Communication in International Journal of Robotics Research,
Vol. 15, n. 6, pp. 629-645, December 1996

Exploiting Image Motion for Active Vision in a Visual Servoing Framework

V. Sundareswaran[1], P. Bouthemy and F. Chaumette

IRISA/INRIA-Rennes
Campus Universitaire de Beaulieu
35042 Rennes Cedex
France

email: sundar@bu.edu, bouthemy@irisa.fr, chaumett@irisa.fr

Abstract

The visual servoing *formalism provides a framework for achieving tasks in an active closed-loop fashion based on visual information. We consider a typical visual servoing approach that uses geometric information about image features for controlling the position and attitude of a camera, and extend the applicability of this approach by using image motion information. We present two different approaches to visual tasks that use motion information. The first uses the focus of expansion which is related to the translational velocity of the camera. The second incorporates the parameters of the 2D affine motion model in control equations. Both these approaches are illustrated by means of a task of aligning the optical axis of the camera with the unknown direction of translational motion of the system on which it is mounted. Results of simulation experiments and real experiments on a six DOF robot with a camera on its end-effector are presented.*

Keywords: Visual servoing, visual motion, active vision, focus of expansion, 2D affine motion.

1 Introduction

In recent years, many approaches have been taken to perform tasks based on visual information. These approaches have been motivated by a need to *react* to visual stimuli in appropriate ways, resulting in a coupling between perception and action. Such approaches fall within the realm of *active vision* because of the nature of the generic mechanism involved: visual input is processed, and based on information derived from the processing, camera behavior is controlled, usually with the goal of attaining a configuration that simplifies further action (Aloimonos et. al 1988, Bajcsy and Campos 1992, Ballard and Brown 1992). Approaches in vision-based behavior span a broad

[1]Present address: Biomedical Engineering, Boston University, 44, Cummington street, Boston, MA 02215, USA

spectrum ranging from methods that perform very specific tasks to methods that provide general framework. An extensive survey of all the methods is beyond the scope of this paper. We mention a few in each of the broad categories.

There are many examples for approaches that perform specific tasks. Nelson and Aloimonos (1989) proposed a scheme to detect obstacles by using motion flow field divergence; Santos-Victor et. al (1993), and Coombs and Roberts (1993) presented methods to steer a camera between two walls, and to veer around obstacles, both methods being based on a simple analysis of the computed optic flow fields. Performing saccades in real time to moving regions of interest has been demonstrated in Murray et. al (1993). Krotkov (1987) presented a method to perform focusing, based on a special search technique. Yuille and Geiger (1990) proposed a method to control movements of stereo cameras to help in solving the correspondence problem. Coombs and Brown (1993) outlined a method to control camera movements to keep stereo cameras locked on a target. Olson and Coombs (1991) presented a real-time vergence control for stereo cameras by estimating vergence error with a cepstral disparity filter.

In the middle of the spectrum, there are methods that perform more general tasks such as gaze control and fixation; these are more general because fixation or gaze control can be used as a means to accomplish other tasks. Grosso et. al (1992) computed the time-to-impact using the vergence control of a binocular camera head. Ballard and Brown (1993) outlined the uses of gaze control, and fixation in particular, in obtaining an object-centered reference-frame and in figure-ground discrimination. A system performing several active visual tasks, including closed-loop gaze-control (based on a differential analysis) for fixating on an object was presented by Grosso and Ballard (1993). A fixation method running in real-time on a head-eye system was described by Pahlavan et. al (1993). Target-tracking methods by cameras mounted on robots have been demonstrated (Allen et. al 1993).

On the far end of the spectrum, there are methods that provide general solutions such as *visual servoing*. Visual servoing methods (Espiau et. al 1992, Feddema and Mitchell 1989, Hashimoto 1993, Papanikolopoulos et. al 1993, Weiss et. al 1987) present control-theoretic approaches for closed-loop control of the position and attitude of the camera, using visual information. Wijesoma et. al (1993) provide a comparison of different vision-based control strategies. These methods are general, and can be used to perform a variety of tasks, each with its specific goal. For instance, the formalism of Espiau et. al (1992) has been used in a wide range of tasks such as absolute positioning (Espiau et. al 1992), target tracking (Chaumette and Santos 1993), and structure estimation (Chaumette et. al 1994). Chaumette et. al (1994), for example, use visual servoing to choose the camera motion that leads to a robust and accurate estimation of 3D structure. Our present work is in extending the power of this class of approaches, by showing how *visual motion* information can be incorporated.

We start from one of the visual servoing formalisms (Espiau et. al 1992). The procedure outlined by Espiau et. al (1992) for determining closed-loop control equations is suited for the use of geometric information about primitives such as points, lines, and circles, all of which can be parametrized. We show how this formalism can make use of visual motion information too. In particular, the methods described in this paper use dynamic image parameters such as focus of expansion or coefficients in the 2D affine motion field model. Such a use of dynamic image parameters for the purpose of visual servoing is new, and it provides a starting point for other interesting closed-loop methods

using these parameters.

The ability to use motion information in visual servoing extends the applicability of active visual techniques. The ability to operate in a dynamic environment is useful and is often critical. Certain remarks are in order here. Using motion information is of relevance when the camera is undergoing certain unknown motion, and/or when mobile objects are in the field of view. This is the case when the camera is acting as a sensor for the host vehicle whose primary mission is not mere visual exploration. This is not an unusual scenario. Consider Figure 1 which depicts navigational situations where a vehicle carrying a camera moves about (to perform tasks such as driving on a highway, or performing a landing sequence), and the camera acts as a sensory device for avoiding obstacles, locating targets etc. Contrast this to a situation where the camera is stationary except when performing a vision-based task (e.g., positioning). It is important to note that in the former case, *usually* there are less degrees of freedom available for control than in the latter case (however, when the motion is only external, as with moving objects in the scene, all the degrees of freedom may be available for control). In the situations in Figure 1, only the rotational motion of the camera is controllable.

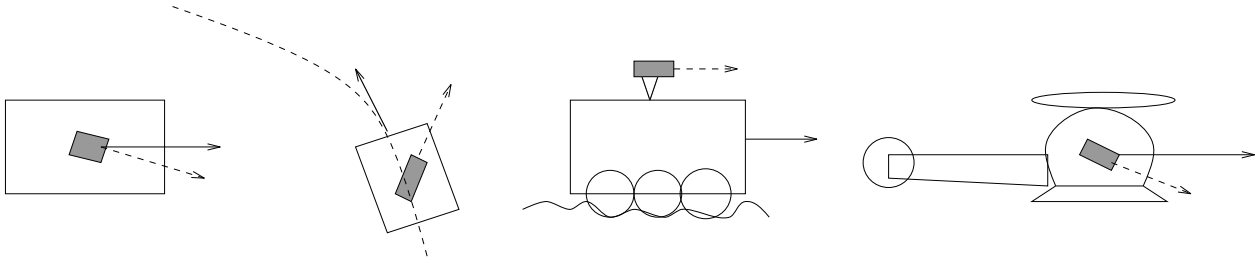


Figure 1: Navigational situations

In this paper, we present methods that control only the rotational velocity of the camera. We examine the use of two kinds of motion information. We show that the focus of expansion can be treated just like geometric information. Thus, a straightforward application of the visual servoing principle is possible. We illustrate this by considering the task to align the optical axis of the camera with the unknown direction of translation–henceforth referred to as the *alignment task*. We also show how the visual servoing can be used in completely new ways, by incorporating the parameters of the 2D affine motion model in the control equations. We demonstrate the approach by presenting control equations for the alignment task. From another viewpoint, the task being solved is an active method to determine the FOE. In other words, the direction of translation (or equivalently, the FOE) is found by an active method that in fact aligns the camera axis with the direction of translation. Barth and Tsuji (1993) have proposed a scheme for the alignment task, based on fixation. A comparison with their scheme is more relevant after our methods are presented, so we postpone a discussion of this to the end.

In summary, we present an approach to the tight coupling of behavior and motion perception. We note that our innovative use of dynamic image parameters in closed-loop control is not limited to the task described here. It has general applicability; it is possible to use such dynamic parameters for various active vision tasks to obtain fast and robust control methods. Our intention

here is to provide a starting point for such approaches. Parts of this work have appeared before (Bouthemy and Sundareswaran 1993, Sundareswaran et. al 1994a, Sundareswaran et. al 1994b, Sundareswaran et. al 1994c).

