Structure From Controlled Motion

François Chaumette, Samia Boukir, Patrick Bouthemy, and Didier Juvin

Abstract—This paper deals with the recovery of 3D information using a single mobile camera in the context of active vision. First, we propose a general revisited formulation of the structure-from-known-motion issue. Within the same formalism, we handle various kinds of 3D geometrical primitives such as points, lines, cylinders, spheres, etc. We also aim at minimizing effects of the different measurement errors which are involved in such a process. More precisely, we mathematically determine optimal camera configurations and motions which lead to a robust and accurate estimation of the 3D structure parameters. We apply the visual servoing approach to perform these camera motions using a control law in closed-loop with respect to visual data. Real-time experiments dealing with 3D structure estimation of points and cylinders are reported. They demonstrate that this active vision strategy can very significantly improve the estimation accuracy.

Index Terms—Computer vision, robotics, active vision, structure from motion, vision-based control.

1 Introduction

RECOVERING 3D structure from images is one of the main issues in computer vision. Among others, like a stereovision approach [16], or the use of projective invariants [33], an appealing way of solving this problem is to use 2D motion computed in image sequences acquired by a monocular camera. However, to get an accurate three-dimensional geometric description of a scene from such 2D usually noisy measurements is not an easy task. It requires the development of efficient algorithms for the analysis and the interpretation of 2D dynamic visual information.

Basically, two main approaches have been investigated to solve the problem of structure from motion : long range motion-based methods and short range motion-based ones. In the former one, images are considered at distant time instants and a large camera displacement is generally performed to obtain accurate results. This approach is based on the extraction of a set of relatively sparse, distinguishable two-dimensional features in the successive images [11], [16], [25], [34], [38], [42]. Inter-frame correspondence is first established between these features. Then, the 3D structure is determined requiring a measure of the camera displacement, expressed by a translation and a rotation matrix. In the latter approach, images are considered at video rate. In this case, emphasis is generally placed on the estimation of the optic flow field between two successive frames [2], [37], [40] or on the direct use of the spatio-temporal derivatives of the image brightness [26]. These observations must also be combined with a measure of the camera velocity (instead of camera displacement) to determine the 3D structure of the scene. In that case, equations for structure estimation become linear. However, a robust and accurate computation of feature correspondences or optic flow fields remains a difficult problem requiring the development of sophisticated time consuming algorithms. Furthermore, the optic flow field and the spatio-temporal derivatives of the image brightness are corrupted by noise, and often partially incorrect near occlusion or motion boundaries, leading to generally poor and unstable results in the 3D reconstruction, in particular when considering object limbs.

To alleviate these problems, hybrid approaches have been proposed [15], [30]. They are based on the utilization of the camera velocity in the 3D reconstruction process, while relying on the tracking of 2D sparse image features. Such methods avoid the intermediate stage of optic flow computation, and involve simple matching process. Furthermore, we will see in this paper that the formulation, based on the use of continuous camera velocities instead of camera displacements, allows us to determine simple closed-form expressions for the 3D structure estimation of geometrical primitives such as circles, spheres and cylinders. However, they still suffer from sensitivity to noise and unsatisfactory accuracy.

A solution to cope with the robustness problems consists in integrating the data obtained from multiple successive camera positions. The methods employed are generally based on Kalman filtering [16], [22]. They enable to estimate both structure and motion (up to a scale factor) if the 3D relative velocity between the scene and the camera is assumed constant over time [27]. An original factorization method based on the singular value decomposition has also been proposed by Tomasi and Kanade [36]. This method does not require a constant 3D relative velocity, but is valid only if an orthographic projection model can be considered.

Nevertheless, all these approaches (long range motion-based methods as well as short range motion-based methods) do not involve any strategy to select motions and trajectories that the camera should undergo (at least, it is essential that objects of interest remain in the field of view of the camera). Hence, an alternative solution that we investigate in this paper, is to follow an active vision

- *F. Chaumette and P. Bouthemy are with IRISA/INRIA Rennes, Campus de Beaulieu, 35042 Rennes-cedex, France. E-mail: {chaumett; bouthemy}@irisa.fr.*
- *S. Boukir is with L3i, Avenue de Marillac, 17042 La Rochelle, -cedex 1,France. E-mail: sboukir@gi.univ-lr.fr.*
- *D. Juvin is with CEA LETI DEIN, 91191 Gif sur Yvette-cedex, France. E-mail: djuvin@chouette.saclay.cea.fr.*

Manuscript received June 8, 1994; revised Dec. 7, 1995. Recommended for acceptance by A. Singh.
For information on obtaining reprints of this article, please send e-mail to: transactions@computer.org, and reference IEEECS Log Number P96005.

approach. Active vision, which can be defined as an intelligent data acquisition process, has been recently introduced by Bajcsy [4] and Aloimonos et al. [3], and has been extended by several researchers including Abbott and Ahuja [1], Ballard [5], Brown [9], and Sandini and Tistarelli [32]. Bajcsy [4] attempted to explore the environment by tuning parameters involved in the processing steps and sensor parameters in order to obtain the best fit between model and data. Aloimonos et al. [3], [18], [20] discussed active vision from a mathematical point of view, focusing on complex problems such as stability, linearity and uniqueness of solutions. They have shown that some vision problems, that are ill-posed and unstable for a passive observer, become well-posed and stable for an active one. Furthermore, active vision can greatly facilitate the computational process: quantities that are complex or time-consuming to compute with a passive system, are much more easily computed with an active one [5].

Research work going on in several groups aims at developing active vision systems with great visual abilities such as control of ocular parameters (e.g. aperture, focus) [1], spatially-varying sensing [12], [28], [35], and gaze control [1], [5], [21], [24], [28], [32], [39]. Indeed, one of the most important requirements in the recovery of depth information is the robustness to the noisy nature of real data. The control of the viewing parameters gives stable and robust means for shape and motion perception. More specifically, the control of ocular parameters allows the system to maintain a suitable image quality despite the degradations that may occur during the acquisition process. The control of gaze is vastly used in binocular camera heads [5], [24], [28], [39]. This mechanism, called *vergence*, consists to bring and maintain the two camera axes to a specified spatial target position: the *fixation point*. This permits to get rid of the inter-frame correspondence problem in the 3D visual perception process [4], [9].

Our concern is to deal with the problem of recovering the 3D spatial structure of a static scene using a single mobile camera by means of an active vision scheme [10]. Within the active vision paradigm, we consider that the camera motion can be controlled. This implies that measurements of the camera velocity from the successive robot joint positions are available. However, they are usually noisy. Therefore, the handling of measurement errors on the camera motion is explicitly incorporated in our scheme. We show in this paper that 3D reconstruction can be solved in a much more efficient way using adequate controlled camera motions, compared to usual dynamic vision approaches. This problem has already motivated some investigations [3], [6], [31], [32], but only for the case of points. Furthermore, effective comparisons between dynamic vision and active vision schemes have not yet been performed through real experiments. In this paper, the problem of 3D reconstruction of a large class of geometrical primitives is addressed at three levels:

- **modeling aspect:** we propose a general revisited formulation of the structure-from-known-motion issue which is valid for the most representative geometrical primitives. Our approach is based on the so-called interaction matrix [14], which relates 2D motion of the projection of the primitive in the image with 3D camera motion, and which enables to determine the 3D position of the limb surface of the primitive.
- **optimization aspect:** we derive sufficient conditions to minimize effects of the different measurement errors which may occur in this process. More precisely, we determine the adequate camera viewpoints and motions which provide a robust and accurate estimation of the considered 3D structure. We apply the visual servoing framework to perform such motions using a control law in closed-loop with respect to visual data.
- **experimental aspect:** we demonstrate with various real-time experiments that our active vision scheme significantly improves the accuracy of the structure estimation.

This paper is organized as follows. In Section 2, we present a general method for the estimation of the 3D structure of geometrical primitives using dynamic vision. Section 3 contains the derivation of the types of camera motions which improve the robustness of the solutions, and describes a vision-based control scheme able to generate such motions. Experimental results demonstrating the efficiency of our method are reported in Section 4. Concluding remarks are given in Section 5.

2 Structure From Motion Using Dynamic Vision

Let us first recall how we can determine the depth of a point in the scene using a moving camera. Let us consider a usual perspective projection model (see Fig. 1). Without loss of generality, the focal length is assumed to be equal to 1. The relation between the 3D point $\underline{X} = (X, Y, Z)$ and its projection $\underline{x} = (x, y, 1)$ on the image plane is given by:

$$\underline{x} = \frac{1}{Z}\,\underline{X}. \tag{1}$$

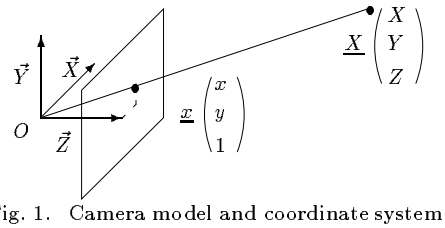


Fig. 1. Camera model and coordinate system.

