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

Turning around an unknown object using visual servoing

François Berry, Philippe Martinet and Jean Gallice
LASMEA, Université Blaise Pascal, UMR 6602 du CNRS,
63177 Aubière cedex, France
E-mail: berry,martinet,gallice@lasmea.univ-bpclermont.fr

Abstract

In this paper, the problem of controlling a motion by visual servoing around an unknown object is addressed. These works can be interpreted as an initial step towards a perception goal of an unmodeled object. The main purpose is to perform motion with regard to the object in order to discover several viewpoint of the object. The originality of our work is based on the choice and extraction of visual features in accordance with motions to be performed. The notion of invariant feature is introduced to control the navigational task around the unknown object. A real-time experimentation with a complex object is realized and shows the generality of the proposed ideas.

Keywords: *Visual servoing, Hybrid task, Navigation, Unknown object, Task function*

1 Introduction

Over the last few years, there has been increasing interest in object perception based on visual servoing. Some approaches consist in evaluating the structure of the object during navigation. In this way, most techniques are based on the "structure from motion" approach and the use of optical flow. But in these methods, a choice has to be made between the complexity of the scene [3] and time computing [9]. Our purpose is to extract visual features of an unknown object in order to perform a motion around it.

In a previous work [2], an approach to generating a motion around a known object (cube) was presented. This approach was based on a visual servoing technique applied to a time varying reference feature. The reference in the sensor frame was computed according to the desired trajectory in robot workspace. For complex scenes, other works propose an automatic selection of visual features (edge, corner, ...). Papanikolopoulos in [7] used a method based on a SSD optical flow technique. This technique may fail however when the image contains a lot of repeated patterns of the same intensity and is also sensitive to large rotations and small changes in lighting. In [6], the authors propose an approach based on geometric constraints. These are imposed by the feature extraction (type of feature, size, number, ...) and the pose estimation process (field of view, focus, ...). In this strategy, the trajectory should be approximately known in order to perform a good selection of image features. In most cases, an initial learning step is necessary to obtain information characterizing the interaction between the apparatus sensor and the environment (Eigen space method [4], image jacobian estimation [5, 10]). So, the proposed method (developed in [1]) is to perform automatically motions around an unmodeled object.

The first section of this paper presents the choice and the kinematic modeling of the visual features retained to perform the navigational task . The following part describes the control aspect from theoretical basis. These basis are applied to complete motions around an unknown object. Finally, results obtained at video rate with our robotic platform and parallel vision system show the validity of this approach.

2 Modeling

Our idea is to perform a navigational task around an unknown object at a relative given distance, and to center the object in image space during motion. So, relative independent visual features of the object must be defined.

2.1 Visual features

For the centering task, the center ($\underline{X} = (X\ Y)^T$) of the bounding box which frames the object in the image has been chosen (Fig. 1). In order to control the navigational task at a relative distance, an invariant feature has to be defined. To control this distance, a geometric feature varying in function of depth is

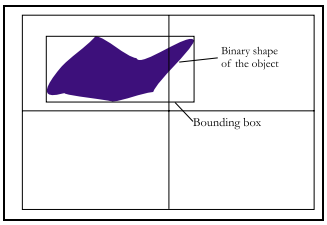


Figure 1: Use of a bounding box in image frame

necessary: this is the case for the projection $\mathcal{L}$ of the shape on an axis Δ (Fig. 2) in image space. This segment $\mathcal{L}$ represents the projection of a segment $\mathcal{S}$ (function of the object) on Δ. From the length of the segment $\mathcal{L}$, it is possible to control the distance between the camera and the object.