2 Visual Servoing

In this section, we review the basic principles of visual servoing. Detailed descriptions can be found elsewhere (Espiau et. al 1992, Feddema and Mitchell 1989, Hashimoto 1993, Papanikolopoulos et. al 1993, Weiss et. al 1987).

The principle of visual servoing is to use visual information as observation in closed-loop control when the desired configuration can be described as a particular visual observation. The control is effected on the camera position and orientation or on an external object such as a robot arm. The only condition is that the instantaneous change in the visual information (in the sense of temporal derivative) as a function of the controllable parameters be known analytically. Intuitively, if the effect of the control parameters on the observation is known, we could provide the appropriate control in order to result in obtaining the desired observation. The visual servoing theory, described in the references mentioned above, provides a framework for determining the *control law* which is simply a set of equations to calculate the control parameters.

More precisely, for a given vision-based task, we have to choose a set $\mathbf{s}$ of visual features suited for achieving the task (for example, the coordinates of an image point, the parameters of a selected line, etc). In order to perform a control law based on $\mathbf{s}$, we need to know the equations for the variation of $\mathbf{s}$ with respect to camera translational and rotational motion (T, Ω). In other words, we have to determine the matrix L described by the following equation:

$$\dot{\mathbf{s}} = L \begin{pmatrix} T \\ \Omega \end{pmatrix} \tag{1}$$

Espiau et. al (1992) showed that the set of parameters whose variation can be written in the form of Equation 1 are such that they only depend on the relative position and orientation between the camera and the scene. This set of parameters includes parameters describing the position in the image of any geometrical features (points, straight lines, etc.). In contrast, parameters such as photometric parameters (which depend on lighting) and those based on image motion (which depend on the relative velocity between the camera and the scene) are excluded. However, we will see in this paper that parameters based on image motion can be used in visual servoing, even if their change rate can not be expressed linearly to the camera velocity as stated in Equation 1.

In the terminology of Espiau et. al (1992), L is called the *interaction matrix* related to $\mathbf{s}$. For example, if $\mathbf{s}$ is the location (x_p, y_p) of the object in the image (say, its center of gravity), this matrix can be obtained from the well-known equations relating the three-dimensional motion of a point to the motion of its projection on the image (Horn 1987) for the considered point:

$$\begin{aligned} \dot{x_p} &= \tfrac{1}{Z(x_p,y_p)} \left[-U + x_p W\right] + A x_p y_p - B\left[1 + x_p^2\right] + C y_p, \\ \dot{y_p} &= \tfrac{1}{Z(x_p,y_p)} \left[-V + y_p W\right] + A\left[1 + y_p^2\right] - B x_p y_p - C x_p. \end{aligned} \tag{2}$$

We assume that we have planar images obtained by the pin-hole perspective approximation, with a focal length of unity. We adopt the standard coordinate systems shown in Figure 2 (the spherical coordinates could be adopted as readily). The translational velocity T has components U, V, and W. The components of the rotational velocity Ω are A, B and C.

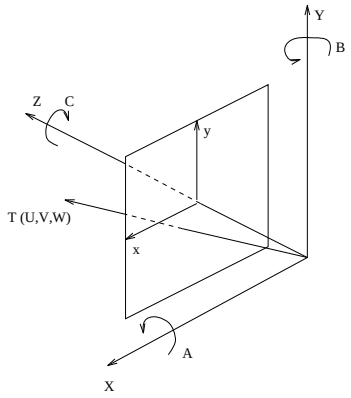


Figure 2: The coordinate systems for the 3D environment and the image. The 3D coordinate system has its origin at the camera projection center, and the image coordinate system has its origin at the center of the image.

Feddema and Mitchel (1989) give results for more complex visual features (image segments), and Espiau et. al (1992) propose a general method for computing the interaction matrix of any visual feature defined upon geometrical primitives and give explicit results for the parameters describing the projection in the image of lines, circles, spheres and cylinders.

A *task function* $\mathbf{e}$ can be defined as

$$\mathbf{e} = M\,(\mathbf{s} - \mathbf{s}^*), \tag{3}$$

where

- $\mathbf{s}$ is the measured visual features currently observed by the camera,
- $\mathbf{s}^*$ is the desired final configuration for $\mathbf{s}$ in the image, and
- M is a constant matrix which allows, for robustness issues, to take into account more visual features than necessary. Let us note that M can simply be chosen as the identity matrix when the number of the selected visual features is equal to the number of the camera d.o.f. controlled by the task.

The control problem thus appears as the regulation of the task function $\mathbf{e}$ to zero or, equivalently, as the minimization of $\|\mathbf{e}\|$ in the image by appropriate camera motion. We would like the task function to decay exponentially towards zero. For such a requirement,

$$\dot{\mathbf{e}} = -\lambda \mathbf{e},$$

where $\lambda(>0)$ is the exponent that controls the speed of the decay. Noting that M and $\mathbf{s}^*$ are constant and assuming for simplicity that the scene is static, we obtain:

$$ML \begin{pmatrix} T \\ \Omega \end{pmatrix} = -\lambda \mathbf{e}. \tag{4}$$

Inverting Equation 4, we get the control law

$$\begin{pmatrix} T \\ \Omega \end{pmatrix} = -\lambda L^+ M^+ \mathbf{e}, \tag{5}$$

where L^+ and M^+ are the pseudo-inverses of L and M.

Generally, the interaction matrix L cannot be exactly computed. For example, in the case of a point, described by Equation 2, L depends on the depth Z which is generally unknown. In such cases, a model of L (say $\hat{L}$) has to be chosen and we finally obtain:

$$\begin{pmatrix} T \\ \Omega \end{pmatrix} = -\lambda \hat{L}^+ M^+ \mathbf{e}, \tag{6}$$

An exponential decay of $\mathbf{e}$ will be ensured under the sufficient condition (Espiau et. al 1992):

$$ML\hat{L}^+ M^+ > 0 \tag{7}$$

in the sense that an $n \times n$ matrix A is positive-definite if $x^T Ax > 0$ for any nonzero $x \in I\!R^n$. Usually, $\hat{L}$ is chosen as L^*, the value of L at convergence. Indeed, in that case, the positivity condition is valid around the desired configuration (Samson et. al 1991).

Thus, the principle of visual servoing is to use visual information to perform closed-loop control to reduce an "error" in the visual information. Various tasks have been performed within the framework of the visual servoing outlined here and described in detail by Espiau et. al (1992). We propose to extend this visual servoing framework by making use of other important visual features such as the focus of expansion or the 2D affine motion parameters.

3 Image motion information

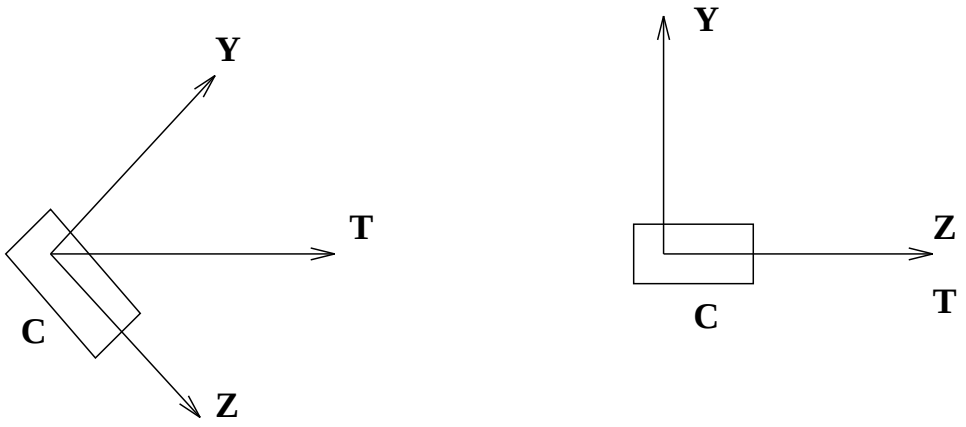
The traditional visual servoing approach described in the previous section provides a method to determine the camera control in such a way as to have a chosen geometric property (eg., an image feature reaches a desired location). For instance, we could bring the camera to position itself at a distance of, say, 30 cm from a square of known size, by requiring that the image projections of the corners of the squares reach predetermined locations on the image (Espiau et. al 1992).

