Visual Servoing from Lines

Nicolas Andreff, Bernard Espiau, Radu Horaud

Visual Servoing from Lines

Nicolas Andreff[†] Bernard Espiau[‡]
Radu Horaud[‡]
[†] LaRAMA – UBP
Institut Français de Mécanique Avancée
BP 265, 63175 Aubière Cedex, France
email: nicolas.andreff@ifma.fr
[‡] INRIA Rhône-Alpes and GRAVIR-IMAG
655, av. de l'Europe, 38330 Montbonnot Saint Martin, France
email: firstname.lastname@inrialpes.fr

January 29, 2003

International Journal of Robotics Research, vol. 21, no. 8, pages 679–700, August 2002.

Abstract

In this paper we present a new approach to visual servoing using lines. It is based on a theoretical and geometrical study of the main line representations which allows us to define a new representation, the so-called binormalized Plücker coordinates. They are particularly well suited for visual servoing. Indeed, they allow the definition of an image line alignment concept. Moreover, the control law which realizes such an alignment has several properties: partial decoupling between rotation and translation, analytical inversion of the motion equations and global asymptotic stability conditions. This control law was validated both in simulation and experimentally in the specific case of an orthogonal trihedron.

Figure 1: Detail of a part to be welded.

1 Motivation

The present paper is mainly concerned with the use of lines in visual servoing[1]. At the origin of the problem is the fact that when dealing with industrial parts, the extraction and the tracking of lines is natural and, now, feasible in real time thanks to new efficient algorithms from computer vision [Low91, MBCM99, TN00, AZ95, DC00, MH02]. Nevertheless, only a few works have really explored the use of lines in visual servoing: [RE87, ECR92, KMM+96, KRD+98, Hag97] are examples we find in robotics. Tracking of lines was also considered for the road following of autonomous vehicles, like in [LD98], but without deriving generic approaches.

This work was performed in the framework of a European project, VIGOR [VIG01]. The main industrial partner was the Odense Steel Shipyard in Denmark and a major concern of the project was to improve the automatic welding of ship parts. In particular, it was required to accurately move a welding tool with respect to a ship part, whose relative position is not exactly known in advance.

The state-of-the-practice combines off-line trajectory generation based on CAD models and on-line following of the generated trajectory. This requires that the CAD models are perfect and that there is no discrepancy between the position of the part with respect to the robot base at trajectory generation time and at trajectory following time. This is a strong constraint in the case of ship welding since the size and the weight of the parts to be welded prevent from achieving an accurate positioning of these parts. Consequently, it was necessary to use sensor-based techniques, such as visual servoing.

[1] Visual servoing is now rich of some hundreds of references, one of the first reviews on the subject is [Cor93] and a tutorial may be found in [HHC96].

Since 75% of the parts to be welded are composed of three intersecting orthogonal planes, where points can not be reliably extracted (see Figure 1), we put the emphasis of the application of our studies to this junction of three orthogonal lines. The question which rapidly occurred was then: what representation for these lines?

Indeed, as soon as we use structures richest than simple points, features in the image result from 3D entities which belong to particular manifolds. The choice of the parameterization of these manifolds (and of the projected structures) strongly influences the design of the control law. Among the expected benefits are the better decoupling, the easier design and stability analysis of the control.

Coming to geometrical aspects, it should be recalled that the 3D Euclidean lines form a 4-dimensional manifold the properties of which are described by the Grassman's geometry. One of the most noticeable property is the absence of a natural (Riemannian) distance. Since such a distance is needed for defining a feedback control, the choice of the parameterization is crucial. Among the classical ones, the Plücker coordinates look particularly interesting since they come from the embedding of the line manifold in a projective space in which things are linear. More, recalling that the camera transformation is also projective, we understand intuitively that some nice properties could be expected.

These issues motivated the work we report on in the present paper[2]. In a first part, we set the general problem of designing a feedback control law for 3D lines, in two steps:

- First, we propose a restriction on classical Plücker coordinates (the so-called *binormalized Plücker coordinates*) which explicitly splits the representation in such a way that depth becomes an independent variable. This allows us to define the notion of image line alignment.

- Second, we define a control law on the related sub-manifold and prove its stability. The use of binormalized Plücker coordinates yields a partial decoupling between rotation and translation control. Thanks to this property, the control law consists of an *analytical* inversion of motion equations and the stability proof exhibits *global* analytic conditions for stationarity.

The second part of the paper presents the application of the method to the case of the orthogonal trihedron (with geometrical interpretation of the stationary points) and, finally, proposes a full practical implementation dedicated to visual servoing by adding a laser spot to the system.