The projection axis Δ is centered in the image frame and makes an angle β with the abscissa axis. So, the coordinates of the projected shape on Δ are given by:

$$\begin{aligned} x_{proj} &= (x.\cos\beta + y.\sin\beta).\cos\beta \\ y_{proj} &= (x.\cos\beta + y.\sin\beta).\sin\beta \end{aligned}$$

where (x, y) represent the coordinates of binary shape points and (x_{proj}, y_{proj}) are the coordinates of their projection on Δ. In other words, $(x_{proj},\ y_{proj}) = Proj(x, y)|_{\Delta}$. In addition, the length of the projected

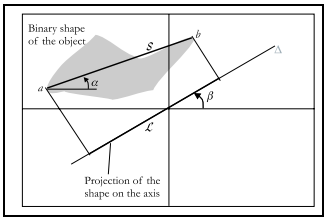


Figure 2: Projection of a binary shape on an axis Δ

segment becomes an invariant feature for a particular choice of the orientation β of the projection axis (see paragraph 3.2).

2.2 Kinematic modeling

For each visual feature $\underline{s}$, it is possible to model their variation $\dot{\underline{s}}$ in function of the camera motion T through the relation:

$$\dot{\underline{s}} = M_{\underline{s}}^T.T$$

where T represents the kinematic screw applied to the sensor and $M_{\underline{s}}^T$ the jacobian matrix (called interaction matrix) relative to the sensor feature $\underline{s}$. In this sequel, the development of each elementary interaction matrix is then presented.

For the centering task, the center of the bounding box is used. So, the interaction can be described by the image jacobian M_M^T of the point M ($\underline{X} = (X\ Y)^T$) (projection of m ($\underline{x} = (x\ y\ z)^T$):

$$\begin{pmatrix} -\frac{1}{z} & 0 & \frac{X}{z} & XY & -1-X^2 & Y \\ 0 & -\frac{1}{z} & \frac{Y}{z} & 1+Y^2 & -XY & -X \end{pmatrix} \quad (1)$$

The second visual feature is the projection $\mathcal{L}$ of a segment $\mathcal{S}$ of the object on one axis Δ. The segment $\mathcal{S}$ can be represented with the vector $\underline{P_{\mathcal{S}}}$, and $\mathcal{L}$ with $\underline{P_{\mathcal{L}}}$ such as:

$$\underline{P_{\mathcal{S}}} = \begin{pmatrix} X_{\mathcal{S}} \\ Y_{\mathcal{S}} \\ L_{\mathcal{S}} \\ \alpha \end{pmatrix} \quad \text{et} \quad \underline{P_{\mathcal{L}}} = \begin{pmatrix} X_{\mathcal{L}} \\ Y_{\mathcal{L}} \\ L_{\mathcal{L}} \end{pmatrix}$$

where $(X_{\mathcal{S}}, Y_{\mathcal{S}})$ (resp. $(X_{\mathcal{L}}, Y_{\mathcal{L}})$) is the middle of $\mathcal{S}$ (resp. $\mathcal{L}$), and $L_{\mathcal{S}}$ (resp. $L_{\mathcal{L}}$) is the corresponding length. The relations between ($\mathcal{S}$) and ($\mathcal{L}$) are easily obtained:

$$\begin{cases} X_{\mathcal{L}} = (X_{\mathcal{S}}.\cos\beta + Y_{\mathcal{S}}.\sin\beta).\cos\beta \\ Y_{\mathcal{L}} = (X_{\mathcal{S}}.\cos\beta + Y_{\mathcal{S}}.\sin\beta).\sin\beta \\ L_{\mathcal{L}} = L_{\mathcal{S}}.cos(\alpha-\beta) \end{cases} \quad (2)$$

The expression of the interaction matrix $M_{\mathcal{S}}^T$ for the segment $\mathcal{S}$ is given in [1]. It is possible to obtain the interaction matrix $M_{\mathcal{L}}^T$ of the projection $\mathcal{L}$ using the following relation:

$$M_{\mathcal{L}}^T = \frac{\partial \underline{P_{\mathcal{L}}}}{\partial \underline{P_{\mathcal{S}}}}.M_{\mathcal{S}}^T$$

where $\frac{\partial P_{\mathcal{L}}}{\partial \underline{P_{\mathcal{S}}}}$ is expressed by:

$$\begin{pmatrix} \cos^2\beta & \sin\beta\cos\beta & 0 & 0 \\ \sin\beta\cos\beta & \sin^2\beta & 0 & 0 \\ 0 & 0 & \cos(\beta-\alpha) & L_{\mathcal{S}}\sin(\beta-\alpha) \end{pmatrix}$$