However, the visual servoing approach is more general. Indeed, it is well-known that we can control *any* parameter, as long as its variation can be expressed in terms of the parameters that can be controlled. We use this classical property by using parameters obtained from image motion. First, we show that we can use the focus of expansion. This is only a modest step forward because the focus of expansion (FOE) is a geometric property; this is because the FOE is a location on the

image plane, and can be treated as such. Nevertheless, this viewpoint brings out the notion of using motion information available as a simple geometric feature (another such feature is the projection of the axis of rotation on the image plane).

Secondly, we show how to use the parameters of the 2D affine model (henceforth, *affine motion parameters*) of the optic flow. Here, we provide a significant extension to the visual servoing framework. The affine motion parameters have been well-studied theoretically (Koenderink and Van Doorn 1975) as well as empirically (Negahdaripour and Lee 1992, Bouthemy and François 1993), and have been used in many applications.

Our work is in the same spirit as Murray et al. (1993) and Cipolla and Blake (1992) who use 2D affine motion parameters in directly controlling camera behavior. We will illustrate our approaches by proposing solutions to a task of aligning the optical axis of a camera with the camera's unknown direction of translation. The initial and final configurations of a camera (C) undergoing alignment is shown in Figure 3. For simplicity, only the projection on the YZ plane is shown. The translation vector is T. For the purposes of this paper, we restrict ourselves to pure external translation of the camera (the first one in Figure 1), and hope that this will provide an initiative to solve the more general cases.



(a) Before Control (b) After Control

Figure 3: Effect of the alignment control: before control, the camera could be viewing in an arbitrary direction; after control, the camera looks in the forward direction.

The alignment task can be accomplished by rotations of the camera with respect to the X and Y axes (i.e., pan and tilt). This is evident if we note that rotations about two axes are sufficient to orient the coordinate system in any desired way. Thus, while the rotation about the Z axis (roll) could also be used, it is not necessary. According to control theory (Espiau et. al 1992), this additional degree of freedom can be used to perfom auxiliary tasks, but we will not explore that possibility here. We will restrict our attention to the computation of the control rotational velocities in the pan and tilt directions.

We show the theoretical development first, and then present experimental results.

4 Control using the FOE

The focus of expansion (x_f, y_f) is the projection of the 3D translational velocity vector on the image plane: $x_f = \frac{U}{W}$, $y_f = \frac{V}{W}$. The location (x_f, y_f) may be treated as a geometric feature and used to obtain control laws. For instance, we could apply control to bring this *feature* to the center of the image, thus accomplishing the specified alignment task.

We only control camera pan A and tilt B, and we know, from the optic flow equations in Equation 2,

$$\begin{pmatrix} \dot{x_f} \\ \dot{y_f} \end{pmatrix} = L \begin{pmatrix} A \\ B \end{pmatrix}, \quad L = \begin{pmatrix} x_f y_f & -(1+x_f^2) \\ 1+y_f^2 & -x_f y_f \end{pmatrix}.$$

In this case, the matrix L can be estimated on-line since it only depends on the position (x_f, y_f) of the FOE measured in the image. In practice, however, an appropriate measure of the FOE is sufficient; in our control scheme (whose details appear later in this section), for instance, we use the direction to the FOE, without affecting the overall task. In Equation 5, we can set M as the identity matrix and we finally obtain:

$$\begin{pmatrix} A \\ B \end{pmatrix} = -\lambda L^{-1} \begin{pmatrix} x_f - x_f^* \\ y_f - y_f^* \end{pmatrix}, \tag{8}$$

where

$$L^{-1} = \frac{1}{1+x_f^2+y_f^2} \begin{pmatrix} -x_f y_f & 1+x_f^2 \\ -(1+y_f^2) & x_f y_f \end{pmatrix}.$$

For the alignment task, $x_f^* = y_f^* = 0$. One could also position the FOE at any desired location (x_f^*, y_f^*) on the image plane using the same method.

4.1 Computing the FOE

Here we describe how we can quickly determine an estimate of the location of the FOE for translational motion (i.e., no rotational motion). Consider the optical flow equations in Equation 2. Since there is no rotational velocity, the equations are simplified:

$$\begin{aligned} u(x,y) &= \tfrac{1}{Z(x,y)}\left[-U + xW\right], \\ v(x,y) &= \tfrac{1}{Z(x,y)}\left[-V + yW\right], \end{aligned} \tag{9}$$

where $Z(x,y)$ is the depth of the point projected at (x,y). It is easy to see that the point of intersection of these image plane vectors gives the Focus of Expansion (FOE). Indeed, at this point $(x = \frac{U}{W}, y = \frac{V}{W})$, $u = v = 0$, and the flow vectors spread out in a radial direction from this point (see Figure 4 (left)). The vector field (u, v), termed the optical flow field, has to be computed in order to determine the FOE as the intersection of the vectors. There is a lack of reliable methods to compute the optic flow rapidly enough to be implementable in current image processing boards so as to use in a closed loop scheme. So, we suggest that it is desirable to estimate the FOE from the *normal flow field* vectors; the normal flow vector is given by (Horn 1987)

$$\vec{v_n} = \frac{-I_t}{\| \vec{\nabla} I \|^2} \vec{\nabla} I,$$

where I is the spatio-temporal image intensity function, and I_t and $\vec{\nabla} I$ are its temporal and spatial derivatives. This is obtained from the image motion constraint equation

$$\frac{dI}{dt} = 0,$$

or, equivalently

$$I_x u + I_y v + I_t = 0,$$

from which only the projection of (u, v) in the local intensity gradient (I_x, I_y) direction (in other words, on the normal to the local *edge*–hence the name *normal flow*) can be computed, and the magnitude of this projection is determined by the partial temporal derivative I_t. This constraint is a manifestation of the *aperture problem* that complicates the computation of the exact optical flow field, and in most cases, the optical flow field can only be approximately computed based upon certain assumptions about the regularity of the field.

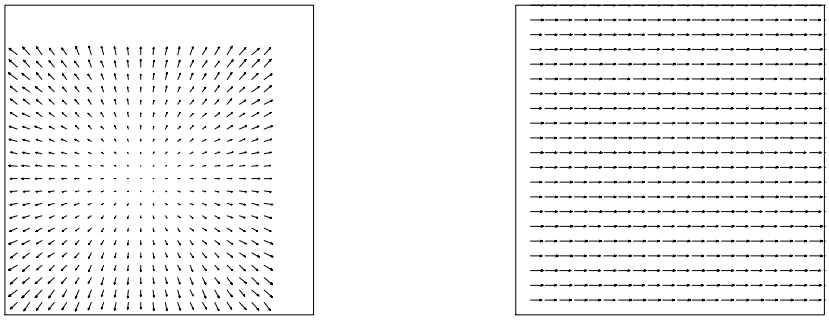


Figure 4: Optical flow for forward (left) and sideways translation (right).

We consider the optical flow induced due to the translation of the camera. When the camera is translating forward (i.e., with the FOE at the center of the image), the flow field obtained is *radial*, similar to the one shown in Figure 4 (left). On the other hand, if the motion is *sideways*, the flow field obtained is a collection of nearly parallel vectors (see Figure 4 (right)). The goal is to rotate the camera in such a way as to result in a radial flow field; one could potentially begin with a parallel flow field (i.e., with the camera looking sideways or upward).

The clue about the x coordinate of the FOE is given by the distribution of the horizontal components of the flow vectors. Consider the two half-regions of the image, indicated by L and R in Figure 5. If the FOE is at the center (that is, on the line dividing the two regions L and R), then the horizontal components of the flow vectors in L will be oriented towards the left, and the horizontal components of the flow vectors in R will all be oriented towards the right, as for the situation in Figure 4 (left). If this is not the case, then the FOE is not located on the dividing line between the two regions. In particular, if *all* the flow vectors on the image are oriented to the right (Figure 4 (right)), then the FOE is on the left, outside the image. In general, the ratio of the number of vectors oriented to the left to the number of vectors oriented to the right yields an

estimate of the x coordinate of the FOE. With this approach, if the FOE is outside the image, we can only determine the direction to the FOE. In a similar fashion, we can estimate the y coordinate of the FOE, using the vertical components of the flow vectors. The fact that we only estimate the direction to the FOE when the FOE is outside the image does not affect the control greatly; we assume that the FOE is on the border of the image in the computed direction and calculate the control.

