Real time visual servoing around a complex object

F. Berry, P. Martinet, J. Gallice
Laboratoire des Sciences et Matériaux pour l'Electronique et dAutomatique
Université Blaise Pascal de Clermont Ferrand
UMR 6602 du C.N.R.S, F-63177 Aubière Cedex, France
E-Mail: berry, martinet, gallice@lasmea.univ-bpclermont.fr

Keywords: *Active perception, Visual servoing, Complex object, Real time, Parallel architecture.*

Abstract

In this paper, we propose a method to perform a motion by visual servoing around an unknown object. The approach developped in this article, can be interpreted as an initial step for a perception goal of an unmodeled object. The originality of our work is based on the building of invariant visual features from the motion to perform. During the experimentations, we use a cartesian robot connected to a real time vision system. A CCD camera is mounted on the end effector of the robot. The experimental results present a linkage of trajectories around small plastic toys.

1 introduction

Most works in vision for robotic deal with the perception of spatial geometry of scenes from a mobile camera. These systems come within the framework of active vision. But the perception of an object can be done at different scales. The highest level of perception is the recognition of the object and the relation between the perceiver and the action to perform on the object [5, 10]. Another approach consists in evaluating the structure of the object during the navigation. In this way, most techniques are based on the "structure from motion" approach and the use of optical flow. But in these methods, we must choose between the complexity of the scene [3] and the time computing [9]. Our goal is to perceive an unknown object in order to perform a motion around it. In this case, we consider that a coarse knowledge of the object is required. So, the perception consists in evaluating the pose and the space filled by the object. From that point of view, the main idea is to approximate the object with a bounding box. This approximation allows us to simplify the motion around the object.

In a previous work [2], we have proposed an approach to generate a motion around a known object (cube). This approach is based on visual servoing technique applied to a time reference feature. We compute the reference in the sensor frame according to the desired trajectory in robot workspace.

In the case of complex scenes, others works propose an automatic selection of visual features (edge, corner, ...). Papanikolopoulos in [6] uses a method based on a SSD optical flow technique. This technique may fail 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 [4], the author proposes an approach based on geometric constraints. These latters are imposed by the feature extraction (types of features, size, number, ...) and the pose estimation process (field of view, focus, ...). But in this strategy, the trajectory should be approximately known to perform a good selection of image features.

In this paper, we use a visual servoing loop [1, 11] to perform different motions around an unmodeled object. This work is a first step in our global approach of object perception and should allow us to build the bounding box.

To reduce image processing, we define an area of interest (bounding frame) which frames the unknown object. We build invariant features to ensure the visual servoing task.

2 Control law

The control law used in this work is based on the task function formalism. In this approach, the control is directly specified in terms of regulation in the image. It may be noted that this approach has the advantage of avoiding the intermediate step of the 3D estimation of the target with regard to the end effector. For a given robotics task, a *target image* is constructed, corresponding to the desired position of the end effector with regard to the environment. It can be shown that all servoing schemes may in general be expressed as the regulation to zero of a function $\underline{e}(\underline{r},t)$ called *the task function* [15]. 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:

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

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. This information depends on the situation between the end effector of the robot and the scene (noted $\underline{r}$).
- 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.

In our case, we consider a motionless environment so we have $\frac{\partial s}{\partial \underline{t}} = 0$ and the variations of $\underline{s}(\underline{r},t)$ are given by the following differential relation:

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

where $\frac{dr}{dt}$=T=($\overrightarrow{V}$, $\overrightarrow{\Omega}$) is the kinematic screw.

T represents the relative velocity between the camera and its environment. The term $\frac{\partial e}{\partial \underline{r}} = C\frac{\partial s}{\partial \underline{r}}$, where $\frac{\partial s}{\partial \underline{r}} = L^T$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 exteroceptive sensor. It allows to take into account most information required to design and analyze sensor based control schemes.

If the image jacobian is not full rank (number of d.o.f > number of independant visual features), it is possible to use an hybrid task.

In an hybrid task, the primary task $\underline{e}_1$ allows to maintain a visual constraint during the trajectory, while the secondary task $\underline{e}_2$ can be seen as representing the minimization of a secondary cost h_s with the gradient $g_s = (\frac{\partial h_s}{\partial \underline{r}})^T$.