In the matrix $M_{\mathcal{S}}^T$, only the sub-matrix corresponding to the length of the projection $L_{\mathcal{L}}$ on Δ is considered, and then the related interaction sub-matrix $M_{L_{\mathcal{L}}}^T$ is given by:

$$\begin{aligned} M_{L_{\mathcal{L}}}^T[1,1] &= \nu_1.\cos\beta \\ M_{L_{\mathcal{L}}}^T[1,2] &= \nu_1.\sin\beta \\ M_{L_{\mathcal{L}}}^T[1,3] &= \nu_2.L_{\mathcal{L}} - \nu_1.X_{\mathcal{S}}.\cos\beta - \nu_1.Y_{\mathcal{S}}.\sin\beta \\ M_{L_{\mathcal{L}}}^T[1,4] &= L_{\mathcal{L}}(X_{\mathcal{S}}.\cos\alpha.\sin\alpha + Y_{\mathcal{S}}(1+\sin^2\alpha) + \\ &\quad \tan(\beta-\alpha).(-X_{\mathcal{S}}.\sin^2\alpha + Y_{\mathcal{S}}.\cos\alpha.\sin\alpha)) \\ M_{L_{\mathcal{L}}}^T[1,5] &= -L_{\mathcal{L}}(Y_{\mathcal{S}}.\cos\alpha.\sin\alpha + \tan(\beta-\alpha). \\ &\quad (-X_{\mathcal{S}}.\cos\alpha.\sin\alpha + Y_{\mathcal{S}}.\cos^2\alpha) + X_{\mathcal{S}}(\cos^2\alpha+1)) \\ M_{L_{\mathcal{L}}}^T[1,6] &= -L_{\mathcal{L}}.\tan(\beta-\alpha) \end{aligned} \quad (3)$$

where $\nu_1 = \frac{z_a - z_b}{z_a z_b}$ and $\nu_2 = \frac{z_a + z_b}{2 z_a z_b}$ (z_a and z_b represent the depth of the points a and b in figure 2).

3 Control aspect

In this section, control law and the visual servoing process are developed. First, the fundamental basis concerning *the task function approach* [8] is summarized, then the development in relation to our application is presented.

3.1 The task function approach

The control law used in this study is based on the *task function* formalism. In this approach, the control is directly specified in terms of regulation in the sensor space (image space). For a given robotic task, a *target image* is built, corresponding to the desired position of the end effector with regard to the environment. In general, it can be shown that all servoing schemes may be expressed as the regulation to zero of a function $\underline{f}(\underline{r}, t)$ called *the task function*. So, the use of a vision sensor allows us to build up such a task function used in visual servoing. It is expressed by the relation:

$$\underline{f}(\underline{r}, t) = C[\underline{s}(\underline{r}, t) - \underline{s}^{\star}] \tag{4}$$

where $\underline{s}^{\star}$ is considered as a reference target image to be reached in the image frame, $\underline{s}(\underline{r}, t)$ is the value of visual information currently observed by the camera (it depends on the situation between the end effector of the robot and the scene (noted $\underline{r}$)), and C is a constant matrix, with which it is possible to take into account more visual information than the number of degrees of freedom of the robot, with good conditions of stability and robustness.

The variations of $\underline{f}(\underline{r}, t)$ are given by the following differential relation:

$$\frac{d\underline{f}(\underline{r}(t), t)}{dt} = \frac{\partial \underline{f}}{\partial \underline{r}} \cdot \frac{d\underline{r}}{dt} + \frac{\partial \underline{f}}{\partial \underline{t}} = C \frac{\partial \underline{s}}{\partial \underline{r}} \cdot \frac{d\underline{r}}{dt} + \frac{\partial \underline{f}}{\partial \underline{t}} \tag{5}$$

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 and the term $\frac{\partial \underline{s}}{\partial \underline{r}} = M^T$ called interaction matrix or image jacobian, characterizes the interaction between the sensor and its environment. The concept of interaction matrix is fundamental for modeling systems using an exteroceptive sensor. It allows one to take into account most information required to design and analyze sensor based control schemes.