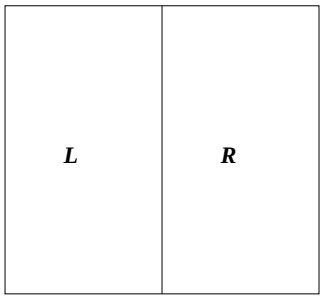


Figure 5: Two halves of the image.

The method described here permits an implementation that is rapid and is accurate enough for the control scheme when the FOE is not exactly at the center. After convergence, however, if the control is not withdrawn, the approximate computation of the FOE creates a minor oscillation of the center about the FOE. As will be seen in the experiments, it is small enough to be negligible but perfect precision is not guaranteed.

We can use normal flow instead of optical flow. We show in the Appendix that the direction of the components of optical flow coincide with the direction of the components of normal flow, with high probability. Since the control proposed here uses only the direction of the components, such a consistency might be sufficient, and it has been verified experimentally. Aloimonos and Duric (1992) proposed a different approach that is based on finding intersection of half planes defined by the normal flow vectors.

We now summarize the method with more details: the normal flow field is computed, and the fraction h_p of positively oriented (i.e., towards the left) horizontal components and the fraction v_p of positively oriented (upwards) vertical components are determined by counting. Assuming a uniform distribution of image gradient directions, the approximate FOE is computed as

$$(-(h_p - 0.5) * W_I/2, -(v_p - 0.5) * H_I/2),$$

where W_I and H_I are the width and the height of the image in focal length units. This gives a location on the image if the FOE is within the image, or a location on the border of the image if the FOE is on the border or is outside the image; in the last situation where the FOE is outside the image, the computed position is in the direction of the FOE.

5 Control using the 2D affine parameters

In this section, we propose to use, under the same servoing formalism, measures obtained from the affine motion parameters. We consider the same (alignment) task as in the previous section, to illustrate the derivation method. We begin by describing the parameters of the 2D affine motion model, and then derive the control laws to do the alignment task using these parameters.

5.1 The 2D affine motion parameters

The 2D affine motion model (Equation 11) is often useful. It is possible to derive expressions for the first-order parameters (affine parameters) assuming that an analytical surface is imaged (i.e., it is possible to describe the depth by a Taylor series expansion). It has been already shown (Negahdaripour and Lee 1992) that the affine parameters can be reliably estimated. Multiresolution methods for the estimation of the affine parameters (Bergen et. al 1992) yield accurate values.

The optical flow equations are repeated here from Equation. 2 (see also Figure 2):

$$\begin{array}{rcl} u(x,y) & = & \frac{1}{Z(x,y)}\left[-U + xW\right] + A\left[xy\right] - B\left[1 + x^2\right] + Cy, \\ v(x,y) & = & \frac{1}{Z(x,y)}\left[-V + yW\right] + A\left[1 + y^2\right] - B\left[xy\right] - Cx. \end{array} \tag{10}$$

Let the first-order approximation be

$$\begin{array}{rcl} u(x,y) & = & a_1 + a_2 x + a_3 y, \\ v(x,y) & = & a_4 + a_5 x + a_6 y. \end{array} \tag{11}$$

Let the first-order model for the imaged surface be (i.e., local planar approximation)

$$Z = Z_0 + \gamma_1 X + \gamma_2 Y.$$

It is easy to show that

$$\frac{1}{Z} = \frac{1}{Z_0}(1 - \gamma_1 x - \gamma_2 y), \tag{12}$$

From Equations (10), (11), and (12), we get (Bouthemy and François 1993, Negahdaripour and Lee 1992):

$$\begin{array}{rclrcl} a_1 & = & -\frac{U}{Z_0} - B, & a_4 & = & -\frac{V}{Z_0} + A, \\ a_2 & = & \frac{1}{Z_0}(\gamma_1 U + W), & a_5 & = & \frac{1}{Z_0}(\gamma_1 V) - C, \\ a_3 & = & \frac{1}{Z_0}(\gamma_2 U) + C, & a_6 & = & \frac{1}{Z_0}(\gamma_2 V + W). \end{array} \tag{13}$$

5.2 Control method

We have,

$$a_1 = -\frac{U}{Z_0} - B, \;\; a_4 = -\frac{V}{Z_0} + A. \tag{14}$$

Since we are considering external motion that does not have rotation, the parameters a_1 and a_4 should be zero if alignment has been achieved (since $U = V = 0$ in this condition). Therefore, the

first approach to use the affine parameters will be to provide control so as to attain $a_1 = a_4 = 0$. The flaw in this observation is that the rotational velocity will *not* be zero once control has begun. This results in a *fixating* situation, instead of the aligned situation where the angular velocity components will be zero. This problem is discussed in more detail in Sundareswaran et al. (1994b). We consider here another approach.

Consider an alternative set of parameters $U_z = \frac{U}{Z_0}$ and $V_z = \frac{V}{Z_0}$. From the equations for a_1 and a_4, it is clear that

$$U_z = -a_1 - B, \;\; V_z = -a_4 + A. \tag{15}$$

If we apply control in such a way to result in zero values for these variables, we will achieve the goal of setting the components U and V of the translational velocity to zero (the tacit assumption is that infinite depth does not occur). This is an intuitive set of parameters to control, because the quantities that we desire to control (namely U and V), are directly related to the chosen parameters.

The derivatives of these parameters are given by

$$\begin{pmatrix} \dot{U}_z \\ \dot{V}_z \end{pmatrix} = \begin{pmatrix} -\dot{a_1} - \dot{B} \\ -\dot{a_4} + \dot{A} \end{pmatrix} = \begin{pmatrix} \frac{\dot{U}}{Z_0} - \frac{U\dot{Z}_0}{Z_0^2} \\ \frac{\dot{V}}{Z_0} + \frac{V\dot{Z}_0}{Z_0^2} \end{pmatrix}. \tag{16}$$

The components of the velocity, which remain constant in a global coordinate system, change however in the camera coordinate system because of the control rotation of the camera axes. Since the rotational velocity is $\Omega = (A, B, C)$ about the three axes, the change in the translational velocity $T = (U, V, W)$ is simply the cross product

$$\dot{T} = -\Omega \times T. \tag{17}$$

In detail,

$$\begin{pmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{pmatrix} = \begin{pmatrix} CV - BW \\ AW - CU \\ BU - AV \end{pmatrix}. \tag{18}$$

From Equations 15, 16, and 18, we get

$$\begin{pmatrix} \dot{U}_z \\ \dot{V}_z \end{pmatrix} = \begin{pmatrix} C & -\frac{W}{Z_0} + \frac{\dot{Z}_0}{Z_0} \\ \frac{W}{Z_0} - \frac{\dot{Z}_0}{Z_0} & C \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} + \begin{pmatrix} -Ca_4 + a_1\frac{\dot{Z}_0}{Z_0} \\ Ca_1 + a_4\frac{\dot{Z}_0}{Z_0} \end{pmatrix}.$$

When the planar approximation to the viewed surface does not have a large angle of inclination with respect to the camera, we have

$$\frac{1}{\tau_c} = \frac{W}{Z_0} \approx -\frac{\dot{Z}_0}{Z_0} \approx \frac{a_2 + a_6}{2}, \tag{19}$$

where τ_c is used to denote the instantaneous *time-to-collision*. Furthermore, we assume that $C = 0$ since we do not control this velocity. We thus obtain:

$$\begin{pmatrix} \dot{U}_z \\ \dot{V}_z \end{pmatrix} = \begin{pmatrix} 0 & \frac{-2}{\tau_c} \\ \frac{2}{\tau_c} & 0 \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} + \begin{pmatrix} \frac{-a_1}{\tau_c} \\ \frac{-a_4}{\tau_c} \end{pmatrix}. \tag{20}$$

The task function is as before:

$$\mathbf{e} = \mathbf{s} - \mathbf{s}^*,$$

where

$$\mathbf{s} = \begin{pmatrix} U_z \\ V_z \end{pmatrix}.$$

For an exponential decay of the task function,

$$\begin{pmatrix} \dot{U}_z \\ \dot{V}_z \end{pmatrix} = -\lambda \begin{pmatrix} U_z - U_z^* \\ V_z - V_z^* \end{pmatrix}.$$

Rearranging Equation 20, we obtain the following control law :

$$\begin{pmatrix} A \\ B \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & -\tau_c \\ \tau_c & 0 \end{pmatrix} \left(\lambda \begin{pmatrix} U_z \\ V_z \end{pmatrix} - \begin{pmatrix} \frac{a_1}{\tau_c} \\ \frac{a_4}{\tau_c} \end{pmatrix} \right). \tag{21}$$

Using the approximations (19), the positivity condition (7) is ensured when the task is realized. Convergence of the control law will thus be obtained if the initial configuration is *not too far* from convergence. Our experiments, wherein divergence was never observed, confirm that such approximations do not disturb the task behavior.