We obtain a global *task function* $\underline{e}$, which takes the form:

$$\underline{e} = W^+\underline{e}_1 + (I_6 - W^+W)g_s^T \tag{3}$$

where: W^+ and $(I_6 - W^+W)$ are two projection operators which guarantee that the camera motion due to the secondary task is compatible with the regulation of $\underline{s}$ to $\underline{s}^*$. W is a full rank matrix such as $Ker(W) = Ker(L^T_{|\underline{s}=\underline{s}^*})$

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

$$\dot{\underline{e}}(\underline{r},t) = -\lambda\underline{e}(\underline{r},t) \tag{4}$$

(λ is a positive scalar constant)
and from the relation 2, we get:

$$T = -\lambda\underline{e} - \frac{\partial \underline{e}}{\partial t} \tag{5}$$

and from the relation, 3 and 5, we deduce the following control law:

$$T = -\lambda\underline{e}(\underline{r},t) - \alpha(I_6 - W^+W)\frac{\partial g_s^T}{\partial t} \tag{6}$$

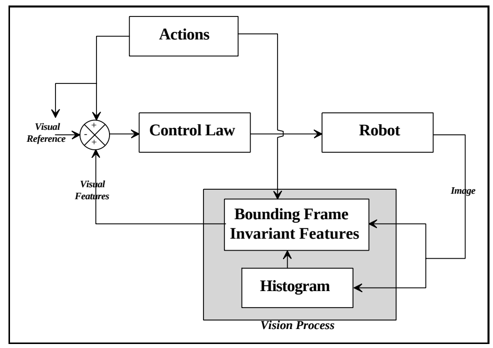

Figure 1: Global scheme of our application

We use this control law, to perform the both goal of our work:

- The first goal of the control law is to gaze the object and to center it in the sensor frame. The translation along the Z axis and the rotation around X and Y axis achieve it. The features used for this task, are the position and the size of the bounding frame in the image.
- Simultaneously, the action (motion around the object) is generated by the translation along X and Y axis and is managed by the second part of the control law. This task is directly controlled by invariant segments in the sensor frame. These invariant features are computed in real time flow processing, from the selected pixels and the action to achieve.

3 Choice of the invariant features

The experimentation is performed with constrasted images, where the background is dark and the object is easy to detect. The pixels of the object is selected with an adaptative threshold computed from a gray scale histogram. The center of the object in image, allows to gaze the object while the projection of the selected pixels along a suitable axis

A allows to maintain a distance between the camera and the object during the motion.

The figure 2 presents the building of this invariant features. We consider that the motion is locally belong in a plane Π. This plane is orthogonal to the optical axis. So an invariant feature can be built with a segment which is orthogonal to the direction of the motion. This segment is along the axis A.

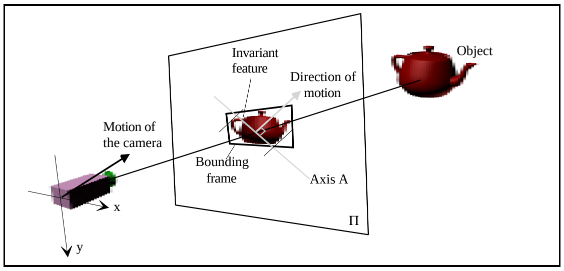

Figure 2: Building of the invariant features

Thus, the primary task in the control law uses two types of visual features: the center of the object in image to gaze the object and the lenght of the segment to maintain the distance camera/object. Also, the primary task controls the translation axis along the optical axis (OZ) and the rotation axis around X and Y. The secondary task performs the motion around the object in using the translation axis along OX and OY. The motion of the camera is easy to specify in the camera frame. For example, a motion on the right is equivalent to a velocity along OX.

4 Experimental results

4.1 Implementation

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 Windis [7, 8]. Windis architecture includes three basic modules.

Figure 3: Robotic platform

WINDIS Window Distributor Subsystem is used for window extraction, the execution of low level processing and the distribution of active windows toward the Window Processing Subsystem. On this board, we extract the grey levels and a list of selected pixels corresponding to the highest gradient.

On WINPROC Window Processing Subsystem, we associated one to sixteen DSP 96002 modules with one distributor module. DSP modules are put together on mother boards and execute medium level processing on windows. From the selected pixels, we extract in real time the bounding frame of the object in the image on a DSP. On another DSP, the histogram of grey levels is computed, and provides an adaptative threshold. This threshold allows to set the level of selected pixels for a robust extraction of the bounding frame and the invariant features.

WINMAN Window Manager Subsystem controls distributor and DSP modules, and executes high level processing of application tasks. Moreover, it is used for the tracking of the active windows throughout the sequence and for the command of the robot. A 68040 based cpu board implements this module.

The management of the system is ensured under VxWorks Real Time Operating System.

4.2 Results

On the figure 4, we show 2 complex objects used for the experimentations.