The velocity screw of the camera frame $(O, \vec{X}, \vec{Y}, \vec{Z})$, plotted in Fig. 1, with respect to the scene is denoted by $T = (V(O), \Omega)$, where $V(O) = (V_X, V_Y, V_Z)$ and $\Omega = (\Omega_X, \Omega_Y, \Omega_Z)$ are respectively the instantaneous translational and rotational velocities. If scene point $\underline{X}$ is static, we get:

$$\dot{\underline{X}} = -V(O) - \Omega \times \underline{X}. \tag{2}$$

Differentiating (1) and using (2) lead to the well known

relations [19]:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} -V_X + xV_Z \\ -V_Y + yV_Z \end{pmatrix} + \begin{pmatrix} \alpha_x \\ \alpha_y \end{pmatrix}, \qquad (3)$$

with $\alpha_x = xy\Omega_X - (1+x^2)\Omega_Y + y\Omega_Z$ and $\alpha_y = (1+y^2)\Omega_X - xy\Omega_Y - x\Omega_Z$. From (3), we can easily derive the expression of the unknown depth Z. We obtain [22]:

$$1/Z = \frac{(\dot{x} - \alpha_x)(xV_Z - V_X) + (\dot{y} - \alpha_y)(yV_Z - V_Y)}{(xV_Z - V_X)^2 + (yV_Z - V_Y)^2} \qquad (4)$$

An estimation of depth Z can thus be computed from the measured values of $x, y, \dot{x}, \dot{y}$ and camera velocity T. Let us note that no information on Z can be retrieved if the camera motion is such that $V_X = xV_Z$ and $V_Y = yV_Z$.

We now introduce the unified formalism capable to handle the case of points, but also planar and volumetric parametric primitives. Let us consider a 3D geometrical primitive $\mathcal{P}_S$ of the scene; its configuration is specified by an equation of the type:

$$h(\underline{X}, \underline{P}) = 0 \ , \quad \forall \underline{X} \in \mathcal{P}_S \qquad (5)$$

where h defines the type of the considered primitive, and the value of parameters $\underline{P}$ specifies its corresponding configuration. The representation of $\mathcal{P}_S$, expressed by parameters $\underline{P}$ (vector of dimension n), is chosen complete and minimal in order that any position of the primitive can be represented by only one value of $\underline{P}$.

Using (1), (5) becomes:

$$h'(\underline{x}, 1/Z, \underline{P}) = 0. \qquad (6)$$

where h' denotes the expression of the resulting equation. Under the trivial condition $\frac{\partial h'}{\partial Z} \neq 0$ which is satisfied in all the non-degenerate cases (a degenerate case occurs for example when a line is projected onto the image plane as a point, or a circle as a segment), the implicit function theorem ensures the existence of a unique function μ such that:

$$1/Z = \mu(\underline{x}, \underline{P}_L) \qquad (7)$$

where representation $\underline{P}_L$, function of $\underline{P}$, is chosen complete and minimal (its dimension is denoted n_L).

Let us denote $\mathcal{P}_i$ the projection in the image plane of $\mathcal{P}_S$. The configuration of $\mathcal{P}_i$ can be formally written as follows:

$$g(\underline{x}, \underline{p}) = 0 \ , \quad \forall \underline{x} \in \mathcal{P}_i \qquad (8)$$

where g defines the type of the 2D image primitive and the value of $\underline{p}$, function of $\underline{P}$, specifies its configuration. Once again, representation $\underline{p}$, of dimension m, is chosen complete and minimal in order that any position in the image of $\mathcal{P}_i$ can be represented by only one value of $\underline{p}$. For planar primitives (a circle for example), the function μ represents the plane in which the primitive lies. For volumetric primitives (sphere, cylinder, torus,...), the function $g(\underline{x}, \underline{p})$ represents the contour of $\mathcal{P}_i$, and the set of 3D points, the projection of which constitutes the set of 2D contour points, defines

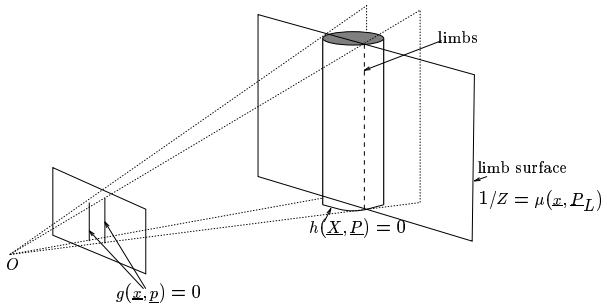


Fig. 2. Perspective projection (g) of a cylinder (h) in the image and corresponding limb surface (μ).

the function $\mu(\underline{x}, \underline{P}_L)$, therefore called the **limb surface** (see Fig. 2).

The relation which links the motion of the primitive in the image (in fact, the temporal derivative $\dot{\underline{p}}$ of parameters $\underline{p}$), and camera motion T can be explicitly derived [14]; it is given by:

$$\dot{\underline{p}} = L_{\underline{p}}^T(\underline{p}, \underline{P}_L)\ T \qquad (9)$$

where $L_{\underline{p}}^T$, called the **interaction matrix** related to $\underline{p}$, fully characterizes the interaction between the camera and the considered primitive.

We are now able to present a general method to reconstruct a 3D geometrical primitive by dynamic vision (i.e., to compute the value of $\underline{P}$ from the measurements of the camera velocity T, and of the image parameters $\underline{p}$ and $\dot{\underline{p}}$).

Let us denote $\mathcal{H}(\underline{p}, \dot{\underline{p}}, \underline{P}_L, T)$ the following function derived from (9):

$$\mathcal{H}(\underline{p}, \dot{\underline{p}}, \underline{P}_L, T) = \dot{\underline{p}} - L_{\underline{p}}^T(\underline{p}, \underline{P}_L)\ T = 0. \qquad (10)$$

Under the condition that $\dfrac{\partial \mathcal{H}}{\partial \underline{P}_L}$ (of dimension $m \times n_L$) is of full rank n_L, the implicit function theorem allows us to express $\underline{P}_L$ with respect to the other parameters involved in (10). Since the dimension n_L of $\underline{P}_L$ has been chosen minimal, we obtain an unique solution:

$$\underline{P}_L = \underline{P}_L\ (T, \underline{p}, \dot{\underline{p}}). \qquad (11)$$

More precisely, for all the primitives that we have considered (straight lines, circles, spheres and cylinders - see the Appendix at the end of this paper), parameters $\underline{P}_L$ are simply determined from the resolution of a linear system derived from (10).

Let us note that it is possible to find the camera motions which do not provide any information on the spatial configuration of the primitive of interest: they are such that $\dfrac{\partial \mathcal{H}}{\partial \underline{P}_L}$ is not of full rank. For example, in the case of a point, we have

$$\frac{\partial \mathcal{H}}{\partial \underline{P}_L} = \begin{pmatrix} V_X - xV_Z \\ V_Y - yV_Z \end{pmatrix}$$

(see (3)), which is of full rank 1 if $V_X \neq xV_Z$ or $V_Y \neq yV_Z$.

Finally, knowing $g(\underline{x}, \underline{p})$ and $\mu(\underline{x}, \underline{P}_L)$, we can solve for parameters $\underline{P}$ which completely define the configuration

of the considered 3D primitive. Since the dimension n of $\underline{P}$ has been chosen minimal, we again obtain an unique solution:

$$\underline{P} = \underline{P}\,(\underline{p}, \underline{P}_L). \tag{12}$$

Let us note that our method is based on a continuous approach, since it uses the measure of the camera velocity. It is basically different from the discrete ones [11], [25], [38], [42] which consider a camera displacement (described by a translation and a rotation matrix) instead of camera velocity. Indeed, from a geometrical point of view (see Fig. 3), our method consists in determining the intersection between a generalized cone (defined by its vertex O and the function $g(\underline{x}, \underline{p})$) and the limb surface (derived as explained above from motion equation (9)). On the other hand, the discrete approach, equivalent to a stereovision paradigm, is based on the intersection between two generalized cones (corresponding to each camera position). If we consider the intersection of two volumes, instead of a volume and a surface, then, it is more complicated to derive closed-form expressions (and therefore to get robust estimations) in the case of complex primitives. For a circle for example, the discrete method proposed in [29] is based on the resolution of a complex non linear system, whereas our method is based on the simple resolution of two linear systems.

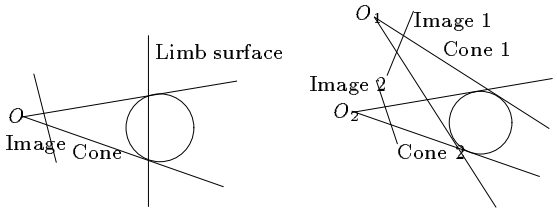


Fig. 3. Difference between continuous (left) and discrete (right) approaches for 3D structure estimation.

The estimation of the position (X_0, Y_0, Z_0) of a point, which has been described at the beginning of this section, can of course be obtained using this formalism. Indeed, in that straightforward case, we have:

$$h(\underline{X}, \underline{P}) = \begin{cases} X - X_0 = 0 \\ Y - Y_0 = 0 \\ Z - Z_0 = 0 \end{cases}$$

$$g(\underline{x}, \underline{p}) = \begin{cases} x - x_0 = 0 \\ y - y_0 = 0 \end{cases},$$

$$1/Z = \mu(\underline{x}, \underline{P}_L) = 1/Z_0. \tag{13}$$

The depth Z_0 is obtained from the motion equation (3) (corresponding in fact to (9)); it is given by (4) (to be related to (11)). Then, we get $X_0 = x_0 Z_0$ and $Y_0 = y_0 Z_0$ (i.e., relation (12)).

The application of this method to the most representative geometrical primitives (straight lines, circles, spheres and cylinders) is described in the Appendix. For each of these primitives, we first present the equations of their respective function $h(\underline{X}, \underline{P})$, $g(\underline{x}, \underline{p})$ and $\mu(\underline{x}, \underline{P}_L)$, and of the related interaction matrix $L_{\underline{p}}^T(\underline{p}, \underline{P}_L)$. Then, we describe how parameters $\underline{P}_L$ can be determined from $\underline{p}$, $\dot{\underline{p}}$ and T. Finally, we present how parameters $\underline{P}$ are obtained from $\underline{p}$ and $\underline{P}_L$. The set of camera motions which cannot provide an estimation of the structure of the considered primitives is also given. As seen on all the considered examples, the closed-form expressions of the 3D parameters to be estimated are simply and systematically obtained from the resolution of two linear systems (the first one serves to determine the parameters of the limb surface using the interaction matrix, the second one to determine parameters $\underline{P}$).