Let us finally note that the observations U_z and V_z are given by

$$\begin{aligned} U_z &= -a_1 - B, \text{ and} \\ V_z &= -a_4 + A, \end{aligned}$$

where A and B, under the conditions presented here–assuming that there is no external agent causing rotational velocity, are nothing but measures of the rotational velocities resulting from the control applied at the preceding instant.

5.3 Calculating the 2D affine parameters

The 2D affine parameters can be calculated directly from the spatial and temporal derivatives of the image sequence. Recalling that the model is

$$\begin{aligned} u(x,y) &= a_1 + a_2 x + a_3 y, \\ v(x,y) &= a_4 + a_5 x + a_6 y, \end{aligned} \tag{22}$$

and the motion constraint equation (Horn 1987) is

$$I_x u + I_y v + I_t = 0,$$

where all the quantities are functions of x and y (indices omitted for simplicity).

It is easy to see that a direct substitution of $u(x,y)$ and $v(x,y)$ from Equation 22 in the last equation yields one equation in six unknowns. Thus, in principle, knowing the spatial and temporal derivatives at six points (belonging to a region whose projected motion satisfies the affine model),

we can solve for all the six affine parameters. In practice, for robustness reasons, a large number of points are considered, and a standard procedure for solving an overconstrained set of equations is used to determine the parameters.

The procedure presented above is the simplest possible approach; other methods using multiresolution (Bergen et. al 1992), or techniques from robust estimation theory (Odobez and Bouthemy 1994) have been developed, but real-time implementations that can be used in closed-loop schemes are still lacking. The sensitivity of the affine parameters has been analyzed by Negahdaripour and Lee (1992). In our experiments, we use the simple procedure presented above, with points taken from all over the image.

5.4 Discussion

In our approaches, we have restricted ourselves to pure external translation of the camera. It would be interesting to examine the general case where there is rotation also. For the FOE-based method, we need a more sophisticated method to determine the FOE in the presence of rotation. For the affine parameter method (Equation 21), the parameters U_z and V_z will be controlled to be zero; but this will not result in alignment (due to the external rotation). Thus, the problem can be alleviated if the external rotation can be measured and the parameters are chosen as

$$U_z = -a_1 - B_{\text{camera}} - B_{\text{external}}, \quad V_z = -a_4 + A_{\text{camera}} - A_{\text{external}},$$

the task of alignment can still be achieved. We would like to note, however, while this inconvenience arising due to external rotation is a problem for the alignment task, it is not necessarily a general constraint for motion information-based tasks.

6 Experiments

The two alignment methods, namely the one using the FOE and the one using the parameters U_z and V_z, have been studied in simulation and have also been implemented in a real system.

6.1 Simulation experiments

Here we present the approach to the simulation, the results of the simulation, and a brief discussion of the results.

6.1.1 Approach

The simulation was carried out using a discrete-time approach. Time ticks were chosen at unit intervals, and the control and the state of the system were calculated at these instants in time. This approach would closely approximate a real implementation with no asynchronous activities (i.e., time durations of actions such as acquiring images and computing the control are roughly constant in different iterations).

All the methods were implemented using the software package MATLAB. No noise was added during the simulations to avoid using an incorrect model of the noise; instead, we carried out experiments on a real system (see section 6.2).

Parameters that change continuously over time were discretized with a fine resolution. For example, the vector T undergoes a continuous change while the control is being applied; this change is given by Equation 17, repeated here:

$$\dot{T} = -\Omega \times T.$$

During the simulation, at a given time tick, we need to recalculate the translation vector T to account for the change since the last time tick; we divided the time interval between the two ticks into 500 equal parts and incrementally modified T.

We describe in the following paragraphs the results of the simulations, and how the results were used to validate the methods. Four different graphs are used to present the results. The first one contains curves showing the components of the 3D translational velocity in the camera coordinate system. For proper alignment, we expect the curves corresponding to U and V to go to zero, and the curve for W to achieve a large positive value equal to the speed (magnitude of the velocity) of the camera. The second graph shows the corresponding variation of the angular velocity components; we expect them to start from zero and go to zero at convergence; in between, they may have a non-zero value. The third graph shows the variation of the norm of the vector $(s - s^*)$ (see Equation 3) constituting the controlled parameters; this should go to zero at convergence with an exponential decay. The fourth and final graph shows the variation of the angle between the translation vector T and the optical axis Z of the camera; this should go to zero as well, at convergence.

6.1.2 Results

The curves in Figure 6 show the results from simulating the FOE-based algorithm. For this experiment, an initial translational velocity of $[2, 8, 2]$ focal units was used; this choice is not critical, and was used for demonstration purposes only. Any other initial translational velocity could be chosen, and the results will be qualitatively similar. A λ value of 0.2 was used. The translational velocity is controlled as desired, with U and V going to zero, leaving W as the only non-zero component. The angular velocities are computed correctly to provide the necessary correction to bring about the alignment. The "clipping" of the angular velocities is related to the qualitative nature of the FOE computation that is simulated. That is, when the FOE is outside the image, we can only compute the point on the border which is in the direction of the FOE from the center of the image. Thus, the angular velocity remains clipped until the FOE "enters" the image. The task function, and the angular separation betwen the optical axis and the translation vector go to zero, as expected.

The second simulation we present is the control using angular velocity, from Equation 21 (use of affine motion parameters). The curves are shown in Figure 7 with same configuration as in the previous case (initial translation $[2, 8, 2]$ focal units, $\lambda = 0.2$).

From the results of typical simulations shown here, the two control methods to perform the alignment task were empirically validated. In these noise-free simulations where the parameters are computed analytically, the methods appeared to be stable, and achieved convergence rapidly.

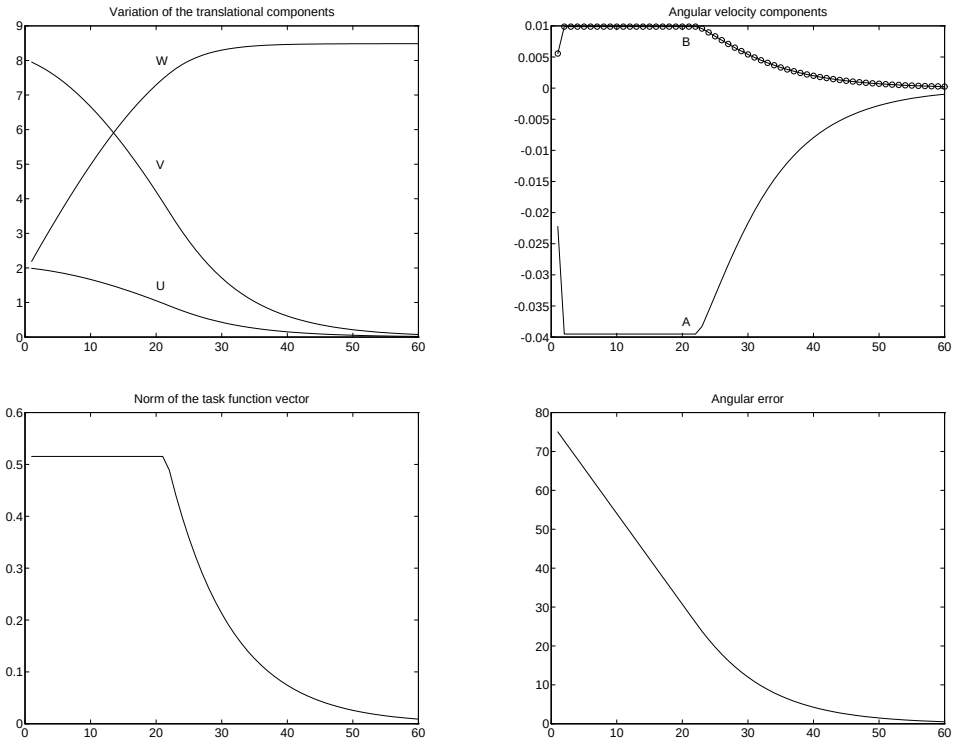


Figure 6: The FOE-based method: results. The top row shows the change in the velocity components (translational velocity components on the left, and the angular velocity components on the right). The bottom row shows the task function decrease on the left, and on the right, the angle between the translational direction and the optical axis. The reason for the flat parts of the curves on top right and bottom left is the due to the feature of the simulated FOE detection method that determines only the direction towards an FOE outside the image.