If the image jacobian is not full rank (n, number of d.o.f > number of independent visual features), it is possible to use an hybrid task. In an hybrid task, the primary task $\underline{e}_1$ maintains a visual constraint during the trajectory, while the secondary task $\underline{e}_2$ can be seen as representing a minimization of a secondary cost h_s with the gradient $\underline{g}_s^T = (\frac{\partial h_s}{\partial \underline{r}})^T$. A global *task function* $\underline{e}$ takes the form:

$$\underline{e} = W^+ \underline{e}_1 + \gamma . (\mathbb{I}_n - W^+ W) \underline{g}_s^T \tag{6}$$

where W^+ and $(\mathbb{I}_n - W^+ W)$ are two projection operators which guarantee that the camera motions due to the secondary task are compatible with the regulation of $\underline{s}$ to $\underline{s}^*$. W is a full rank matrix such as $Ker(W) = Ker(M^T)$. The parameter γ is used to tune the preponderance between the primary and the secondary task. Considering an exponential decay of $\underline{e}(\underline{r}, t)$:

$$\dot{\underline{e}}(\underline{r}, t) = -\lambda \underline{e}(\underline{r}, t) \tag{7}$$

with λ a positive scalar constant and in applying relation 5 to the global task function $\underline{e}$, the kinematic screw can be expressed with:

$$T = -\left(\frac{\partial \underline{e}}{\partial \underline{r}}\right)^{-1} \left(\lambda \underline{e} + \frac{\partial \underline{e}}{\partial t}\right) \tag{8}$$

To ensure the stability of the system, the following condition

$$\left(\frac{\partial \underline{e}}{\partial \underline{r}}\right) \cdot \left(\widehat{\frac{\partial \underline{e}}{\partial \underline{r}}}\right)^{-1} > 0 \tag{9}$$

must be verified [8]. This is done when the combination matrix C is fixed to $W.M^{T+}$. In addition, the previous condition is always verified when choosing $\left(\frac{\partial \underline{e}}{\partial \underline{r}}\right)^{-1} = \mathbb{I}_6$.

Considering a motionless environment, it gives $\frac{\partial \underline{s}}{\partial \underline{t}} = 0$ and $\frac{\partial \underline{e}_1}{\partial \underline{t}} = 0$. Finally, from the relations 6 and 8, the control law has the following expression:

$$T = -\lambda \underline{e}(\underline{r}, t) - \gamma (\mathbb{I}_n - W^+ W) \frac{\partial \underline{g}_s^T}{\partial t} \tag{10}$$

3.2 Moving around an unknown object

In this section, the general control law (Eq.10) is adapted in order to move around an unknown object. So, it is necessary to use an hybrid task composed of:

- a primary task, where the goal is to gaze at the object, to center it in the sensor frame and to hold a constant distance between the camera and the object.

- a secondary task which generates the translation along the X and Y axis .

As a result the visual feature is modeled by:

$$\underline{s}(\underline{r},t) = \begin{pmatrix} X \\ Y \\ L_{\mathcal{L}} \end{pmatrix}$$

With such features, only 3 d.o.f can be controlled (i.e. R_x, R_y, T_z), and 2 d.o.f are needed for the navigational task (i.e. T_x, T_y). So, only the reduced system to these d.o.f is considered, and then the interaction matrix associated to $\underline{s}^*(\underline{r},t)$ becomes:

$$M^T_{|\underline{s}=\underline{s}^*} = \begin{pmatrix} -\frac{1}{z} & 0 & 0 & 0 & -1 \\ 0 & -\frac{1}{z} & 0 & 1 & 0 \\ \nu_1^*.\cos\beta & \nu_1^*.\sin\beta & \nu_2^*.L_{\mathcal{L}}^* & 0 & 0 \end{pmatrix}$$

