# Use of first derivative of geometric features in Visual Servoing

P. Martinet F. Berry 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

*We have been interested in a visual servoing system for several years. Visual servoing is based on an array of measurements taken from a set of images and used each time as an error function to compute a control vector. This is applied to the system (robot and camera) and makes it move in order to reach a desired situation, at the end of the task, directly depicted in the image.*

*The originality of this recent work consists in improving the visual servoing approach. To do this, we consider a signal sensor vector constructed of a geometrical feature (such $(x, y)$ point coordinates, line parameters, etc) and its first derivative. In this paper, we show how to work out the interaction matrix. We have tested this new approach on workstation in the case of the point feature. We have also implemented it on our robotic platform. The overall results show a great improvement due to the action of this new signal sensor. We are extending this approach to more complex features.*

## 1 Introduction

The use of visual information in robot control has been approached by several techniques. Many articles propose tracking algorithms based on the detection of moving objects in image space, and make a path planning with a Kalman filter [1][12][13]. However, work dealing with the integration of visual information in a robot control loop shows two possible approaches.

The first approach, "Look and Move", has something in common with the articles quoted above. This is a technique where the position of the robot is estimated with the data of the vision system. From this estimation, the control to be applied to the robot end effector is calculated to complete the task. In this case, it is essential to have a model of the robot and to compute at each iteration the situation between the robot and the target. However, even if this sort of visual servoing has been preferred for a long time because of its simplicity of adjustment, various kinds of errors cause interference [2][4]. First of all, there are errors in the visual extraction due to the visual sensor itself. Also, errors in the modelling of the robot arise from the difficulty of considering all the physical and electrical characteristics of the robotic system. Finally, there are some errors in estimation of the situation, arising from lack of precision in the calibration of the camera, and some interpretation errors.

These drawbacks are eliminated by the second approach, "Visual Servoing", recently introduced [4][7][15], which controls the robot end effector by introducing the visual information directly into the control loop. The control law is expressed in the sensor frame and does not need any interpretation step. In this article, we present a control law where the signal sensor has been improved with its first derivatives. The experimental results show the advantages of this approach, such as the number of axes that can be controlled and the robustness of the control law.

## 2 Visual servoing

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 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. Then, a robotics control scheme is developed directly based on disparities between current and target image [4][6].

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.

For a given task, the problem consists, first of all, in choosing relevant visual information to achieve the task, and then in constructing the constant matrix $C$. The visual information provided by the wrist-mounted sensory apparatus is modelled as a set of *elementary signals* $\underline{s}$ associated with the 2D geometric primitives in the image corresponding to the projection of the 3D primitives in the scene. The interaction between sensor and scene is described (in the matrix form) by

$$\frac{\partial \underline{s}}{\partial \underline{r}} = L_{\underline{s}}^T \tag{2}$$

called the interaction matrix which links the interaction between the robot and its environment. In the case of visual servoing, we obtain (see [4] for example) the relation

$$\dot{\underline{s}} = L_{\underline{s}}^T T \tag{3}$$

where

- $\dot{\underline{s}}$ is the time variation of $\underline{s}$.
- $T$ is the velocity screw of the camera $T = (\underline{V}, \underline{\Omega})^T$, three translations and three rotations.
- $L_{\underline{s}}^T$ the *interaction matrix*.

By combining (1) and (3), and considering an exponential convergence (for a task positioning) of all components of the task function (i.e. $\dot{e} = -\lambda e$, where $\lambda$ is a positive scalar constant), we obtain the following expression of the control law:

$$T = -\lambda L_{\underline{s}}^{T+}(\underline{s} - \underline{s}^*) \tag{4}$$

where $L_{\underline{s}}^{T+}$ represents the pseudo-inverse matrix of $L_{\underline{s}}^T$.

Figure 1 shows the visual servoing scheme.