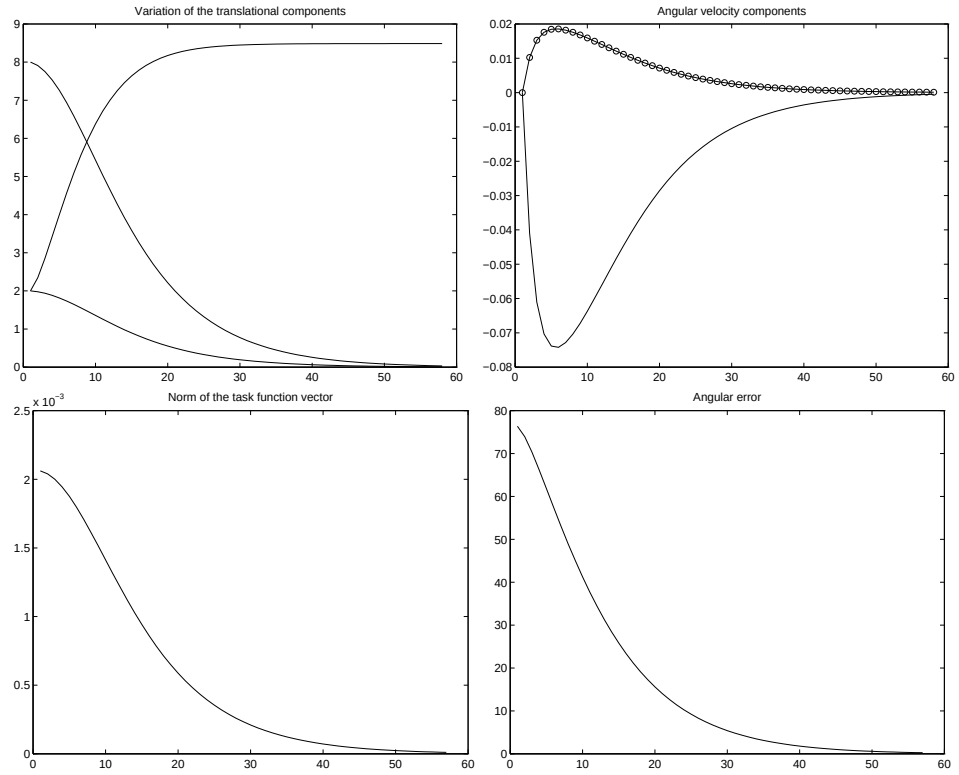


Figure 7: The angular velocity based method: results. The top row shows the change in the velocity components (translational velocity components on the left, and the angular velocity components on the right). The bottom row shows the task function decrease on the left, and on the right, the angle between the translational direction and the optical axis.

Theoretically however, there will be a problem in the angular velocity based method for large depth (since U_z and V_z are zero for infinite depth). Our simulation results did not suffer from this problem even for very large value of depth (until 10000 focal units). This is undoubtedly due to the fact that no noise was considered in the simulations. When calculated on real image sequences, large values of depth are likely to be problematic because the affine parameters will be very sensitive under such conditions. We have not yet quantified the effect of large depth since our experimental system described below does not allow imaging objects located far away.

6.2 Real experiments

In the experiments, we used a camera with a field of view of about 35 degrees mounted on a six degrees-of-freedom cartesian robot AFMA (see Figure 8). The camera can be positioned and oriented in the workspace with an accuracy of 0.5 mm and 0.05 degrees. The camera output is digitized by an image processing board (EDIXIA). For the experimental results reported here, the images are digitized and then subsampled by a factor of four (without any filtering) by the EDIXIA board and sent to the host (Sun Sparc 20). The size of the images processed is 128×182 pixels. All the image processing and control velocity computations are carried out on the host and the computed control is transmitted to the robot controller. The transmissions to and from the host occur via a BIT 3 Sbus/VMEbus board. A minimum of ten milliseconds are required for the controls to take effect. About 100 milliseconds are required for the entire process of acquiring, subsampling and transferring an image. Note that for the implementation, the camera parameters obtained from calibration, namely the center of the image and the interpixel distances, are required.

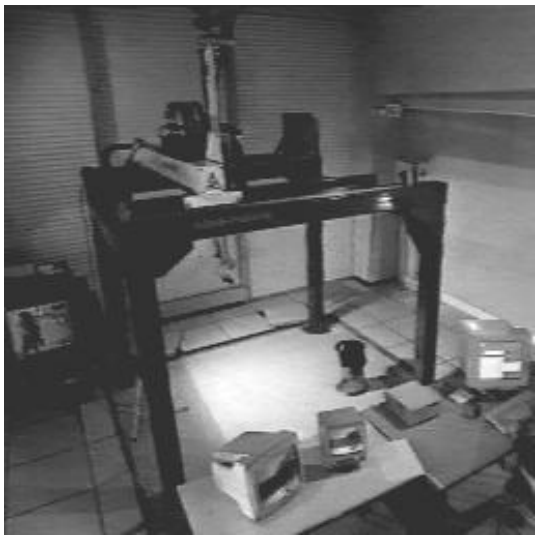


Figure 8: Experimental cell where all the experiments were conducted. See text for description.

Smoothing of the image is done before the computation of the spatial and temporal derivatives (always done with only two images) using a separable Gaussian filter whose discrete representation

is

$$[0.0223211 \;\; 0.229742 \;\; 0.495015 \;\; 0.229742 \;\; 0.0223211],$$

which corresponds to a σ of 0.75. The derivatives are calculated using a $4 \times 4 \times 2$ cell in the x, y and t directions respectively, as suggested by Horn (1987).

Numerous experiments were carried out using our implementation of the methods described. We only report results from representative experiments here. They were conducted indoors; a sample image can be seen in Figure 9. The translational motion was towards the floor with cluttered objects; the floor was not fronto-planar, but with an average angle of inclination in the range 45-70 degrees (note that it is 90 degrees for a frontal planar surface) between the floor surface and the optical axis. For all the following experiments, the amplitude of the translational motion was equal to 1.5 cm/s and the distance between the camera and the floor varied from 3 m to 0.75 m (which approximatively leads to a time-to-collision from 200 to 50). Finally, at the initial camera position, the angle between its optical axis and the direction of translation was equal to 25 deg.

Figure 9: A typical image obtained during one of the control maneuvers.

6.2.1 The FOE method

The FOE method described in section 4 has been implemented. The FOE is calculated as described in section 4.1. The control loop consists of the following steps which are repeated:

- obtain two successive images,
- compute the FOE location,
- compute the rotational velocity control required using the control law in Equation 8, and
- apply the control rotational velocity for a finite duration.

Note that the control is applied for only a finite duration during each iteration. This is because the qualitative method used for the FOE computation works only for pure translation. The total time spent in one iteration is 800 ms.

The result of a typical experiment is shown in Figure 10. The plot shows the variation of the angle between the direction of translation T and the optical axis Z with respect to the iteration number of the control law. As expected, the angle decreases and converges to zero. The final error plotted in Figure 10 is 0.2 degrees. In this experiment, the duration of the applied control was set to 200 ms and the value of λ to 0.35. These two parameters the product of which determines the speed of convergence have been tuned in order to preserve the stability of the system while minimizing the speed of convergence.

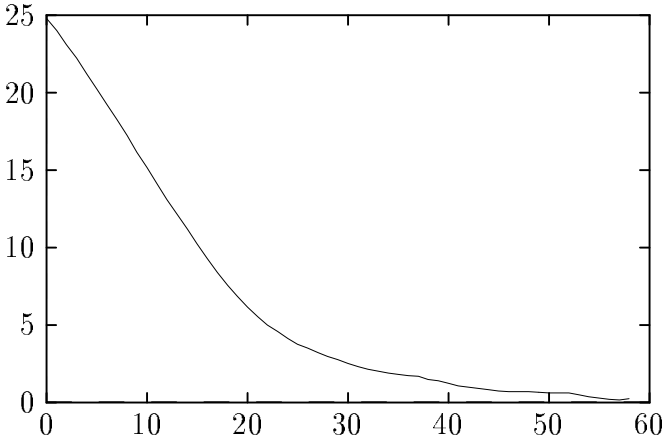


Figure 10: The angular error plot for the FOE method.