From the kernel of $M^T_{|\underline{s}=\underline{s}^*}$, the motions allowed by the interaction can be given by:

$$Ker(M^T_{|\underline{s}=\underline{s}^*}) = \left\{ \begin{pmatrix} 1 & 0 & \frac{-\nu_1^*.\cos\beta}{\nu_2^*.L_{\mathcal{L}}^*} & 0 & -\frac{1}{z} \\ 0 & 1 & \frac{-\nu_1^*.\sin\beta}{\nu_2^*.L_{\mathcal{L}}^*} & \frac{1}{z} & 0 \end{pmatrix} \right\},$$

Considering a motion around the object such as $\underline{T} = \begin{pmatrix} A.\cos\theta \\ A.\sin\theta \end{pmatrix}$, the general form of the allowed camera motion is:

$$T = \begin{pmatrix} T_x = A.\cos\theta \\ T_y = A.\sin\theta \\ T_z = A.\frac{-\nu_1^*.(\cos\beta.\cos\theta+\sin\beta.\sin\theta)}{\nu_2^*.L_{\mathcal{L}}^*} \\ R_x = A.\frac{\sin\theta}{z} \\ R_y = -A.\frac{\sin\theta}{z} \end{pmatrix}$$

This motion is composed of a combination of translation and rotation along the x and y axis. However, the translation along the optical axis (T_z) is not null. In other words, the decoupling of T_z is only done when $\cos\beta.\cos\theta+\sin\beta.\sin\theta = 0$. This condition is obtained for $\theta = \beta + \frac{\pi}{2}$, so the orientation of the axis Δ must be orthogonal to the motions around the object (projected in the image space) (Fig. 3). The task function can be written like:

$$\underline{e} = M^{T+}_{|\underline{s}=\underline{s}^*}(\underline{s}-\underline{s}^*) + \gamma(\mathbb{I}_5 - M^{T+}_{|\underline{s}=\underline{s}^*} M^T_{|\underline{s}=\underline{s}^*})\underline{g}_s^T$$

where $(\mathbb{I}_5 - M^{T+}_{|\underline{s}=\underline{s}^*} M^T_{|\underline{s}=\underline{s}^*})$ is an orthogonal projector. Then, the control law is given by:

$$T = -\lambda\underline{e} - \gamma(\mathbb{I}_5 - M^{T+}_{|\underline{s}=\underline{s}^*}.M^T_{|\underline{s}=\underline{s}^*})\frac{\partial \underline{g}_s^T}{\partial t} \qquad (11)$$

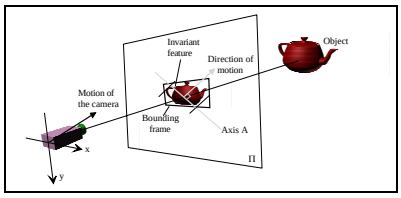


Figure 3: Orientation of axis Δ in comparison to the camera motions.

In our case, the secondary cost function h_s is defined by:

$$h_s = \frac{1}{2}(x - x_o - V_x t)^2 + \frac{1}{2}(y - y_o - V_y t)^2$$

where (x, y) represents the position of the camera, (x_0, y_0) is the initial position (in our case $(x_0, y_0) = (0, 0)$) and (V_x, V_y) is the velocity of the camera used for the navigation. In other words, the velocity (V_x, V_y) allows one to describe the motion around the object. For example a vertical motion on top at $0.1m.s^{-1}$ is achieved for $V_x = 0$ and $V_y = -0.1m.s^{-1}$. The gradient of this cost function is given by $\underline{g}_s^T$ is:

$$\underline{g}_s^T = \begin{pmatrix} (x - x_o - V_x t) \\ (y - y_o - V_y t) \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

4 Results

4.1 Experimental context

Our experimental cell is composed of a cartesian robot with 6 d.o.f. A CCD camera is embedded on the end effector and is connected to the vision parallel architecture named *Windis*. In function of the motion to be performed, both vision and control processes have to be adapted. In the vision process, this is the visual features extraction, and in the control process this is the sensor vector and the corresponding interaction matrix.