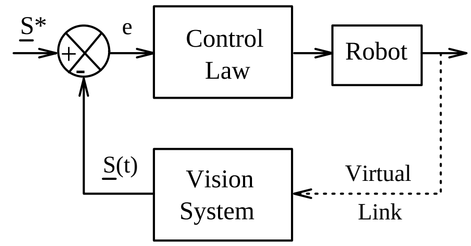


Fig. 1 Visual servoing scheme.

## 3 The visual signal used

In all the work done on visual servoing, only a geometric type of signal sensor has been used [4][11][17]. Even if some articles, such as [5][3], show that improvements can be obtained in visual servoing by combining the control loop with different types of tools such as a Kalman filter or the generalized likelihood ratio test, the signal sensor used is still purely geometric. However, we note a recently developed approach in [16] using 2D affine motion parameters, in order to execute camera autosetting with regard to the target.

Our approach consists in improving the signal sensor by using, not only a geometric feature (point, line,etc), but also the first derivative of this feature. By using such signals, we increase the dimension of the vector $S$, which enables more axes to be controlled. We write the new sensor signal like this:

$$\underline{S} = (\underline{s}, \dot{\underline{s}})^T \tag{5}$$

## 4 The interaction matrix

From the derivation of equation (5), we obtained:

$$\dot{\underline{S}} = (\dot{\underline{s}}, \ddot{\underline{s}})^T \tag{6}$$

With equation (3), we can express the second derivative of $\underline{s}$ with the following expression:

$$\ddot{\underline{s}} = \dot{L}_{\underline{s}}^T T + L_{\underline{s}}^T \dot{T} \tag{7}$$

In the case of task positioning (which concerns us in this paper), it is reasonable to consider an exponential convergence of the kinematic screw $T$ of the form:

$$\dot{T} = -\alpha T \tag{8}$$

where $\alpha$ is a diagonal matrix enabling the speed of decay for each axis to be controlled:

$$\alpha = \begin{pmatrix} \alpha_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha_6 \end{pmatrix} \tag{9}$$

Under these conditions and using relation (3), it is possible to rewrite (6) in the following form:

$$\dot{\underline{S}} = \begin{pmatrix} L_{\underline{s}}^T \\ \dot{L}_{\underline{s}}^T - L_{\underline{s}}^T \alpha \end{pmatrix} T = \mathcal{L}_{\underline{S}}^T T \tag{10}$$

where we have defined the new interaction matrix $\mathcal{L}_{\underline{S}}^T$ as:

$$\mathcal{L}_{\underline{S}}^T = \begin{pmatrix} L_{\underline{s}}^T \\ \dot{L}_{\underline{s}}^T - L_{\underline{s}}^T \alpha \end{pmatrix} \tag{11}$$

$\mathcal{L}_{\underline{S}}^T$ can be obtained even if only $L_{\underline{s}}^T$ is known.

## 5 The Case of the point feature

In the simple case of the point feature $\underline{x} = (x, y, z)$ in 3D space, the sensor signal is given by $\underline{S} = (\underline{X}, \dot{\underline{X}})$ where $\underline{X}$ and $\dot{\underline{X}}$ represent respectively the $(X, Y)$ position of a point and $(\dot{X}, \dot{Y})$ its first derivative in the image frame.

$\dot{\underline{X}}$ can be evaluated from the measurement of the gap position between two successive samples. In the case of more complex features, the derivative signal $\dot{\underline{s}}$ can be evaluated by measuring the "optical flow".

In this paper, this is the reason why we talk indifferently of "optical flow" or "first derivative".