6.2.2 The Affine Parameters method

The affine parameters method described in section 5 has also been implemented. The affine parameters are computed using an over-constrained set of equations by considering the intensity derivatives from all over the image, thresholded by gradient magnitude to suppress contribution from relatively uniform regions where estimates are noisy.

The robot, as before, is commanded to move the camera with a certain translational velocity. The control loop consists of the following steps which are repeated:

- Obtain two successive images,
- compute the affine parameters of the flow field,
- compute the rotational velocity control required using the control law in Equation 21, and
- apply the control rotational velocity.

Here, two different programs, one in which the control is applied for a finite duration (again, 200 ms) and another in which the control is applied in a continuous manner, have been implemented. Each iteration took 800 ms for the first program and 400 ms for the second one.

The error plots from experiments using the two different implementations are shown in Figure 11. On the left is the plot (final error = 0.05 degrees) for the implementation where the control is applied

for a finite duration (discrete control), and on the right, for continuous control (final error = 0.15 degrees).

For the discrete control, λ was set to 0.7. A choice of a higher value than in the FOE method has been made possible since the affine parameters are computed more accurately using a global linear regression than the position of the FOE. This allows the system to obtain a more precise and rapid convergence, as well as a better stability.

Furthermore, for the discrete control, while the two successive images are acquired, the control is withdrawn; this means that the affine parameters a_1 and a_4 provide only the translational velocity information, and hence the control computation is accurate. On the other hand, for the continuous control, a_1 and a_4 contain the rotational velocity terms in addition to the *observation*, namely the translational velocity terms (which is the information we need). As an effect of the inaccuracies introduced in the observations, the continuous control converges more slowly and less precisely. An iteration takes however half as much time, leading to a total time to convergence nearly equal for the two methods.

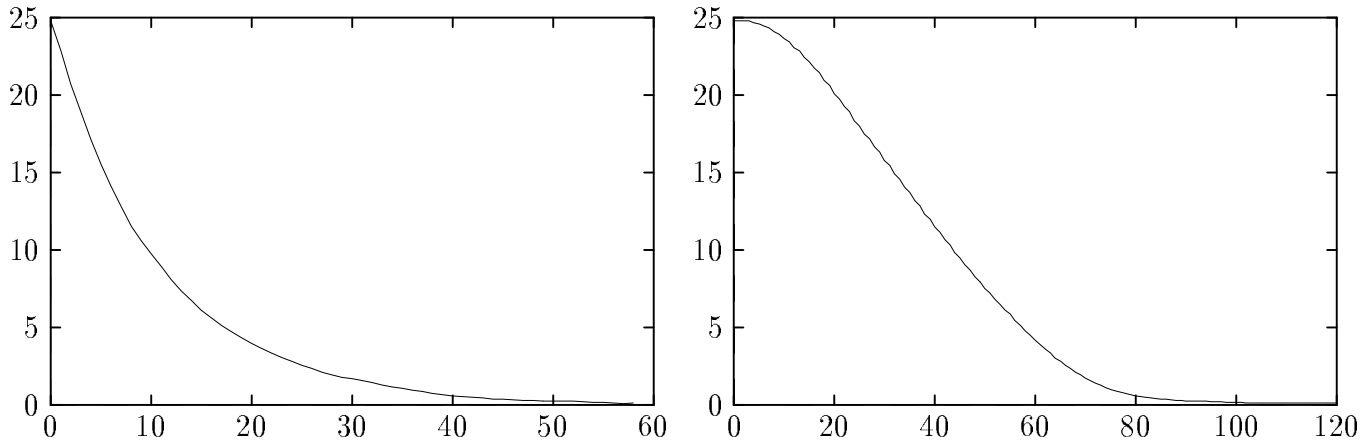


Figure 11: The angular error plots for the affine parameter method; on the left is the plot for the method using discrete control, and on the right is the plot for the method using continuous control.

We finally note that the different assumptions made in deriving the control laws (frontal planar surface, $\frac{1}{\tau_c} \approx -\frac{\dot{Z}_0}{Z_0} \approx \frac{a_2+a_6}{2}$, etc) do not have any noticeable influence in the realization of the task.

7 Discussion

In this paper, we have proposed an approach to use image motion information in an active visual task.

Barth and Tsuji (1993) also describe a method for achieving alignment of the optical axis with the translational direction. The method relies on the ability of the camera to fixate at a scene point, and based on a simple analysis of the optic flow field near the fixation point, saccades are made to eventually achieve alignment. They present a theoretical justification for their method, and show simulation and real experiments. Their scheme differs from ours in that they require the camera to

be able to track a 3D point (fixation), and the saccade is calculated in a qualitative fashion. Also, their scheme requires the computation of optic flow; even though their method does not require an accurate flow field, they do not indicate how this can be exploited.

We have not examined the possibility of predicting motion. Standard methods such as Kalman filtering could be employed to predict the FOE or the affine motion parameters; this would increase the stability of the methods.

The methods proposed here require the following camera calibration information: the *center* of the image (where the optical axis of the camera intersects the image plane) and the interpixel distance in both x and y directions. These parameters are used in the control equations.

In the affine parameter method, we assume that the affine approximation to the optical flow field is valid. This is supported by several useful methods developed based on the affine approximation (Bouthemy and François 1993, Negahdaripour and Lee 1992). Nevertheless, this approximation can fail for the entire image when there are objects located at very different depth in the scene, or moving objects of significant size. Motion-based segmentation of the image into regions (Bouthemy and François 1993) could be one possible solution, but far too complex to be implemented in such a closed-loop procedure. However, recently designed multi-resolution robust estimation methods such as (Odobez and Bouthemy 1994) can cope up with these situations.

We have described elsewhere an application for the alignment task (Bouthemy and Sundareswaran, 1993). This application explored a coupling of qualitative vision methods and active vision methods, and we believe the coupling is an interesting direction that needs to be explored. We outlined this idea by considering a qualitative method that can detect moving objects seen by a camera that itself is under motion (Bouthemy and François 1993). The qualitative method under consideration works well if the motion of the camera is, for example, along its optical axis. To improve this solution, we do not attempt to find a general solution to extend the method, but to use system capabilities to circumvent the restrictions in order to be able to still use the same method that is simple enough and known to be robust and efficient. A promising approach consists in designing active schemes in order to dynamically place the camera in some known tractable configuration and then to use simple interpretation methods. This is precisely a goal of the active visual task described here. The proposed coupling of a qualitative method and an active visual method was validated by a real example in which a mobile camera detects a moving object. In part, it was also a demonstration to show that the servoing methods described here work not only in the case of a static environment, but also when there are small moving objects in the scene.

8 Conclusions

In this paper, we have proposed the use of motion information in the visual servoing framework where only geometric information has been used so far. Two control schemes, one using the focus of expansion, and the other using affine motion parameters, were presented. Experimental results from a camera mounted on a robot serve to validate our proposal. A control method presented here has been used successfully to provide an interesting new direction of coupling between qualitative and active visual methods. Future work includes processing on the image-processing board to improve

speed and performance, and to investigate other forms of tight coupling between camera behavior and motion information.

We believe that the work presented here provides a starting point for formal approaches to closed-loop control using motion information; these are expected to be useful for active visual tasks involving a camera undergoing motion and/or is monitoring moving objects.

Acknowledgements

The first author was supported by an INRIA post-doctoral fellowship during the period of this work, and by a grant from the Office of Naval Research (# N00014-93-1-0381) to Lucia M. Vaina at Boston University during the preparation of this paper.