The main limitation of our method is that it assumes the presence of rather simple objects in the scene. On the other hand, it seems more efficient to obtain a global analytical structure estimation of geometric primitives, described by vectors of parameters, than a discrete sparse one given by the 3D positions of some points or segments. Furthermore, our approach could be used for more complex parametric primitives, such as torus or ellipsoïd for example.

3 Structure From Motion Using Active Vision

In the previous section, it has been shown that the continuous approach exploiting camera velocity is an attractive solution to cope with the 3D reconstruction of a static scene. However, structure estimation from a monocular image sequence is very sensitive to the measurement errors [13] and to the nature of the camera movements [15]. The experimental results reported in the next section will confirm that important errors on the structure estimation appear when no particular strategy concerning camera motion is defined. We have also pointed out that some given motions are not able to provide any 3D-information. Therefore, one of the goal of an active vision scheme is to find an optimal camera motion which could lead to a robust and non biased estimation of the 3D spatial structure. In this section, we state the problem in terms of the minimization of the errors occurring in the reconstruction process. Two kinds of errors are of particular concern: the first one is related to the discretization step that affects our continuous method, the second one is due to the unavoidable measurement errors on the image data and on the camera motion.

3.1 Suppression of the Discretization Effects

The method presented in Section 2 is based on the measurement of $\dot{\underline{p}}$, i.e., the temporal derivative of the parameters representing the considered image primitive. The exact value of $\dot{\underline{p}}$ is generally unreachable and the image measurements only supply $\Delta\underline{p}$, the variation of $\underline{p}$ during the time interval Δt between two successive images. Using $\Delta\underline{p}/\Delta t$ instead of $\dot{\underline{p}}$ in the derivations mentioned above induces errors in the 3D reconstruction. These errors due to the discretization step can be important, as it will be seen in the experimental results.

On the other hand, if we can ensure that $\dot{\underline{p}} = \Delta\underline{p}/\Delta t, \forall t,$

the discretization step will have no effect. Such a condition is satisfied if and only if:

$$\ddot{\underline{p}} = \cdots = \underline{p}^{[n]} = 0, \forall t. \tag{14}$$

From (9), we have

$$\dot{\underline{p}} = L_{\underline{p}}^T(\underline{p}, \underline{P}_L)\, T = f(\underline{p}, \underline{P}_L, T).$$

Thus:

$$\ddot{\underline{p}} = L_{\underline{p}}^T(\underline{p}, \underline{P}_L)\, \dot{T} + \frac{\partial f}{\partial \underline{p}}\dot{\underline{p}} + \frac{\partial f}{\partial \underline{P}_L}\dot{\underline{P}}_L. \tag{15}$$

A sufficient and general condition to satisfy (14) is to constrain the camera motion to be such that:

$$\dot{\underline{p}} = \dot{\underline{P}}_L = 0, \forall t. \tag{16}$$

Indeed, in that case, we have $T \in$ Ker $L_{\underline{p}}^T, \forall t$. Using (16), we can easily show that $\dot{T} \in$ Ker $L_{\underline{p}}^T, \forall t$, from which we deduce $\ddot{\underline{p}} = 0, \forall t$. A simple recurrence can finally be employed to check that (14) is ensured.

In other words, a solution to suppress the discretization error is that the equation of the limb surface remains unchanged in the camera coordinate system, and that the projection of the 3D primitive constantly appears at the same position in the image while the camera is moving.

We can show that, except for points and lines, the first condition $\dot{\underline{p}} = 0$ implies the second one $\dot{\underline{P}}_L = 0$, which reduces the problem to a **fixation** situation. We will see in Section 3.3 that the visual servoing approach [14], [17], [41], is perfectly suited to generate such camera motions. It enables the definition of control laws in closed-loop with respect to visual data, which automatically compute the camera motion satisfying particular constraints, such as $\dot{\underline{p}} = 0$ and $\dot{\underline{P}}_L = 0$.

Let us note that the condition that we have exhibited is only sufficient, and not necessary. Indeed, camera motion exists such that $\ddot{\underline{p}} = 0$ with $\dot{\underline{p}} \neq 0$. For a point for example, we can easily show, by differentiating (3), that $\ddot{\underline{p}} = 0$ when the camera motion is a pure translation parallel to the image plane performed at constant velocity, i.e., $V_X = V_1, V_Y = V_2$ and $V_Z = \Omega_X = \Omega_Y = \Omega_Z = 0$. More generally, determining all the solutions of the non linear system (14) seems out of reach. Moreover, they deeply depend on the considered primitive since they require the knowledge of $\frac{\partial f}{\partial \underline{p}}$ and $\frac{\partial f}{\partial \underline{P}_L}$. On the other hand, condition (16) is valid for any kind of primitives. It has the supplementary advantage that the primitive will remain in the field of view of the camera during the estimation process.

3.2 Minimizing Effects of the Measurement Rrrors

Another important point in an active vision paradigm is to select configurations of the camera with respect to the primitive of interest, likely to provide an estimation as robust as possible. More precisely, we show in this section that the effects of the measurement errors on the estimation of the 3D spatial structure of a primitive depend on the position of the primitive in the image. Therefore, we propose to constrain the camera motion in order to **focus** on the primitive, so that the projection of the primitive to be reconstructed is located at the position in the image that minimizes the effects of the measurement errors.

Let P designate one component of $\underline{P}$, the parameter vector representing the considered primitive. Let us recall that P depends on information extracted from the image $(\underline{p}, \dot{\underline{p}})$ and on the measured camera velocity T. If we suppose that the measurement errors on $\underline{p}, \dot{\underline{p}}$ and T are not correlated, the uncertainty σ_P on the estimation of P can be written in the form:

$$\begin{aligned}(\sigma_P)^2 &= \sum_{i=1}^{m}\left(\frac{\partial P}{\partial p_i}\right)^2(\sigma_{p_i})^2 + \sum_{j=1}^{m}\left(\frac{\partial P}{\partial \dot{p}_j}\right)^2(\sigma_{\dot{p}_j})^2 \\ &+ \sum_{k=1}^{6}\left(\frac{\partial P}{\partial T_k}\right)^2(\sigma_{T_k})^2.\end{aligned} \tag{17}$$

where $p_{i(i=1..m)}$, $\dot{p}_{j(j=1..m)}$, and $T_{k(k=1..6)}$ are, respectively, the different components of $\underline{p}$, $\dot{\underline{p}}$ and T. Minimizing σ_P is equivalent to minimizing each term $\left(\frac{\partial P}{\partial a}\right)^2$ where a denotes any of the variables $p_i, \dot{p}_j$ and T_k. Since we are interested in determining the optimal positions of the primitive in the image, we have to find the values of $\underline{p}$ such that:

$$\left(\frac{\partial\left(\frac{\partial P}{\partial a}\right)^2}{\partial p_j}\right) = 0\ , \quad \forall a \text{ and } \forall j = 1 \text{ to } m. \tag{18}$$

To find all the solutions of this non linear system in an analytical way seems unreachable. However, we have derived the following particular solutions of interest:

- for a point, the effects of the measurement errors on the estimation of its depth Z are minimized
  (i.e., $\frac{\partial\left(\frac{\partial Z}{\partial a}\right)^2}{\partial x} = \frac{\partial\left(\frac{\partial Z}{\partial a}\right)^2}{\partial y} = 0$,
  $\forall a \in \{x, y, \dot{x}, \dot{y}, V_X, V_Y, V_Z, \Omega_X, \Omega_Y, \Omega_Z\}$),
  if the point constantly appears at the center of the image $(x = \dot{x} = y = \dot{y} = 0, \forall t)$ during the time interval required for the estimation, and if we have $V_Z = \Omega_Z = 0$. In other words, the camera must be displaced on a sphere the center of which is the point to be reconstructed. It is interesting to notice that we get conclusions similar as those obtained in [6], [12] and [32], where the interest of locating the fixation point of an active binocular head in the center of the image is demonstrated.
- for a sphere, the effects of the measurement errors are minimized, if the image of the sphere remains a circle centered in the image and if $\Omega_Z = 0$. The optimal trajectory of the camera is thus the same as in the previous case.
- for a straight line, the effects of the measurement errors are minimized, if the line always appears centered and vertical (respectively horizontal) in the image, and if we also have $V_Y = V_Z = \Omega_X = 0$ (respectively $V_X = V_Z = \Omega_Y = 0$).

- for a cylinder, similarly, effects of the measurement errors are minimized, if the projections of its two limbs lie astride the image center in a symmetric manner, vertically, and if $V_Y = 0$ (or horizontally with $V_X = 0$). The camera must be displaced on a circle around the cylinder axis (see Fig. 4).

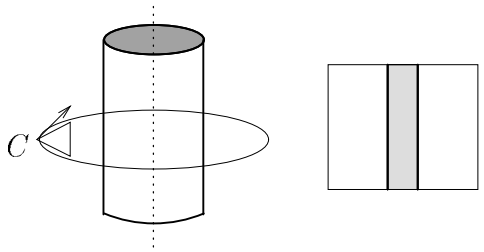


Fig. 4. Optimal camera motion and resulting image in the cylinder case.

Unfortunately, due to the complexity of the stated problem, we have not been able to prove that these solutions are unique. On the other hand, we have checked that numerous configurations really do not minimize all the $\left(\frac{\partial P}{\partial a}\right)^2$ terms, and thus, are not likely to provide a robust estimation [8].

We now describe how it is possible to automatically compute the camera motion satisfying the constraints described above.