In the case of a point feature, we note that [4] has shown that the interaction matrix $L_{\underline{s}}^T$ is expressed by the relation:

$$L_{\underline{s}}^T = \begin{bmatrix} -\frac{1}{z} & 0 & \frac{X}{z} & XY & -(1+X^2) & Y \\ 0 & -\frac{1}{z} & \frac{Y}{z} & (1+Y^2) & -XY & -X \end{bmatrix} \tag{12}$$

With relation (11), we deduce $\mathcal{L}_{\underline{S}}^T$:

$$\mathcal{L}_{\underline{S}}^T = \begin{bmatrix} -\frac{1}{z} & 0 & \frac{X}{z} & XY & -(1+X^2) & Y \\ 0 & -\frac{1}{z} & \frac{Y}{z} & (1+Y^2) & -XY & -X \\ \frac{\alpha_1}{z} + \frac{\dot{z}}{z^2} & 0 & \frac{\dot{X}z - X\dot{z}}{z^2} - \frac{\alpha_3 X}{z} & \dot{X}Y + X\dot{Y} - \alpha_4 XY & -2\dot{X}X + \alpha_5(1+X^2) & \dot{Y} - \alpha_6 Y \\ 0 & \frac{\alpha_2}{z} + \frac{\dot{z}}{z^2} & \frac{\dot{Y}z - Y\dot{z}}{z^2} - \frac{\alpha_3 Y}{z} & 2\dot{Y}Y - \alpha_4(1+Y^2) & \alpha_5 XY - \dot{X}Y - X\dot{Y} & \alpha_6 X - \dot{X} \end{bmatrix} \tag{13}$$

We note that it has been shown experimentally[4][9][11] that we can use an interaction matrix calculated only for the equilibrium situation. We note this matrix $\mathcal{L}_{\underline{S}=\underline{S}^*}^T$. The use of this interaction matrix allows us to decrease the computing time and to avoid any singularities during the control process.

By intuition, if the observed point is located on the optical axis $Oz$ in an equilibrium situation, it is obvious that any translation or rotation along or around this axis is without effect on the features measured in image space. In these conditions, the dimension of the kinematic screw can be limited to 4 instead to 6. The interaction matrix is then square ($4\times4$) and it is possible to compute its inverse, noted $\mathcal{L}_{\underline{S}=\underline{S}^*}^{-1}$. In the case of more complex features, we use the pseudo-inverse matrix instead of the inverse.

From relation (13), it is easy to show that at the equilibrium situation ($\underline{X}^* = 0; \dot{\underline{X}}^* = 0; \underline{z}^* = 0$), the third and the sixth columns of $\mathcal{L}_{\underline{S}=\underline{S}^*}^T$ are equal to zero, so we obtain:

$$\mathcal{L}_{\underline{S}=\underline{S}^*}^T = \begin{bmatrix} -\frac{1}{z^*} & 0 & 0 & -1 \\ 0 & -\frac{1}{z^*} & 1 & 0 \\ \frac{\alpha_1}{z^*} & 0 & 0 & \alpha_5 \\ 0 & \frac{\alpha_2}{z^*} & -\alpha_4 & 0 \end{bmatrix} \tag{14}$$

To facilitate parameter choice, we can assume the same parameter for the translations along $x$ and $y$ ($\alpha = \alpha1 = \alpha2$), and the same one for the rotations around $x$ and $y$ ($\beta = \alpha4 = \alpha5$). The matrix becomes:

$$\mathcal{L}^T_{\underline{S}=\underline{S}^*} = \begin{bmatrix} -\frac{1}{z^*} & 0 & 0 & -1 \\ 0 & -\frac{1}{z^*} & 1 & 0 \\ \frac{\alpha}{z^*} & 0 & 0 & \beta \\ 0 & \frac{\alpha}{z^*} & -\beta & 0 \end{bmatrix} \tag{15}$$

Then, the inverse of $\mathcal{L}^T_{\underline{S}=\underline{S}^*}$ is:

$$(\mathcal{L}^T_{\underline{S}=\underline{S}^*})^{-1} = \frac{1}{\alpha-\beta} \begin{bmatrix} \beta z^* & 0 & z^* & 0 \\ 0 & \beta z^* & 0 & z^* \\ 0 & \alpha & 0 & 1 \\ -\alpha & 0 & -1 & 0 \end{bmatrix} \tag{16}$$

And, the final control law is:

$$\boxed{Tc = -\lambda(\mathcal{L}^T_{\underline{S}=\underline{S}^*})^{-1}(\underline{S} - \underline{S}^*)}$$

When $\alpha$ is larger with regard to $\beta$, we obtain the same expression of the control law as in [4] for the point feature concerning the use of rotation velocities. In this case, $\dot{s}$ has little influence in the general control law behaviour. The closer $\alpha$ is to $\beta$, the more the derivative aspect of the control is preponderant. A global stability approach to the control law is being achieved, and will enable the different gains used to be determined.