APPENDIX

Consistency in component directions of normal and optical flow vectors

Consider the optical flow vector $\vec{v}$ at a point (x, y) on the image plane. Let the local image gradient at (x, y) be along the unit normal vector $\vec{n}$, and $\vec{e_x}$ and $\vec{e_y}$ be the unit vectors along the x and y axes respectively. Then the normal flow is $\vec{v_n} = (\vec{v} \cdot \vec{n})\vec{n}$. If $\vec{v} \cdot \vec{e_x} > 0$, then $\vec{v_n} \cdot \vec{e_x} > 0$ with high probability; similarly for the case $\vec{v} \cdot \vec{e_x} < 0$, and for $\vec{v} \cdot \vec{e_y}$. The situation is described pictorially in Figure 12. In the figure, without loss of generality, we have chosen the flow vector $\vec{v}$ in one of the quadrants. If the angle subtended by $\vec{v}$ with the x axis is Φ, then the probability that the sign of $\vec{v_n} \cdot \vec{n}$ agrees with the sign of $\vec{v} \cdot \vec{e_x}$ is given by

$$\frac{\pi - \Phi}{\pi},$$

assuming a uniform distribution of the direction of $\vec{n}$. In other words, the signs will disagree if the normal vector is found in the shaded region of Figure 12, and this happens with probability Φ/π. At the same time, the probability that the sign of $\vec{v_n} \cdot \vec{e_y}$ agrees with the sign of $\vec{v} \cdot \vec{e_y}$ is

$$\frac{\frac{\pi}{2} + \Phi}{\pi}.$$

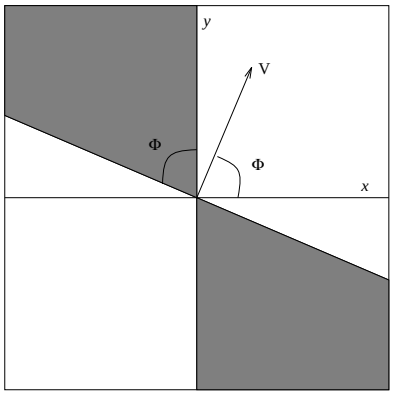


Figure 12: Illustration for the agreement of the signs of $\vec{v} \cdot \vec{e_x}$ and $\vec{v_n} \cdot e_x$.

Since $\Phi \leq \pi/2$, the probability of being correct is more than half (in fact, if one of them is close to half, the other one will be close to one, which means that if the sign of one component (x, say) being correct is close to "chance," then the sign of the other component (y) will be correct with a probability close to 1). For clarity of explanation, we have omitted the degenerate cases $\Phi = 0$ and $\Phi = \pi/2$.

References

[1] Allen, P.K., et al. 1993. Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Trans. on Robotics and Automation*, 9(2):152–165.

[2] Aloimonos, Y., and Duric, Z. 1992. Active egomotion estimation: A qualitative approach. *Lecture Notes in Computer Science 588: Computer Vision–ECCV '92*, ed. G. Sandini. Santa Margherita: Springer-Verlag, pp. 497–510.

[3] Aloimonos, Y., Weiss, I., and Bandyopadhyay, A. 1988. Active vision. *Int. J. Comp. Vision*, 1:333–356.

[4] Bajcsy, R., and Campos, M. 1992. Active and exploratory perception. *CVGIP: Image Understanding*, 56(1):31–40.

[5] Ballard, D.H., and Brown, C.M. 1992. Principles of animate vision. *CVGIP: Image Understanding*, 56(1):3–21.

[6] Barth, M.J., and Tsuji, S. 1993. Egomotion determination through an intelligent gaze control strategy. *IEEE Trans. on Sys. Man and Cyb.*, 23(5):1424–1432.

[7] Bergen, J.R., et al. 1992. Hierarchical model-based motion estimation. *Lecture Notes in Computer Science 588: Computer Vision–ECCV '92*, ed. G. Sandini. Santa Margherita: Springer-Verlag, pp. 237–252.

[8] Bouthemy, P. and François, E. 1993. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *Int. J. Comp. Vision*, 10(2):157–182.

[9] Bouthemy, P. and Sundareswaran, V. 1993. Qualitative motion detection with a mobile and active camera. *Proc. Intl. Conf. on Digital Signal Processing*, ed. A. G. Constantinides. Cyprus: pp. 444–449.

[10] Chaumette, F., et al. 1994. Optimal estimation of 3d structures using visual servoing. *Proc. CVPR*, Seattle: IEEE, pp. 347–354.

[11] Chaumette, F., and Santos, A. 1993. Tracking a moving object by visual servoing. *Proc. 12th world congress IFAC, Vol. 4*, Sydney: Pergamon, pp. 409–414.

[12] Cipolla, R., and Blake, A. Surface orientation and time to contact from image divergence and deformation. *Lecture Notes in Computer Science 588: Computer Vision–ECCV '92*, ed. G. Sandini. Santa Margherita: Springer-Verlag, pp. 187–202.

[13] Coombs, D., and Brown, C. 1993. Real-time binocular smooth pursuit. *Int. J. Comp. Vision*, 11(2):147–164.

[14] Coombs, D., and Roberts, K. 1993. Centering behavior using peripheral vision. In *Proc. CVPR*, New York: IEEE, pp. 440–445.

[15] Espiau, B., Chaumette, F., and Rives, P. 1992. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326.

[16] Feddema, J.T., and Mitchell, O.R. 1989. Vision-guided servoing with feature-based trajectory generation. *IEEE Trans. on Robotics and Automation*, 5(5):691–700.

[17] Grosso, E., and Ballard, D.H. 1993. Head-centered orientation strategies in animate vision. *Proc. Fourth ICCV*, Berlin: IEEE Computer Society Press, pp. 395–402.

[18] Grosso, E., Tistarelli, M., and Sandini, G. 1992. Active/dynamic stereo for navigation. *Lecture Notes in Computer Science 588: Computer Vision–ECCV '92*, ed. G. Sandini. Santa Margherita: Springer-Verlag, pp. 516–525.

[19] Hashimoto, K. ed. 1993. *Visual Servoing*, volume 7 of World Scientific Series in Robotics and Automated Systems. Singapore: World Scientific.

[20] Horn, B.K.P. 1987. *Robot Vision.* Cambridge: The MIT Press.

[21] Koenderink, J.J., and van Doorn, A.J. 1975. Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer. *Optica Acta*, 22(9):773–791.

[22] Krotkov, E. 1987. Focusing. *Int. J. Comp. Vision*, 1:223–237.

[23] Murray, D.W., et al. 1993. Reactions to peripheral image motion using a head/eye platform. *Proc. Fourth ICCV*, Berlin: IEEE Computer Society Press, pp. 403–411.

[24] Negahdaripour, S. and Lee, S. 1992. Motion recovery from image sequences using only first order optical flow information. *Int. J. Comp. Vision*, 9(3):163–184.

[25] Nelson, C., and Aloimonos, J. 1989. Obstacle avoidance using flow field divergence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11:1102–1106.

[26] Odobez, J-M., and Bouthemy, P. 1994. Robust multiresolution estimation of parametric motion models in complex image sequences. *Proc. 7th Conf. Eusipco*, Edinburgh: pp. 411-414.

[27] Olson, T.J., and Coombs, D.J. 1991. Real-time vergence control for binocular robots. *Int. J. Comp. Vision*, 7(1):67–89.

[28] Pahlavan, K., Uhlin, T., and Eklundh, J-O. 1993. Dynamic fixation. *Proc. Fourth ICCV*, Berlin: IEEE Computer Society Press, pp. 412–419.

[29] Papanikolopoulos, N., Khosla, P., and Kanade, T. 1993. Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision. *IEEE Trans. on Robotics and Automation*, 9(1):14–35.

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

[31] Santos-Victor, J., et al. 1993. Divergent stereo for robot navigation: learning from bees. In *Proc. CVPR*, New York: IEEE, pp. 434–439.

[32] Sundareswaran, V., Bouthemy, P., and Chaumette, F. 1994a. Active camera alignment using dynamic image parameters. *Lecture Notes in Computer Science 801: Computer Vision –ECCV '94*, ed. J-O. Eklundh. Stockholm: Springer-Verlag, pp. 111–116.

[33] Sundareswaran, V., Bouthemy, P., and Chaumette, F. 1994b. Visual servoing using dynamic image parameters. Research report 2336. Rennes, France: INRIA.

[34] Sundareswaran, V., Chaumette, F., and Bouthemy, P. 1994c. Visual servoing using image motion information. *Proc. IAPR/ IEEE workshop on visual behaviors*, ed. W. Martin, Seattle: IEEE, pp 102–106.

[35] Weiss, L.E., Sanderson, A.C., and Neuman, C.P. 1987. Dynamic sensor-based control of robots with visual feedback. *IEEE Trans. on Robotics and Automation*, 3(5):404–417.

[36] Wijesoma, S.W., Wolfe, D.F.H., and Richards, R.J. 1993. Eye-to-hand coordination for vision guided robot. *Int. J. Robotics Res.*, 12(1):65–78.

[37] Yuille, A. and Geiger, D. 1990. Stereo and controlled movement. *Int. J. Comp. Vision*, 4:141–152.