servoing. In *13th IARP/IEEE Int. Conf in Pattern Recognition, ICPR'96*, volume A, pages 297–301, Vienna, Austria, August 1996.
- [8] P. Martinet, F. Berry, and J. Gallice. Use of first derivative of geometric features in visual servoing. In *IEEE Int. Conf on Robotics and Automation*, volume 4, pages 3413–3419, Mineapolis, USA, April 1996.
- [9] P. Martinet, J. Gallice, and D. Khadraoui. Vision based control law using 3d visual features. In *Proc. WAC 96*, volume 3, pages 497–502, Montpellier, France, May 1996.
- [10] N. Papanikolopoulos, P.K. Khosla, and T. Kanade. Vision and control techniques for robotic visual tracking. In *IEEE Int. Conf. on Robotics and Automation*, pages 857–864, Sacramento, USA, April 1991.
- [11] C. Samson, M. Le borgne, and B. Espiau. *Robot control : the task function approach*. Oxford University Press, 1991.
- [12] V. Sundareshwaran, P. Bouthémy, and F. Chaumette. Active camera self-orientation using dynamic image parameters. In *ECCV*, volume 2, pages 111–116, May 1994.
- [13] R. Swain and M. Devy. Visually-guided navigation of a mobile robot in a structures environment. In *5th International Symposium on Intelligent Robotic Systems*, pages 143–152, Stockholm, Sweden, 8-11 July 1997.
- [14] S. Tsuji and J.Y. Zheng. Visual path planning by a mobile robot. In *10th International Joint Conference on Artificial Intelligence*, 1987.
- [15] J. Wang and W.J. Wilson. 3d relative position and orientation estimation using kalman filter. In *IEEE Int. conf. on Robotics and Automation*, volume 3, pages 2638–2645, Nice, France, 10-15 May 1992.