# 6 Results

In this section, the control law described above is illustrated through simulation and experimental results.

## 6.1 Simulation results

To evaluate the performance of the control law, a simulator was written in C language on a HP-UX Workstation (HP 9000/735). In this simulator, the model of the camera is taken into account, but we do not consider the dynamic characteristics of the robot.

For all the tests, we consider an illuminated point in the image plane $(X, Y)$. Its initial position is $(225, 223)$ and $(0, 0)$ represents the origin of the image frame. The task is correctly performed when the final location of the point merges into the origin of the image frame. To show the advantages of the new control law, we tested it by introducing a pipeline delay of 160 ms in the control scheme. For example, this delay can be interpreted as the accumulation time of the following delays:

1. delay in image storing at video rate (40ms).
2. delay due to optical flow processing. In the case of more complex scenes, processing of $\dot{s}$ can be obtained with an optical flow measure. Typically, P.K. Allen [1] has implemented the Horn et Shunck algorithm [8] in real time video with a delay of 120 ms.

This pipeline delay can be assimilated as a pure delay between action and reaction of the robot. It does not affect the sampling period used in the control scheme. Figure 2 represents the evolution of visual information (position $(x, y)$ in the image plane) with and without optical flow in the control law. In the figures, one iteration corresponds to 40 ms.

Without optical flow, the control law behaves as a second order system with exponential damped oscillations. We note an overshoot of 70 pixels in the simulation.

With optical flow in the control law, control of all axes is achieved faster with a smaller overshoot (25 pixels). We can say, then, that the main contribution of the optical flow, in this case, is to stabilize the control process.

## 6.2 Experimental results

To validate our approach, we programmed the control law on our experimental robotic platform. This is a cartesian robot with 6 degrees of freedom (built by the firm AFMA Robot) and the parallel vision system WINDIS [10] [14]. This whole platform is controlled by a VME system, and can be programmed in C language under the VxWorks real time operating system.

*1. Overview of the parallel vision system*

This architecture, developed in collaboration with INRIA in Sophia Antipolis, France, implements the concept of active windows. Two boards have been developed for low and intermediate level processing. At video rate, several active windows (of varying size and position) are extracted from the image. Low level processing is then executed, and results are sent to the DSP module (intermediate level processing) through Disbus.

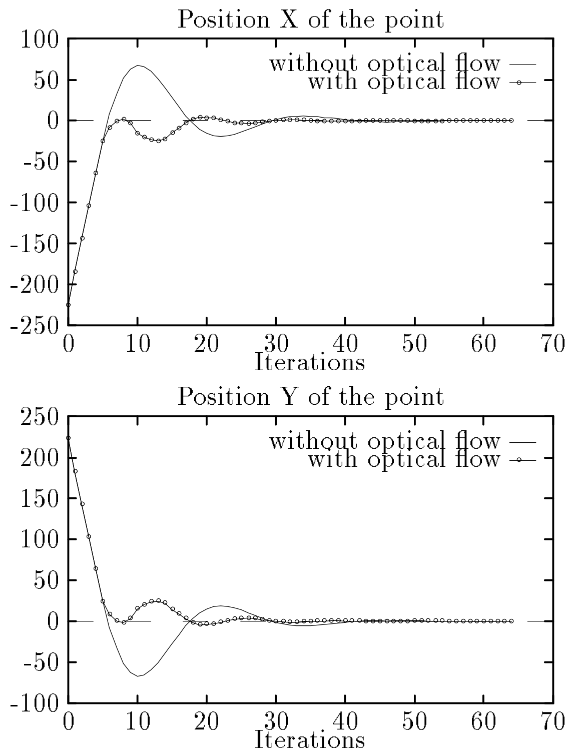


Fig. 2 Visual information : position in image space.