3.3 Image-Based Closed-Loop Control

Active vision aims at improving the knowledge of the environment by means of adequate camera motions. A control law in closed-loop with respect to visual data, such as **visual servoing** [14], [17], [41], is perfectly suited to generate such motions. This approach is based on the regulation to zero of a task function $\underline{e}$ which can be written as follows [14]:

$$\underline{e} = W^+ W C\ (\underline{p} - \underline{p}^*) + (\mathbb{I}_6 - W^+ W)\underline{e}_2 \qquad (19)$$

where:

- $\underline{p}$ denotes the 2D parameters describing the projection in the image of the primitive at which the camera is fixating at or focusing on. $\underline{p}$ is measured at each iteration of the control law.
- $\underline{p}^*$ is the target value of $\underline{p}$ to be obtained. In order to suppress the discretization error, we have to satisfy $\dot{\underline{p}} = 0$. In this fixation task, $\underline{p}^*$ is set to the initial measured value of $\underline{p}$. In the focusing task, $\underline{p}$ has to reach a desired value to obtain a robust estimation ($x = y = 0$ for a point for example); in that case, $\underline{p}^*$ must then be equal to this particular value.
- C is a matrix which represents the inverse Jacobian of the vision-based task. Ideally, this matrix is chosen as the pseudo-inverse of the interaction matrix related to $\underline{p}$:

$$C = L_{\underline{p}}^{T+}\ (\underline{p}, \underline{P}_L). \qquad (20)$$

But, since the real value of $\underline{P}_L$ is unknown, we choose:

$$C = L_{\underline{p}}^{T+}\ (\underline{p}, \widehat{\underline{P}_L}) \qquad (21)$$

where $\widehat{\underline{P}_L}$ is the current estimation of the parameters of the limb surface obtained by the method described in Section 2.

- $\underline{e}_2$ is a secondary task which allows the camera to move along a desired trajectory (on a sphere or a circle for example). $\underline{e}_2$ also permits to satisfy the additional constraint $\dot{\underline{P}}_L = 0$ for the point and straight line cases.
- W^+W and $\mathbb{I}_6 - W^+W$ are two projection operators which guarantee that the camera motion due to the secondary task is compatible with the regulation of $\underline{p}$ to $\underline{p}^*$ ($\mathbb{I}_6$ is the 6×6 identity matrix and W is a full rank matrix such that Ker W = Ker $L_{\underline{p}}^T$. More details are given in [14]).

Once the task function $\underline{e}$ is defined, a simple control law, which computes camera velocity T and ensures an exponential decrease of $\underline{e}$, is given by [14]:

$$T = -\lambda\, \underline{e} - (\mathbb{I}_6 - W^+W)\frac{\partial \underline{e}_2}{\partial t} \qquad (22)$$

where λ (> 0) is the factor that controls the speed of the decay and where the term $(\mathbb{I}_6 - W^+W)\frac{\partial \underline{e}_2}{\partial t}$ is tied to the generation of a non zero camera motion when the vision-based task is realized (i.e., when $\underline{p} = \underline{p}^*$).

For example, in the case of the point reconstruction, we have (see (3)):

$$L_{\underline{P}}^T = \begin{pmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{pmatrix} \qquad (23)$$

from which we can easily compute $L_{\underline{p}}^{T+}$. Since $L_{\underline{p}}^T$ is a full rank matrix, we also have $W = L_{\underline{p}}^T$ and $W^+ = L_{\underline{p}}^{T+}$. Furthermore, the secondary task $\underline{e}_2$ is chosen as:

$$\underline{e}_2 = \begin{pmatrix} X(t) - X(0) - V_1(t) \\ Y(t) - Y(0) - V_2(t) \\ Z(t) - Z(0) + xV_1(t) + yV_2(t) \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad (24)$$

where $(X(t), Y(t), Z(t))^T$ is the current camera position, $(X(0), Y(0), Z(0))^T$ is its initial position and $V_1(t)$ and $V_2(t)$ are any temporal functions such that $V_1(0) = V_2(0) = 0$, whose role is to make the camera move in the $\vec{X}$ and $\vec{Y}$ directions. Let us note that at least one of these two functions has to be non zero so that the camera really moves under the constraint $\underline{p} = \underline{p}^*$. For instance, we can set $V_1(t) = Vt$ (resp. $V_1(t) = \frac{1}{2}\Gamma t^2 + Vt$) if a constant camera velocity (resp. a constant acceleration) is desired.

As far as the third component of $\underline{e}_2$ is concerned, we have chosen $Z(t) = Z(0) + xV_1(t) + yV_2(t)$ [8], in order to satisfy the additional constraint $\dot{Z} = 0$ that suppresses the discretization error. Finally, when the focusing task is

realized (i.e. $x = y = 0$), we get:

$$T = -(\mathbb{I}_6 - W^+W)\frac{\partial \underline{e}_2}{\partial t} = \frac{Z}{1+Z^2}\begin{pmatrix} -Z\dot{V}_1(t) \\ -Z\dot{V}_2(t) \\ 0 \\ \dot{V}_2(t) \\ -\dot{V}_1(t) \\ 0 \end{pmatrix} \quad (25)$$

which allows the camera to move on the sphere the center of which is the considered point, if the required estimated value Z corresponds to the real one (otherwise, the focusing task is not realized and the control law generates camera motions compensating the introduced perturbation). We can also check on (25) that $T, \dot{T}, ...T^{[n]}$ all belong to Ker $L_{\underline{P}}^T$.

Let us finally note that it is straightforward to elaborate similar control laws for other primitives such as straight line, cylinder, and so on [8].

4 Experimental Results

We present in this section the experimental results obtained for the 3D structure estimation of a point and a cylinder. For each of these primitives, we compare the results delivered by a dynamic vision approach (i.e., unconstrained camera motion) with those given by the active vision approach in order to demonstrate the improvement brought by the latter.

Our experimental system is composed of a camera mounted on the end effector of a six d.o.f. robot arm (see Fig. 5). The image processing part is performed on a commercial board. A classical robot speed control scheme has been used to perform the specified camera velocities, which allows the successive image acquisitions and structure estimations to be executed while the camera is continuously moving. In all the experiments described below, results have been obtained at a rate of 12.5 Hz, which means that each iteration of the 3D structure estimation is achieved in 80 milliseconds. Let us recall that an iteration k consists in:

- acquiring an image,
- extracting the visual features $\underline{p}_{(k)}$ and their velocity $\underline{\dot{p}}_{(k)}$,
- measuring the camera velocity $T_{(k)}$,
- estimating the parameters $\underline{P}_{(k)}$ describing the 3D spatial structure of the considered primitive, and
- if active vision is used instead of dynamic vision, computing the next desired camera velocity $T_{(k+1)}$ by visual servoing.

Let us note that $\underline{\dot{p}}_{(k)}$ is computed using the expression $(\underline{p}_{(k)} - \underline{p}_{(k-n)})/(n\Delta t)$, where Δt is the duration of each estimation iteration (i.e., 80 ms), and $n \geq 1$. Similarly, $T_{(k)}$ is measured from the robot joint positions recorded at iterations (k) and $(k-n)$.

Let us also point out that the estimated structure parameters depicted on all the following plots have been obtained by computing the average of the three successive estimated values $\underline{P}_{(k)}$, $\underline{P}_{(k-1)}$ and $\underline{P}_{(k-2)}$.

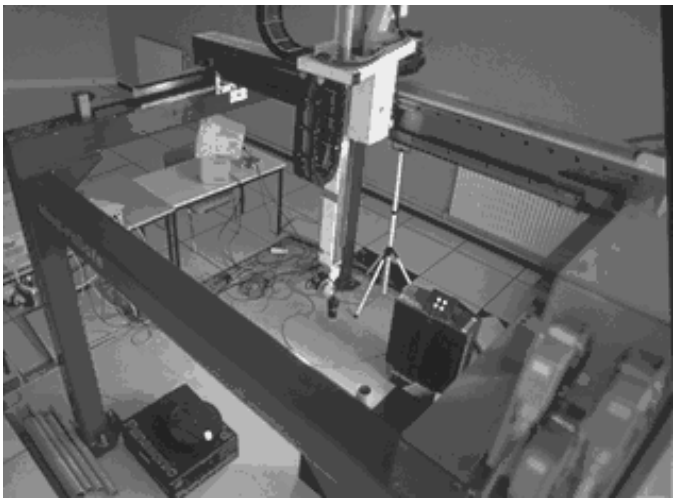


Fig. 5. Our experimental cell at IRISA composed of a camera mounted on the end effector of a six d.o.f. cartesian robot.

4.1 Case of the 3D Point Reconstruction

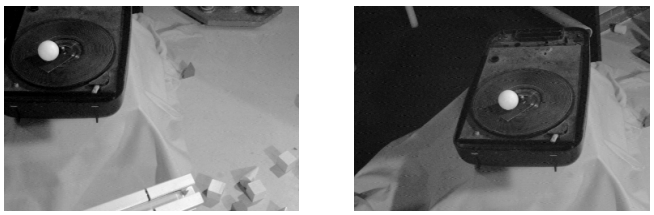
The first image acquired by the camera is depicted in Fig. 6.a. The point that we consider is the center of gravity of the white ball which is in the field of view of the camera. The image processing step simply consists in extracting and tracking along the image sequence the center of gravity of the ellipse corresponding to the projection of the ball in the image. The image is segmented using a simple thresholding technique.