To reduce the execution time, an area of interest is defined. From this area of interest, a bounding box which frames the unknown object is built, and the coordinates of the center of this box are computed. For a particular navigational task, an invariant feature in the direction of the motion has to be built (section

2.1). First, knowing the direction of the motion an axis of projection Δ is defined in image space as orthogonal to this direction of motion. Second, the segment is obtained by projection of the shape on this axis and the length is computed (Fig. 4). Finally, as

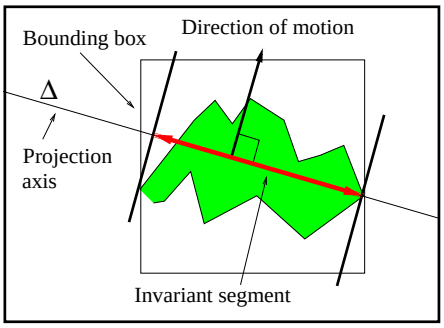


Figure 4: Invariant feature construction

a navigational task imposes movement around an object, the vision process can be affected by the different lighting conditions encountered during this movement. So, it is necessary to adapt some characteristics. Particularly, the low level extraction of the bounding box needs to adapt the thresholds according to the lighting conditions. So, at each iteration an histogram is computed and all thresholds are modified. All of this implementation is made at video rate (40 ms).

4.2 Experimental results

For the navigational task, two motions are performed around a little rubber giraffe. Considering the object centered at a given distance (Height of object= 30% of image size (170 pixels)), the camera moves to the left side while keeping the object centered and then rises above the object. (Fig. 5). For the con-

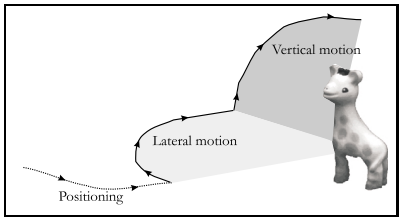


Figure 5: *Trajectory around the giraffe.*

trol law, the parameters are $\lambda = 0.8$, $\gamma = 1.0$ and $z^* = 0.7m$. These parameters are tuned experimentally in accordance with the task to perform. The velocities applied to the effector are $T_x = -0.08m.s^{-1}$ and $T_y = -0.08m.s^{-1}$. The rotation axis (in image) is respectively the height and the width of the bounding frame. The visual reference feature is chosen from the last measure during the previous motion. Such choice allows one to keep the same distance for both motions. Figure 6 represents the evolution of the ob-

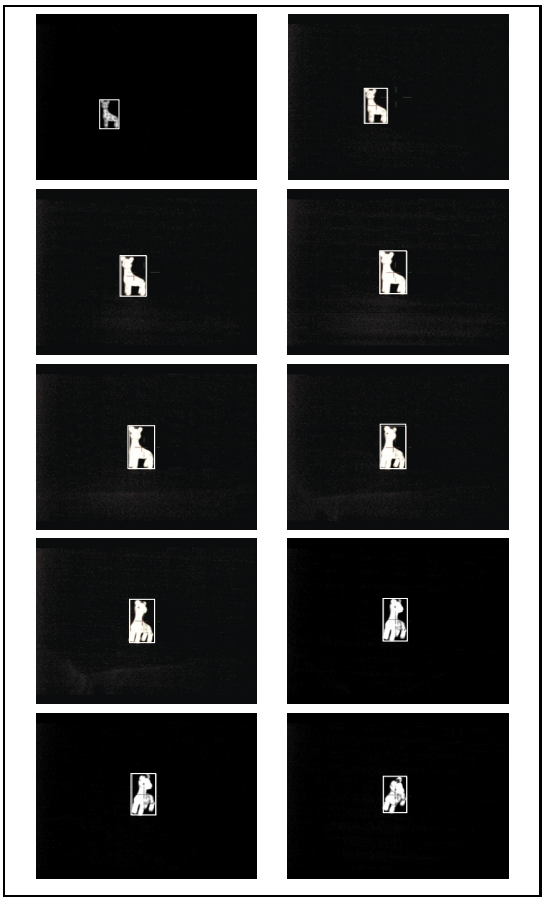
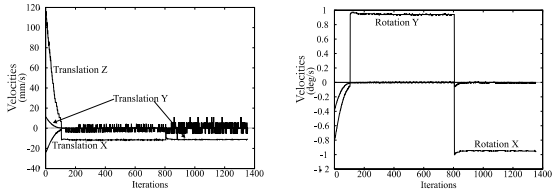