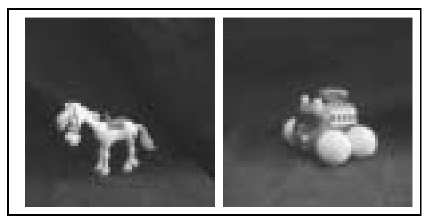

Figure 4: Type of objects used for the experimentations

For these results, we have chosen the "car toy" (right photo) and the trajectory is composed of two motion (cf. Fig. 5)

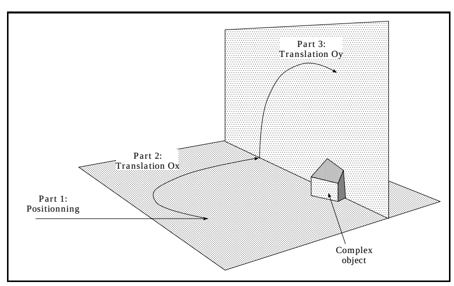

Figure 5: Overview of the experimentations

In this experimentation, we perform a linkage of trajectories around the toy. The trajectory is composed by three parts. At first, we perform a positionning task so as to center the object in the image and so as to set the width to 25% of the image. In a second part, we perform a translation along the Ox axis, with a velocity such V_x=-0.02 m.s^{-}1, and the last part is a translation such V_y=-0.02 m.s^{-}1.

The Graph. 6 and 7 show the translation and the rotation velocities. We can remark, that the rotation velocity is perfectly complementary to translation velocity. This fact does not disturb the primary task and consequently ensures a perfect gaze control. We note it, on the graph 8 where the errors are perfectly regulated to zero.

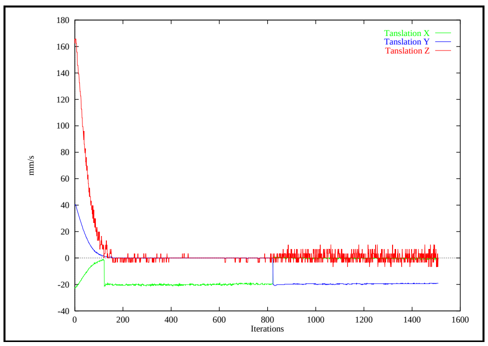

Figure 6: Translation velocities

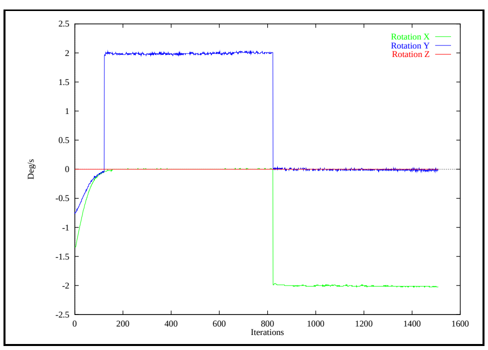

Figure 7: Rotation velocities

5 Conclusion

We have presented a work that can be considered as a first step for the techniques of perception of unknown objects. This method is based on the visual servoing techniques and is particularly robust. This first real time implementation shows the useful of our approach and actually we work on the next steps of a strategy of perception.

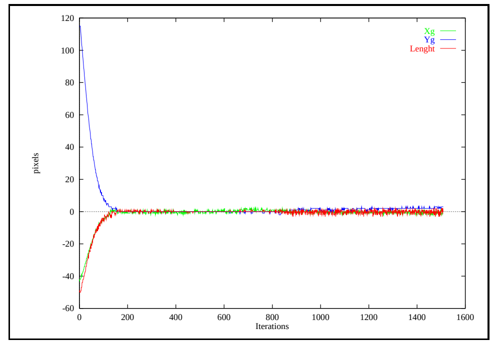

Figure 8: Errors on visual features

References

[1] Espiau B., Chaumette F., and Rives P. A new approach to visual servoing in robotics. In *ICRA92*, volume 8, pages 313–326, Paris, june 1992. IEEE Trans. on Robotics and Automation.

[2] Berry F., P. Martinet, and J. Gallice. Trajectory generation by visual servoing. IEEE Int. Conference on Intelligent Robots and Systems, September 1997.

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

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

[5] Aloimonos J and Bandyopadhyay A. Active vision. In *ICCV87*. Proc. First Int. Conf. Computer Vision, 1987.

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

[7] Martinet P., P. Rives, P. Fickinger, and J.J. Borelly. Parallel architecture for visual servoing applications. Workshop on Computer Architecture for Machine Perception, CAMP'91, December 1991.

[8] Rives P., J.J. Borelly, J. Gallice, and P. Martinet. Parallel architecture for visual servoing applications. Workshop on Computer Architecture for Machine Perception, CAMP'93, December 1993.

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

[10] Bajcsy R. Active perception. In *Proc. IEEE* **76**, pages 996–1005, August 1988.

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