4.1.1 Results Using Dynamic Vision

From the initial position, the camera moves with successive different constant velocities. More precisely, in the reported experiment, we have set $V_X = V_Y = V_Z = 40$ mm/s, $\Omega_X = \Omega_Y = \Omega_Z = 3.5$ dg/s from iterations 0 to 30, 60 to 90 and 120 to 150, and $V_X = V_Y = V_Z = -40$ mm/s, $\Omega_X = \Omega_Y = \Omega_Z = -3.5$ dg/s from iterations 30 to 60, 90 to 120 and 150 to 180. These motions do not satisfy the fixating and focusing constraints of our active scheme. It induces a motion of the point in the image of about 2 pixels between each image acquisition. The obtained results using $n = 1$ (i.e. two successive images) are shown in Fig. 7.a. Plots represent the 3D coordinates (X, Y, Z) of the considered point, estimated at each iteration of our algorithm. The position (X, Y, Z) of the point is expressed in a world reference coordinate system (corresponding to the first camera position). The plots should then correspond to constant values over time. As expected, unstable and noisy results can be observed in that experiment where no particular strategy as far as camera motion is concerned has been selected: the standard deviation on depth Z is about 3 cm and errors can reach 10 cm (that is 12.5%).

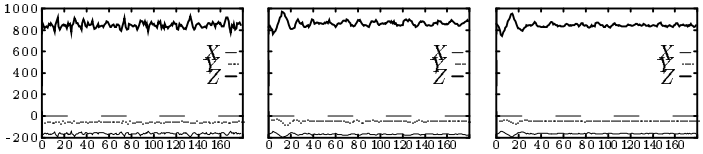
If larger values of n are considered (see Fig. 8.a where a value of $n = 20$ has been used), more stable results are obtained. However, as explained in Section 3.1, discretization effects due to the important image motion magnitude and change of depth between instants k and $k-n$, induce a bias in the estimation. In this experiment, the mean value

of estimated depth Z is 82 cm when considering the first camera motion (iterations 0 to 30, 60 to 90, etc.), while it is 86 cm for the other one.



(a) (b)

Fig. 6. Images acquired at the initial camera position (a) and after the realization of the focusing task (b).

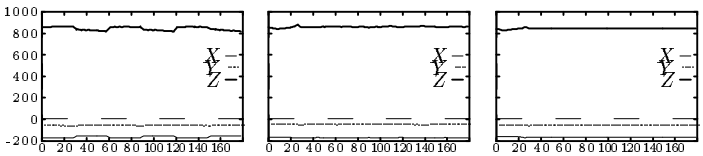


(a) (b) (c)

Fig. 7. Successive values of the 3D point coordinates X, Y, Z (expressed in mm) estimated with $n = 1$: (a) using a dynamic vision approach, (b) obtained with camera motion allowing only to suppress the discretization error, (c) using the complete active vision scheme.

4.1.2 Results Using Active Vision

As proposed in Section 3.1 to improve these results, we first constrain the camera movement in order to suppress the discretization error. Consequently, the point projection must remain static in the image, and the distance between the camera and the point has to be maintained constant. To perform that task, we use the control law described in Section 3.3 by setting $V_1(t) = V_2(t) = 0.1t$ from iterations 0 to 30, 60 to 90, etc, and $V_1(t) = V_2(t) = -0.1t$ from iterations 30 to 60, 90 to 120, etc. This leads to a translational camera velocity with equivalent amplitude than in the previous experiment. Results using $n = 1$ are shown in Fig. 7.b. During the first iterations, errors remain large. This is due to the fact that several iterations are required by the control law to correctly achieve the fixation task. After this short transient period, results are better since the standard deviation on Z is now equal to 1.7 cm. They are however not yet optimal and will be further improved thanks to the focusing step of our active vision scheme.



(a) (b) (c)

Fig. 8. Same experiments of the 3D point structure estimation as in Fig. 7 but with $n = 20$.

Using $n = 20$ leads to more accurate results (see Fig. 8.b): the obtained estimations are quite stable, with a standard deviation on Z equal to 3.6 mm. This last experiment involving a large time interval also proves that the discretization step has really no more influence.

As explained in Section 3.2, still more robust results can be obtained by positioning the camera in such a way that the point constantly appears at the center of the image (see Fig. 6.b). Indeed, effects of the measurement errors are minimized for that position.

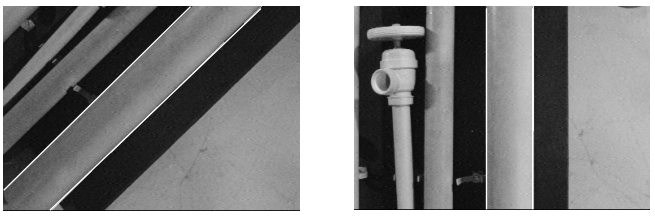
The estimated 3D point coordinates obtained after the realization of the focusing task are shown in Fig. 7.c and Fig. 8.c. For $n = 1$, we obtain a standard deviation on Z equal to 1 cm and, for $n = 20$, it is equal to 1.3 mm (that is 0.2%). These results are particularly stable and accurate compared with all the previous ones.

Because of the quality of the obtained results using a large value of n and a simple average filter, we have not implemented a method to integrate all the successive measured data. Such a method [27], [36] would of course be of great interest if inaccurate results were obtained, or if the camera motion was unknown, which is not within the scope of this paper. In conclusion, these experiments confirm that an important camera translation between the two considered images is essential to obtain stable results (which is quite well-known). Furthermore, they demonstrate that a continuous method can reach stable results without any bias provided adequate camera motions are performed. Finally, they prove that the accuracy of the estimation can be improved by controlling the location in the image of the primitive projection.

4.2 Case of the Cylinder Reconstruction

Let us now apply this approach to the 3D reconstruction of a cylinder. In that case, the parameters to be estimated are its radius R, the direction (U, V, W) of its axis and the coordinates (X_0, Y_0, Z_0) of the point of the cylinder axis the nearest to the camera (see Appendix Section A.4).

The initial image acquired by the camera is shown in Fig. 9.a (note the superimposed two white lines corresponding to the two selected limbs of the cylinder). The image processing step now consists in tracking these two straight lines along the image sequence and in determining the (ρ, θ) parameters describing their position in the image. The extraction, maintenance and tracking of the contour segments (in fact a list of edge points) are achieved in 40 ms. The technique we use is described in [7]. It is based on a local and robust matching of the moving edge-points constituting the selected line.



(a) (b)

Fig. 9. Images acquired at the initial camera position (a) and after the realization of the focusing task (b).

Since the observed phenomena are similar to those already described in the case of the point reconstruction, we only report the results obtained using a not constrained camera motion with $n = 1$ (see Fig. 10), and results obtained using our active vision scheme with $n = 20$ (see Fig. 11) once the focusing task has been achieved (see Fig. 9.b). We can point out the very significant improvement resulting from the use of the active vision paradigm. As far as depth estimation is concerned, the maximal error on Z_0 is about 2 mm using active vision, which can be compared very favorably to the important error (about 5 cm) obtained using only dynamic vision. As far as cylinder radius is concerned, the error between its real value (40 mm) and the estimated one is plotted in Fig. 12. After the first iterations, this error is lower than 0.5 mm and generally around 0.2 mm, whereas the cylinder is located at 80 cm from the camera. These accurate results validate the proposed active vision scheme.

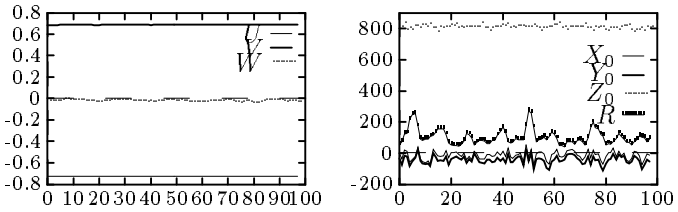


Fig. 10. Successive values of the 3D cylinder parameters estimated using dynamic vision for $n = 1$ ($V_X = V_Y = V_Z = 50$ mm/s, $\Omega_X = \Omega_Y = \Omega_Z = 3$ dg/s).

5 Conclusion

We have described an original formulation of the problem of reconstructing 3D parametric geometrical primitives using a mobile monocular camera. The introduction of the interaction matrix related to the primitive under concern allows us to define a general and attractive framework which can be applied to usual primitives such as points and straight lines, but also to more complex primitives such as cylinders, circles and spheres, without additional complexity in the derivation of the solution. Let us recall that our method only requires an explicit equation of the primitive to be reconstructed, from which we can determine the equations of the limb surface, the projection of the primitive in the image, and the related interaction matrix. Thanks to a continuous paradigm based on the link between the image motion and the camera velocity, our 3D structure estimation scheme simply consists in determining the intersection of the limb surface and the viewing cone.

Since the nature of camera motion affects the accuracy of the results, we have focused on this critical aspect of dynamic vision. We have mathematically and experimentally shown that very noticeable improvements can be obtained in the 3D reconstruction, if the camera viewpoint is properly selected, and if optimal camera motions are generated. Our approach consists in particular in fixating at and focusing on the 3D primitive to be determined. This confirms the point of view of previous works on the

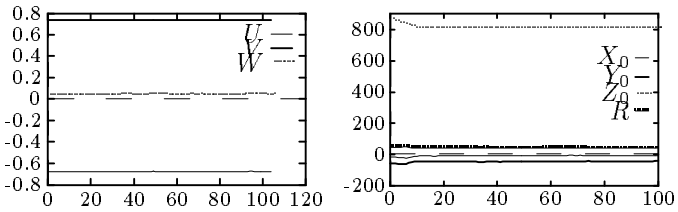


Fig. 11. Successive values of the 3D cylinder parameters estimated using the active vision scheme for $n = 20$.