Figure 6: *Evolution of the object during navigation.*

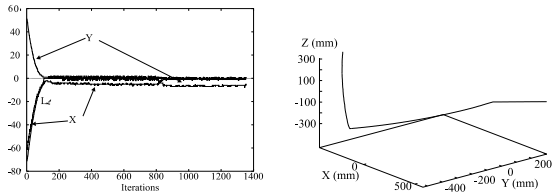
ject during the servoing and figure 7 presents the velocity of the kinematic screw. The servoing task is composed of three steps. The first step concerns the positioning task, second and third steps - the navigational tasks. Velocities become noisy during navigational task and particularly noisier during the third step. In the latter, the width of the object is used instead of the height (used in the second step), and in our implementation, the discretization in x and y direction are not the same: in vertical direction, only one frame is used. The image is "compressed" in this direction and there is a filtering effect along this direction. The motions around the object are performed



(a) (b)

Figure 7: *Translation and rotation velocities*

after the positioning task (Fig. 8.b). The choice of the desired distance z^* between the camera and the object is very important and determines the achievement of the secondary task. Though this distance cannot be measured and is set arbitrarily, thus the residual error (Fig. 8.a) is mainly due to this estimation.



(a) (b)

Figure 8: *(a)Error on features. (b) 3D Trajectory of the effector.*

5 Conclusion

Many studies in visual servoing concern known objects. The present study adresses the problem of "how to move" in respect to an unknown object. One application is the first step towards a recognition process where it is necessary to perform known motion around the object. The proposed method is based on the visual servoing techniques and is particularly robust. A study of the different interaction relations for the visual features has shown the allowed motions around an object and the conditions of good achievement. All experiments have been successfully implemented on our robotic platform and have shown the validity of such approach.

References

[1] F. Berry, P. Martinet and J. Gallice. Real time visual servoing around a complex object. In *IEICE Trans. on Informations and Systems, Special Issue on Machine Vision Applications*, To be published.

[2] F. Berry, Martinet P. and Gallice J. Visual feedback in camera motion generation: Experimental results. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 1, pages 513–518, Kyongju, Korea, 17-21 October 1999.

[3] F. Chaumette, Boukir S., Bouthemy P. and Juvin D. Structure from controlled motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(5):492–504, May 1996.

[4] K. Deguchi and Takhashi I. Image based simultaneous control of robot and target object motions by direct image interpretation method. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 1, pages 375–380, Kyongju, Korea, 17-21 October 1999.

[5] M. Jägersand, O. Fuentes and R. Nelson. Experimental evaluation of uncalibrated visual servoing for precision manipulation. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol.3, pages 2874–2880, Albuquerque, USA, 1997.

[6] F. Janabi-Sharifi and Wilson W.J. Automatic selection of image features for visual servoing. *IEEE Journal of Robotics and Automation*, 5(3):404–417, october 1997.

[7] N.P. Papanikolopoulos. Selection of features and evaluation of visual measurements during robotic visual servoing tasks. *Journal of Intelligent and robotic systems*, (13):279–304, 1995.

[8] C. Samson, Le Borgne M. and Espiau B. *Robot Control. The task function approach.* ISBN 0-19-8538057. Clarendon Press, Oxford, 1991.

[9] G. P. Stein and Shashua A. Direct methods for estimation of structure and motion from three views. Technical report, Report of M.I.T No.1594, November 1996.

[10] H. I. Suh. *Visual servoing of robot manipulators by fuzzy menbership function based neural networks*, vol. 7 of Robotics and Automated Systems, pages 285–315. World Scientific Publishing, 1993.