Windis architecture comprises three basic modules (corresponding to three different VME boards). Depending on the complexity of the application, the system will be built around one or more of these basic modules in a such a way that we will be able to attempt video rate performances. The three basic modules are :

- **WINDIS Window Distributor Subsystem:** This module is used for window extraction, the execution of low level processing and the distribution of active windows toward the Window Processing Subsystem. Window distribution consists in dispatching lists of selected pixels and grey levels through the window bus at the rate of 20 Mhz .

- **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. Window processing modules provide a geometric description of the required primitive in each window. Four DSP modules can be used with each VME mother board. Each module has 256k 32 bits of general memory data and can execute program developed in C at the rate of 33 Mhz.

- **WINMAN Window Manager Subsystem :** The window manager 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. A 68040 based cpu board implements this module.

For each level, we introduced parallelism allowing us to reach video rate for most of the application tasks. The management of such a system is a tricky job and requires the use of a real time operating system. For ease of use, we chose to develop the whole system under VxWorks.

### 6.3 Experimental results

In this section, we present different experimental results. The CCD camera is embedded on the end effector of the robot and is connected to the vision system WINDIS.
For the whole experimentation process, the initial position of the camera is close to 1 meter from the target. The target is composed of a simple "Led". The application runs at video rate (40 ms), but the vision system introduces a pipeline delay of 80 ms (due to image storage and processing in the window concerned).
The vision process is based on the extraction of the centre of gravity of the illuminated point in the image and on the active window tracking of this point along the sequence.

Figure 3 represents the evolution of visual information with and without optical flow.

Figure 4 gives the real trajectory of the point in the image space. We can observe that stabilization of the process also occurs in the experimental site.

To show the stabilization effect of optical flow, we introduced another delay. This delay of 2 iterations may represent the processing time for the extraction of the optical flow. It increases the pipeline delay by 80 ms, so the total amount of this delay becomes 160 ms. This delay represents the time between vision sensor acquisition and actuator control.

Figure 5 shows us the evolution of visual information with and without optical flow taking into account the total pipeline delay of 160 ms.

As in simulation tests, we can verify the stabilization of the control process due to the optical flow. It avoids too much overshoot while canalizing the energy of the system. Figure 6 also shows that the convergence of the control task is greatly improved and the number of iterations decreases from 100 to 50. In fact, the stabilization can be interpreted as a "phase advance corrector", which we usually find in the traditional "Proportional Derivative".

## 7 Conclusion and future work

A new control law, based on a visual servoing approach has been developed. The introduction of optical flow improves the stability and the convergence of the control process.

The main goal of the present work is to show the validity of this new approach. We considered a pipeline delay close to the real implementation of an optical flow extraction algorithm [1] in the vision process. We have introduced this delay into the visual servoing scheme and showed that its effect can be reduced. In our opinion, these first results are satisfying and attractive.

We are now working on the real time implementation of an optical flow extraction algorithm. This algorithm gives us the speed of geometric features in the image space. We will also be extending our new approach to more complex features, such as segments, circles etc.

## References


[1] Allen P.K., B. Yoshimi, A. Timcenko, Real time visual servoing, In Proceeding of the IEEE Conference on Robotics and Automation, 1991.

[2] Bajcsy R., Active Perception, In Proceeding of the IEEE, vol. 76, n.8, August 1988.

[3] Bensalah F.,F. Chaumette, Détection de rupture de modèle appliquée à l'asservissement visuel, Rapport de recherche INRIA, n.2425, November 1994, Rennes, France.

[4] Chaumette F, La relation vision-commande : Théorie et application à des tâches robotiques, Phd Thesis IRISA/INRIA, July 1990, Rennes, France.