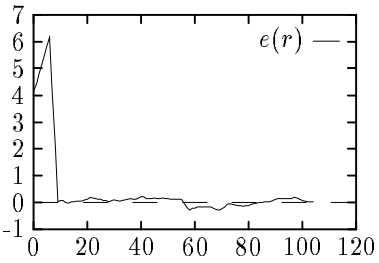


Fig. 12. Errors (in mm) between the real value of the cylinder radius and the estimated one using the active vision scheme.

promising strength of active vision paradigms [3], [4], [6], [32] and gaze-holding [5], [24]. We have stressed that this active vision approach can be adequately performed using visual servoing. A real-time version of this structure from controlled motion scheme has been implemented on an experimental system, and it turns out to be powerful and efficient.

Current work is devoted to the development of global perceptual strategies able to appropriately combine a succession of such optimal individual primitive reconstruction steps to recover the complete spatial structure of complex scenes [23].

Appendix

We describe in this appendix how the 3D structure estimation method described in Section 2 can be applied to the most representative geometrical primitives: straight lines, circles, spheres and cylinders.

A.1 Case of the Straight Line

A straight line can be represented as the intersection of two orthogonal planes:

$$h(\underline{X}, \underline{P}) = \begin{cases} A_1X + B_1Y + C_1Z = 0 \\ A_2X + B_2Y + C_2Z + D_2 = 0 \end{cases} \quad (26)$$

$$\text{with} \begin{cases} A_1^2 + B_1^2 + C_1^2 = 1 \\ A_2^2 + B_2^2 + C_2^2 = 1 \\ A_1A_2 + B_1B_2 + C_1C_2 = 0. \end{cases}$$

A minimal and complete representation of the resulting 2D line can be derived:

$$g(\underline{x}, \underline{p}) = x\cos\theta + y\sin\theta - \rho = 0 \quad (27)$$

$$\text{with} \begin{cases} \cos\theta &= A_1/\sqrt{A_1^2 + B_1^2} \\ \sin\theta &= B_1/\sqrt{A_1^2 + B_1^2} \\ \rho &= -C_1/\sqrt{A_1^2 + B_1^2}. \end{cases}$$

Furthermore, the function μ defined in (7) is obtained from (26) as follows:

$$1/Z = \mu(\underline{x}, \underline{P}_L) = Ax + By + C \tag{28}$$

with $A = -A_2/D_2, B = -B_2/D_2$ and $C = -C_2/D_2$. Finally, the relation between the image line motion and the camera velocity is given by the interaction matrix related to $\underline{p} = (\rho, \theta)$. More precisely, we have [14]:

$$\begin{aligned} \dot{\underline{p}} &= \begin{pmatrix} \lambda_\rho \cos\theta & \lambda_\rho \sin\theta & -\lambda_\rho \rho \\ \lambda_\theta \cos\theta & \lambda_\theta \sin\theta & -\lambda_\theta \rho \end{pmatrix} V(0) \\ &+ \begin{pmatrix} (1+\rho^2)\sin\theta & -(1+\rho^2)\cos\theta & 0 \\ -\rho\cos\theta & -\rho\sin\theta & -1 \end{pmatrix} \Omega \end{aligned} \tag{29}$$

with $\lambda_\rho = -A\rho\cos\theta - B\rho\sin\theta - C$, and $\lambda_\theta = B\cos\theta - A\sin\theta$.

We now have to estimate the parameters of the two planes which define the straight line from the measured values of $\underline{p}$, $\dot{\underline{p}}$ and T.

Let us first point out that parameters A_1, B_1 and C_1 are straightforwardly obtained from ρ and θ. Next, as stated in Section 2, parameters describing function μ are determined from the measure of the camera velocity and the resulting straight line motion in the image. Indeed, A, B, C are given by the resolution of the following linear system:

$$\begin{cases} -A\rho\cos\theta - B\rho\sin\theta - C &= \lambda_\rho \\ -A\sin\theta + B\cos\theta &= \lambda_\theta \\ A\cos\theta + B\sin\theta - C\rho &= 0 \end{cases} \tag{30}$$

where λ_ρ and λ_θ are obtained from (29):

$$\begin{cases} \lambda_\rho &= \dfrac{\dot{\rho} + (1+\rho^2)(\Omega_Y\cos\theta - \Omega_X\sin\theta)}{V_X\cos\theta + V_Y\sin\theta - \rho V_Z} \\ \lambda_\theta &= \dfrac{\dot{\theta} + \rho(\Omega_X\cos\theta + \Omega_Y\sin\theta) + \Omega_Z}{V_X\cos\theta + V_Y\sin\theta - \rho V_Z}. \end{cases} \tag{31}$$

Finally, we get $D_2 = 1/\sqrt{A^2+B^2+C^2}$, $A_2 = -AD_2$, $B_2 = -BD_2$ and $C_2 = -CD_2$.

We can easily show, by computing $\dfrac{\partial \mathcal{H}}{\partial \underline{P}_L}$ defined from (10), that camera motions unable to provide any information on the spatial location of the 3D straight line are such that $V_X\cos\theta + V_Y\sin\theta - \rho V_Z = 0$ (see also (31)). These motions are such that the center of projection O constantly belongs to the plane defined by the initial position of O and the image line.

A.2 Case of the Circle

A circle can be represented as the intersection of a sphere and a plane which contains the center of the sphere:

$$h(\underline{X}, \underline{P}) = \begin{cases} (X-X_0)^2+(Y-Y_0)^2+(Z-Z_0)^2-R^2=0 \\ U(X-X_0)+V(Y-Y_0)+W(Z-Z_0)=0 \end{cases} \tag{32}$$

where R is the radius of the circle, X_0, Y_0 and Z_0 are the coordinates of its center and U, V, W are the components of the vector normal to the circle plane (with $U^2+V^2+W^2=1$). Function μ defined by (7) is directly obtained from (32):

$$1/Z = \mu(\underline{x}, \underline{P}_L) = Ax + By + C \tag{33}$$

$$\text{where } \begin{cases} A &= U/(UX_0+VY_0+WZ_0) \\ B &= V/(UX_0+VY_0+WZ_0) \\ C &= W/(UX_0+VY_0+WZ_0) \end{cases}$$

The ellipse which results from the projection of a circle can be defined as follows [8]:

$$g(\underline{x}, \underline{p}) = x^2 + a_1y^2 + 2a_2xy + 2a_3x + 2a_4y + a_5 = 0 \tag{34}$$

$$\text{with } \begin{cases} a_1 = (B^2K + 1 - 2BY_0)/a_0 \\ a_2 = (ABK - BX_0 - AY_0)/a_0 \\ a_3 = (ACK - CX_0 - AZ_0)/a_0 \\ a_4 = (BCK - CY_0 - BZ_0)/a_0 \\ a_5 = (C^2K + 1 - 2CZ_0)/a_0 \\ a_0 = A^2K + 1 - 2AX_0 \\ K = X_0^2 + Y_0^2 + Z_0^2 - R^2 \end{cases} \tag{35}$$

The motion of the ellipse parameters is given by [14]:

$$\begin{aligned} \dot{a_1} &= 2(Ba_2 - Aa_1)V_X + 2a_1(B - Aa_2)V_Y \\ &\quad +2(Ba_4 - Aa_1a_3)V_Z \\ &\quad +2a_4\Omega_X + 2a_1a_3\Omega_Y - 2a_2(a_1+1)\Omega_Z \\ \dot{a_2} &= (B - Aa_2)V_X + [Ba_2 - A(2a_2^2 - a_1)]V_Y \\ &\quad +[A(a_4 - 2a_2a_3) + Ba_3]V_Z \\ &\quad +a_3\Omega_X + (2a_2a_3 - a_4)\Omega_Y + (a_1 - 2a_2^2 - 1)\Omega_Z \\ \dot{a_3} &= (C - Aa_3)V_X + [A(a_4 - 2a_2a_3) + Ca_2]V_Y \\ &\quad +[Ca_3 - A(2a_3^2 - a_5)]V_Z \\ &\quad -a_2\Omega_X + (1 + 2a_3^2 - a_5)\Omega_Y + (a_4 - 2a_2a_3)\Omega_Z \\ \dot{a_4} &= (a_3B + a_2C - 2Aa_4)V_X + (a_4B + a_1C - 2Aa_2a_4)V_Y \\ &\quad +(Ba_5 + Ca_4 - 2Aa_3a_4)V_Z \\ &\quad +(a_5 - a_1)\Omega_X + (2a_3a_4 + a_2)\Omega_Y - (2a_2a_4 + a_3)\Omega_Z \\ \dot{a_5} &= 2(Ca_3 - Aa_5)V_X + 2(Ca_4 - Aa_2a_5)V_Y \\ &\quad +2(Ca_5 - Aa_3a_5)V_Z \\ &\quad -2a_4\Omega_X + 2(a_3a_5 + a_3)\Omega_Y - 2a_2a_5\Omega_Z. \end{aligned} \tag{36}$$

Parameters $\underline{P}_L = (A, B, C)$ are given by the mean least squares solution of the linear system defined by (36), from which we can deduce parameters U, V, and W. We can show that $\dfrac{\partial \mathcal{H}}{\partial \underline{P}_L}$ is always of full rank 3, provided a camera translational motion is performed. This means that an estimation of the 3D circle parameters can be obtained for any camera motion with a non zero translational component. Finally, using (35), we obtain the following linear system:

$$\begin{cases} (B^2 - A^2a_1)K + 2Aa_1X_0 - 2BY_0 &= a_1 - 1 \\ (AB - A^2a_2)K - (B - 2Aa_2)X_0 - AY_0 &= a_2 \\ (AC - A^2a_3)K - (C - 2Aa_3)X_0 - AZ_0 &= a_3 \\ (BC - A^2a_4)K + 2Aa_4X_0 - CY_0 - BZ_0 &= a_4 \\ (C^2 - A^2a_5)K + 2Aa_5X_0 - 2CZ_0 &= a_5 - 1 \end{cases} \tag{37}$$

The resolution of (37) allows us to simply determine X_0, Y_0, Z_0 and K (and thus R).