[5] Chaumette F,A. Santos, Tracking a moving object by visual servoing, IFAC 12th Triennal World Congress, 1993, Sydney, Australia.

[6] Espiau B., F. Chaumette, P Rives, A new approach to visual servoing in robotics, IEEE Trans. on Robotics and Automation, vol. 8, n.3, pp.313-326, June 1992, Paris, France.

[7] Feddema J.T., O.R. Mitchell, Vision-guided servoing with feature-based trajectory generation, IEEE Trans. on Robotics and Automation, vol. 5, n.5, pp.691-700, October 1989.

[8] Horn B. K. P. and B. Shunck, Determining optical flow, Artificial intelligence, n.17, pp.185-203, 1983.

[9] Khadraoui D., G. Motyl , P. Martinet, J. Gallice, F. Chaumette, Visual servoing in robotics scheme using a camera/laser-stripe sensor, Publication interne IRISA n.898, January 1995.

[10] Martinet P., P. Rives, P. Fickinger, and J.J. Borrelly, Parallel Architecture for Visual Servoing Applications. Workshop on Computer Architecture for Machine Perception, CAMP'91, December 1991, Paris, France.

[11] Motyl G., Couplage d'une caméra et d'un faisceau laser en commande référencée vision Phd Thesis, September 1992, Clermont-Ferrand, France.

[12] Papanikolopoulos N., P.K. Khosla, T. Kanade, Vision and Control techniques for robotic visual tracking, In Proceeding of the IEEE Conference on Robotics and Automation, April 1991, Sacramento, USA.

[13] Papanikolopoulos N., P.K. Khosla, T. Kanade, Visual tracking of a moving target by a caméra mounted on a robot: A combination of control and vision, IEEE Trans. on Robotics and Automation, vol. 9, n.1, pp.14-35, February 1993, Sacramento, USA.

[14] Rives P., J.J. Borrelly, J. Gallice, and P. Martinet, A Versatile Parallel Architecture for Visual Servoing Applications, Workshop on Computer Architecture for Machine Perception, CAMP'93, December 1993, News Orleans, USA.

[15] Samson C., M. Le Borgne, B. Espiau, Robot Control : The task function approach Oxford University Press, 1991.

[16] Sundareswaran V., P. Bouthémy, F. Chaumette, Active camera self-orientation using dynamic image parameters ECCV, May 1994.

[17] Urban J.P., Une approche de type asservissement visuel appliquée à la robotique Phd Thesis, June 1990, Clermont-Ferrand, France.

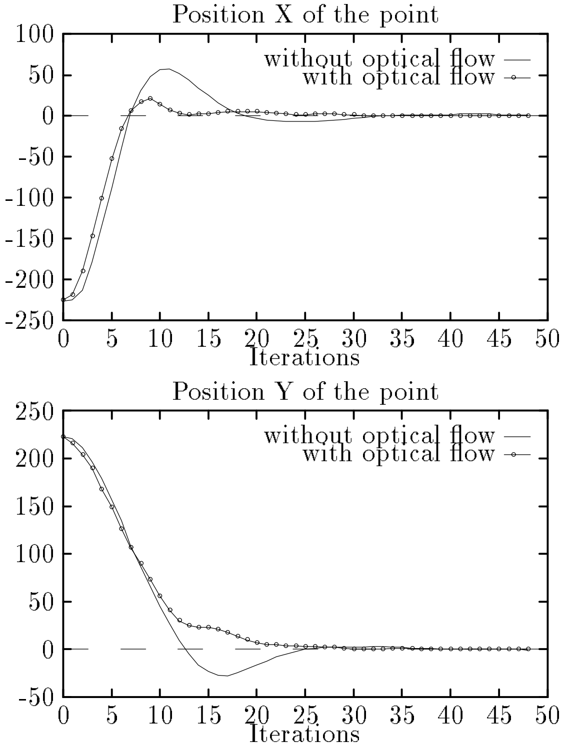


Fig. 3 Visual information : position in image space.