A.3 Case of the Sphere

The image of a sphere of radius R and center with coordinates (X_0, Y_0, Z_0) is again characterized by an ellipse of equation:

$$x^2 + a_1 y^2 + 2a_2 xy + 2a_3 x + 2a_4 y + a_5 = 0 \tag{38}$$

$$\text{with} \begin{cases} a_1 &= (R^2 - X_0^2 - Z_0^2)/a_0 \\ a_2 &= (X_0 Y_0)/a_0 \\ a_3 &= (X_0 Z_0)/a_0 \\ a_4 &= (Y_0 Z_0)/a_0 \\ a_5 &= (R^2 - X_0^2 - Y_0^2)/a_0 \\ a_0 &= R^2 - Y_0^2 - Z_0^2 \end{cases} \tag{39}$$

and function μ is given by [14]:

$$1/Z = \mu(\underline{x}, \underline{P}_L) = Ax + By + C \tag{40}$$

$$\text{with} \begin{cases} A &= X_0/(X_0^2 + Y_0^2 + Z_0^2 - R^2) \\ B &= Y_0/(X_0^2 + Y_0^2 + Z_0^2 - R^2) \\ C &= Z_0/(X_0^2 + Y_0^2 + Z_0^2 - R^2) \end{cases} \tag{41}$$

where $\underline{P}_L = (A, B, C)$, which represents the normal of the limb plane of the sphere, can be computed exactly as in the circle case. Finally, the radius and the position of the sphere can be determined using (39) and (41).

A.4 Case of the Cylinder

Let us now apply the same approach to the reconstruction of a cylinder, the equation of which is given by:

$$\begin{aligned} h(\underline{X}, \underline{P}) &= (X - X_0)^2 + (Y - Y_0)^2 + (Z - Z_0)^2 \\ &- (UX + VY + WZ)^2 - R^2 = 0 \end{aligned} \tag{42}$$

where R is the radius of the cylinder, (U, V, W) represents the direction of its axis, and (X_0, Y_0, Z_0) is the point of the cylinder axis which is the nearest to O. The parameters to be estimated are thus $\underline{P} = (U, V, W, X_0, Y_0, Z_0, R)$.

In the non-degenerate cases, the projection in the image plane of a cylinder is constituted by two straight lines $\mathcal{D}_1(\rho_1, \theta_1)$ and $\mathcal{D}_2(\rho_2, \theta_2)$. Function μ defined by (7) is given by [8]:

$$1/Z = \mu(\underline{x}, \underline{P}_L) = A\,x + B\,y + C \tag{43}$$

where $A = X_0/K$, $B = Y_0/K$, and $C = Z_0/K$ (with $K = X_0^2 + Y_0^2 + Z_0^2 - R^2$) are the components of the vector normal to the limb plane of the cylinder.

Besides, the motion of the cylinder in the image resulting from camera motion is described by the interaction matrix related to $\underline{p} = (\rho_1, \theta_1, \rho_2, \theta_2)$:

$$\begin{aligned} \dot{\underline{p}} &= \begin{pmatrix} \lambda_{\rho_1}\cos\theta_1 & \lambda_{\rho_1}\sin\theta_1 & -\lambda_{\rho_1}\rho_1 \\ \lambda_{\theta_1}\cos\theta_1 & \lambda_{\theta_1}\sin\theta_1 & -\lambda_{\theta_1}\rho_1 \\ \lambda_{\rho_2}\cos\theta_2 & \lambda_{\rho_2}\sin\theta_2 & -\lambda_{\rho_2}\rho_2 \\ \lambda_{\theta_2}\cos\theta_2 & \lambda_{\theta_2}\sin\theta_2 & -\lambda_{\theta_2}\rho_2 \end{pmatrix} V(0) \\ &+ \begin{pmatrix} (1+\rho_1^2)\sin\theta_1 & -(1+\rho_1^2)\cos\theta_1 & 0 \\ -\rho_1\cos\theta_1 & -\rho_1\sin\theta_1 & -1 \\ (1+\rho_2^2)\sin\theta_2 & -(1+\rho_2^2)\cos\theta_2 & 0 \\ -\rho_2\cos\theta_2 & -\rho_2\sin\theta_2 & -1 \end{pmatrix} \Omega \end{aligned} \tag{44}$$

$$\text{with} \begin{cases} \lambda_{\rho_1} &= -(A\rho_1\cos\theta_1 + B\rho_1\sin\theta_1 + C) \\ \lambda_{\theta_1} &= B\cos\theta_1 - A\sin\theta_1 \\ \lambda_{\rho_2} &= -(A\rho_2\cos\theta_2 + B\rho_2\sin\theta_2 + C) \\ \lambda_{\theta_2} &= B\cos\theta_2 - A\sin\theta_2. \end{cases} \tag{45}$$

Parameters $\underline{P}_L = (A, B, C)$ can be determined by solving the linear system (45), λ_{ρ_1}, λ_{θ_1}, λ_{ρ_2} and λ_{θ_2} being directly obtained from (44), exactly as in the previous case of the straight line.

It can be derived from (44) that the camera motions which do not permit to solve the linear system (45) are such that the center of projection O always remains in one of the planes defined by the initial position of O and the image lines (i.e. $V_X\cos\theta_1 + V_Y\sin\theta_1 - \rho_1 V_Z = 0$, or $V_X\cos\theta_2 + V_Y\sin\theta_2 - \rho_2 V_Z = 0$). Indeed, in that case, only two equations of (45) are valid, whereas three are necessary to obtain a unique solution. Moreover, if the camera motion is such that O constantly belongs to the line resulting from the intersection of the two planes (i.e., such that $V_X\cos\theta_1 + V_Y\sin\theta_1 - \rho_1 V_Z = 0$ and $V_X\cos\theta_2 + V_Y\sin\theta_2 - \rho_2 V_Z = 0$), then λ_{ρ_1}, λ_{θ_1}, λ_{ρ_2} and λ_{θ_2} are all let indeterminate.

Next, parameters $\underline{P}$ describing the location and the structure of the cylinder can easily be obtained from the intersection of the limb plane $\mathcal{P}_L$ and the planes $\mathcal{P}_{t1}$ and $\mathcal{P}_{t2}$, defined by O and the two image lines $\mathcal{D}_1$ and $\mathcal{D}_2$ (see Fig. 13). That intersection is formed by the two lines $\mathcal{D}_{\mathcal{P}_{L1}}$ and $\mathcal{D}_{\mathcal{P}_{L2}}$, from which we can determine the position of the cylinder axis. Indeed, its equation, which is sufficient to determine U, V, W, X_0, Y_0 and Z_0, is given by the intersection of the two planes $\mathcal{P}_{n1}$ and $\mathcal{P}_{n2}$, orthogonal to $\mathcal{P}_{t1}$ and $\mathcal{P}_{t2}$ and containing $\mathcal{D}_{\mathcal{P}_{L1}}$ and $\mathcal{D}_{\mathcal{P}_{L2}}$ respectively [8]. Finally, the cylinder radius R is obtained by computing the distance between the cylinder axis and $\mathcal{D}_{\mathcal{P}_{L1}}$ or $\mathcal{D}_{\mathcal{P}_{L2}}$.

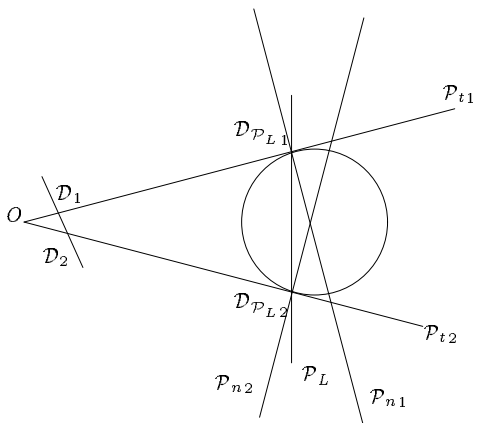


Fig. 13. Top view of a cylinder.

It is also possible to develop a similar method for estimating the 3D structure of a cylinder using only the projection of one limb in the image [8]. Such a method is useful when the correspondence between straight lines extracted from the image and limbs of the cylinder is not obvious to determine.

Acknowledgments

This work has been done at IRISA/INRIA Rennes in collaboration with CEA LETI DEIN under contract 1 91 C 244 00 31315 01 1. It was also partly supported by Brittany County Council (Région Bretagne) under contribution to student grant.

References

[1] L. Abbot and N. Ahuja: "Active surface reconstruction by integrating focus, vergence, stereo, and camera calibration," *Proc.* 3^{rd} *ICCV*, Osaka, Japan, pp. 489-492, Dec. 1990.

[2] G. Adiv: "Inherent ambiguities in recovering 3d motion and structure from a noisy flow field," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 5, pp. 477-489, May 1989.

[3] J. Aloimonos , I. Weiss and A. Bandopadhay: "Active vision," *Proc.* 1^{st} *ICCV*, London, England, pp. 35-54, June 1987.

[4] R. Bajcsy: "Active perception," *Proc. of the IEEE*, vol. 76, no. 8, pp. 996-1005, Aug. 1988.

[5] D. Ballard: "Animate vision," *Artificial Intelligence*, vol. 48, pp. 57-86, 1991.

[6] A. Bandopadhay, B. Chandra and D. Ballard: "Egomotion using active vision," *Proc. CVPR 86*, Miami, pp. 498-503, June 1986.

[7] S. Boukir, P. Bouthemy, F. Chaumette and D. Juvin: "Real-time contour matching over time in an active vision context," *Proc.* 8^{th} *Scandinavian Conf. on Image Analysis*, vol. 1, pp. 113-120, Tromso, Norway, May 1993.