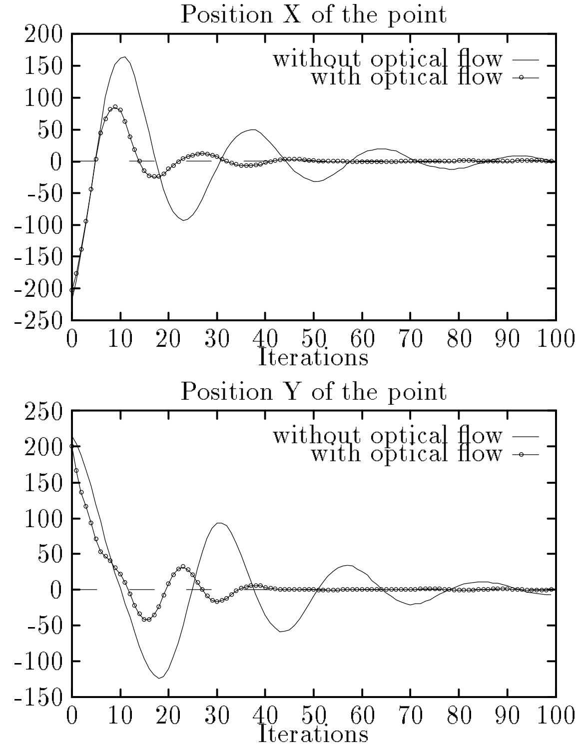


Fig. 5 Visual information : position in image space.

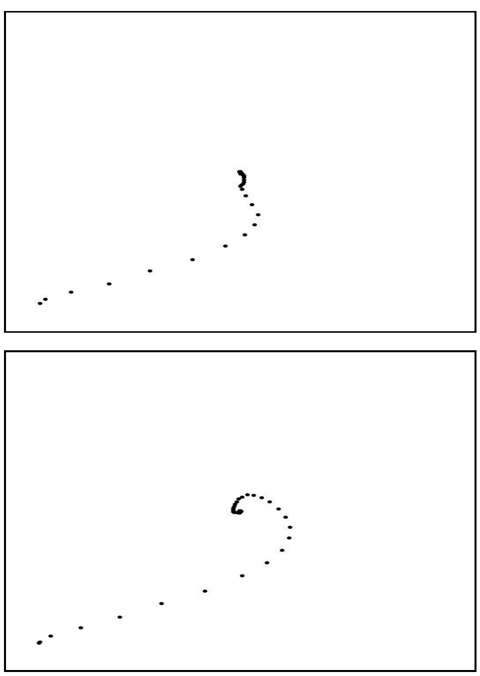

Fig. 4 Point trajectory in image space with and without optical flow.

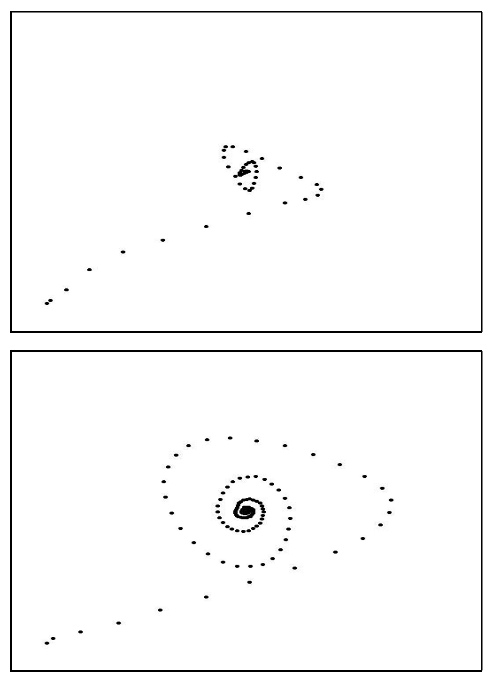

Fig. 6 Point trajectory in image space with and without optical flow.