[8] S. Boukir: "Reconstruction 3D d'un environnement statique par vision active," PhD thesis, no. 1074, IRISA, University of Rennes I, France, Oct. 1993.

[9] C. Brown: "Issues in selective perception," *Proc.* 11^{th} *ICPR*, The Hague, Netherlands, vol. 1, pp. 21-24, Aug. 1992.

[10] F. Chaumette, S. Boukir, P. Bouthemy and D. Juvin: "Optimal estimation of 3D structures using visual servoing," *Proc. CVPR 94*, pp. 347-354, Seattle, June 1994.

[11] C. Chien and J.K. Aggarwal: "Model construction and shape recognition from occluding contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 4, pp. 372-389, Apr. 1989.

[12] J. Crowley, P. Bobet and M. Mesrabi: "Layered control of a binocular camera head," *Int. Journal of Patt. Recog. and Artif. Intel.*, vol. 7, no. 1, pp. 109-116, Feb. 1993.

[13] K. Daniilidis and H.H. Nagel: "Analytical results on error sensitivity of motion estimation from two views," *Proc.* 1^{st} *ECCV*, pp. 199-208, Antibes, France, Apr. 1990.

[14] B. Espiau, F. Chaumette and P. Rives: "A new approach to visual servoing in robotics," *IEEE Trans. on Robotics and Automation*, Vol 8, no. 3, pp. 313-326, June 1992.

[15] B. Espiau and P. Rives: "Closed-loop recursive estimation of 3D features for a mobile vision system," *IEEE Int. Conf. on Robotics and Automation*, Raleigh, North Carolina, vol. 3, pp. 1436-1443, Apr. 1987.

[16] O. Faugeras: *Three-dimensional computer vision: a geometric viewpoint*. Cambridge, Mass: MIT Press, 1993.

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

[18] J. Herve and Y. Aloimonos: "Exploratory active vision: theory," *Proc. CVPR 92*, Champaign, Ill., pp. 10-15, June 1992.

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

[20] L. Huang and J. Aloimonos: "Relative depth from motion using normal flow: an active and purposive solution," *Proc. IEEE Workshop on Visual Motion*, Princeton, N.J., pp. 196-204, Oct. 1991.

[21] S. Li, I. Miyawaki, H. Ishiguro and S. Tsuji: "Realizing active vision by a mobile robot," *Proc. IEEE Workshop on Visual Motion*, Princeton, N.J., pp. 205-210, Oct. 1991.

[22] L. Matthies,T. Kanade and R. Szelisli: "Kalman filter-based algorithms for estimating depth from image sequences," *IJCV*, vol. 3, pp. 209-236, 1989.

[23] E. Marchand and F. Chaumette: "Active Visual 3D Perception," *IEEE Int. Workshop on Vision for Robots*, Pittsburgh, pp. 10-17, Aug. 1995.

[24] E. Milios, M. Jenkin and J. Tsotsos: "Design and performance of trish, a binocular robot head with torsional eye movements," *Int. Jour. of Patt. Recog. and Artif. Intel.*, vol. 7, no. 1, pp. 51-68, Feb. 1993.

[25] A. Mitiche and G. Habelrih: "Interpretation of straight line correspondences using angular relations," *Pattern Recognition*, vol. 22, no. 3, pp. 299-308, 1989.

[26] S. Negahdaripour and B. Horn: "Direct passive navigation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 1, pp. 168-176, Jan. 1987.

[27] J. Oliensis and J. Inigo Thomas: "Incorporating motion error in multi-frame structure from motion," *IEEE Workshop on Visual Motion*, Princeton, N.J., pp. 8-13, Oct. 1991.

[28] K. Pahlavan and J.-O. Ekhlund: "A head-eye system: Analysis and design," *CVGIP: Image Understanding*, vol. 56, no. 1, pp. 41-56, July 1992.

[29] R. Safaee-Rad, I. Tchoukanov, K.C. Smith and B. Benhabib: "Three-Dimensional Location Estimation of Circular Features for Machine Vision," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 5, pp. 624-639, Oct. 1992.

[30] P. Rives and M. Xie: "Towards dynamic vision," *Proc. IEEE Workshop on Interpretation of 3D Scenes*, Austin, Tex., Nov. 1989.

[31] G. Sandini and M. Tistarelli: "Recovery of depth information: camera motion as an integration to stereo," *Proc. IEEE Workshop on Motion*, Charleston, S.C., pp. 39-43, May 1986.

[32] G. Sandini and M. Tistarelli: "Active tracking strategy for monocular depth inference over multiple frames," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 13-27, Jan. 1990.

[33] A. Shashua: "Projective depth: a geometric invariant for 3D reconstruction from two perspective/orthographic views and for visual recognition," *Proc.* 4^{th} *ICCV*, Berlin, Germany, pp. 583-590, May 1993.

[34] B. Thompson, P. Lechleider and E. Stuck: "Detecting moving objects using the rigidity constraint," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 2, pp. 162-166, Feb. 1993.

[35] M. Tistarelli and G. Sandini: "Dynamic aspects in active vision," *CVGIP: Image Understanding*, vol. 56, no. 1, pp. 108-129, July 1992.

[36] C. Tomasi and T. Kanade: "Shape and motion from image streams under orthography: a factorization method," *IJCV*, vol. 9, no. 2, pp. 137-154, Nov. 1992.

[37] D. Vernon and M. Tistarelli: "Using camera motion to estimate range for robotic part manipulation," *IEEE Trans. on Robotics and Automation*, vol. 6, no. 5, pp. 509-521, Oct. 1990.

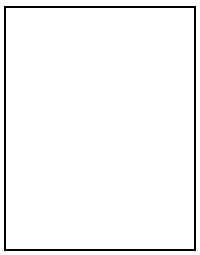
[38] M. Viala, C. Faye, J.-P. Guérin and D. Juvin: "Cylindrical object reconstruction from a sequence of images," *Proc. SPIE Intelligent Robots and Visual Communications*, Boston, Nov. 1992.

[39] T. Viéville, J.-O. Ekhlund, K. Pahlavan and T. Uhlin: "An example of artificial oculomotor behavior," *Proc.* 7^{th} *IEEE Symp. on Intelligent Control*, Glasgow, Scotland, pp. 348-353, Aug. 1992.

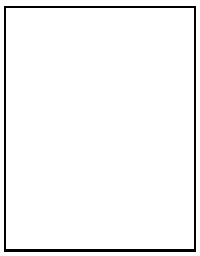
[40] A. Waxman, B. Parsi and M. Subbarao: "Closed-form solutions to image flow equations for 3D structure and motion," *IJCV*, vol. 1, no. 3, pp. 239-258, Oct. 1987.

[41] L. Weiss and A. Sanderson: "Dynamic sensor-based control of robots with visual feedback," *IEEE Journal of Robotics and Automation*, vol. 3, no. 5, pp. 404-417, Oct. 1987.

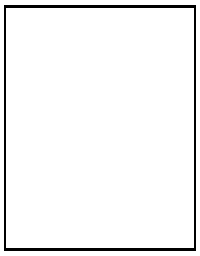
[42] J. Weng, T. Huang and N. Ahuja: "Motion and structure from line correspondences: closed-form solution, uniqueness, and optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 3, pp. 318-336, Mar. 1992.



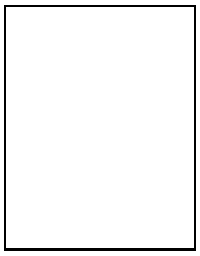
François Chaumette graduated from Ecole Nationale Supérieure de Mécanique, Nantes, in 1987, and received the Ph.D degree in Computer Science from the University of Rennes, France, in 1990. He received in 1991 the AFCET / CNRS Prize of the best french thesis in Automatic Control. Since 1990, he has been with Inria, at Irisa in Rennes, where he is currently "Chargé de Recherche". His research interests include robotics, computer vision, and especially the coupling between these two research domains (vision-based control, active vision and purposive vision).



Samia Boukir received the computer science engineering degree from the University of Algiers, Algeria, in 1989, and the Ph.D. degree from the University of Rennes, France, in 1993. Since 1993, she has been an assistant professor at University of La Rochelle, France, and has worked at L3i (Laboratoire d'Informatique et d'Imagerie Industrielles) on image processing. Her current research interests include robotics, image processing, and computer vision, particularly in the area of three-dimensional and dynamic scene analysis.



Patrick Bouthemy graduated from Ecole Nationale Supérieure des Télécommunications, Paris, in 1980, and received the Ph.D degree in Computer Science from the University of Rennes, France, in 1982. From December 1982 until February 1984 he was employed by INRS-Télécommunications, Montréal, P.Q., Canada, in the Department of Visual Communications. Since April 1984 he has been with Inria, at Irisa in Rennes. He is currently "Directeur de Recherche" Inria. His major research interests are concerned with both visual perception and image-based communication: motion analysis (rigid motion and deformable motion), statistical models for image sequence processing (MRF models, robust estimation, Bayesian estimation), dynamic scene interpretation and active vision.



Didier Juvin graduated Engineer in Electronics from Institut Supérieur d'Electronique, Paris, since 1977. He first worked at Thomson on simulators developing image synthesis algorithms. Then he joined CEA (French Atomic Energy Commission) in 1980 at the DEIN (Electronics and Nuclear Instrumentation Department) and performed his thesis in image processing in 1983. He then developped with IRIT a new parallel concept for image processing. He was responsible of the Sympati 2 project that led to an industrial product (Centralp: Openvision Systems). He is Head of the Parallel Architecture Group inside the Software and Architecture Laboratory of LETI. This group is concerned with massively parallel architecture of computers, and its applications in industrial environment. His principal work is concerned with developing parallel computer architectures devoted to image processing and pattern recognition.