[2]A shorter version of this version can be found in ([AE02])

The remainder of this paper is organized as follows. Section 2 is devoted to line parameterizaton and Section3 to control definition. In section 4, we address practical issues specific to the orthogonal trihedron. Finally, Sections 5 and 6 report respectively simulation and experimental results.

- Vectors are denoted with lower case upright bold letters (e.g.$\mathbf{u}$). Unit vectors are denoted as underlined vectors (e.g. $\underline{\mathbf{u}}$).
- Points in the 3D space are written with upper case bold letters ($\mathbf{P}$) while points in the image are written with lower-case bold characters ($\mathbf{p}$).
- 3D Lines are noted with calligraphic upper case letters (e.g. $\mathcal{L}$) and image lines with calligraphic lower case letters (e.g. ℓ).
- Angular velocity is written Ω and linear velocity V.
- The scalar product is noted with the transpose formalism (e.g. $\mathbf{a}^T\mathbf{b}$) and the cross product of $\mathbf{a}$ by $\mathbf{b}$ is written $\mathbf{a} \times \mathbf{b}$.
- The notation * (e.g. $\underline{\mathbf{u}}^*$) denotes a desired value and the notation $(t = 0)$ (e.g. $\underline{\mathbf{u}}(t{=}0)$) represents an initial value.

Table 1: Notation used in this paper

2 Line parameterization

2.1 Binormalized Plücker coordinates

2.1.1 Recalls on Plücker coordinates

The (projective) Plücker coordinates[Plü65, BR79, Fau93, SK52] of a line in space, or 3D line, $(\mathcal{L})$ are the 6 coordinates of a point in the 5-dimensional projective space P^5. This point is sometimes called the *Plücker point*[GO97]. Its coordinates are usually noted as a couple of two 3-vectors defined up to a common scale factor:

$$\mathcal{L} = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \tag{1}$$

where $\mathbf{u} \neq \mathbf{0}$ denotes the orientation of $(\mathcal{L})$ and $\mathbf{v}$ contains the coefficients of the *interpretation plane* equation, the latter being the plane passing by the origin and $(\mathcal{L})$. Since physical lines in space form a 4-dimensional manifold, called the *Klein quadric*[PPR98] or the *Plücker hypersurface*[CEG+96], $\mathbf{u}$ and $\mathbf{v}$ satisfy the following constraint:

$$\mathbf{u}^T\mathbf{v} = 0 \tag{2}$$

It expresses, via the so-called *Grassmannian*, that a line intersects itself.

The Euclidean Plücker coordinates are obtained by normalizing $\mathbf{u}$:

$$\mathcal{L} = (\underline{\mathbf{u}}, \mathbf{h}) \;\; ; \;\; \underline{\mathbf{u}}^T\mathbf{h} = 0 \; ; \; \underline{\mathbf{u}}^T\underline{\mathbf{u}} = 1 \tag{3}$$

where $\underline{\mathbf{u}} = \mathbf{u}/\|\mathbf{u}\|$ and $\mathbf{h} = \mathbf{v}/\|\mathbf{u}\|$. Hence, they can be seen as *normalized Plücker coordinates*[PPR98]. Notice that, due to Euclidean structure, the constraint $\underline{\mathbf{u}}^T\mathbf{h} = 0$ expresses that $\mathbf{h}$ is orthogonal to the interpretation plane of $(\mathcal{L})$. It can moreover be shown that, for any point $\mathbf{P}$ on $(\mathcal{L})$:

$$\mathbf{h} = \mathbf{P} \times \underline{\mathbf{u}} \tag{4}$$

Remark 1 *Considering that the 3D space is oriented, one can easily see that* $(\underline{\mathbf{u}}, \mathbf{h})$ *and* $(-\underline{\mathbf{u}}, -\mathbf{h})$ *are two different oriented lines even though they correspond to the same unoriented line. Moreover, (4) shows that* $(\underline{\mathbf{u}}, \mathbf{h})$ *and* $(-\underline{\mathbf{u}}, \mathbf{h})$ *are symmetric with respect to the origin.*

Remark 2 $\mathbf{h} = 0$ *iff* $(\mathcal{L})$ *passes through the origin.*
Since the case where a line passes through the center of projection is degenerate in computer vision, we will assume in the remainder of this paper that it never occurs.

Remark 3 *Notice that* $(\underline{\mathbf{u}}, \mathbf{h})$ *is a bivector as McCarthy[Mac90] defined it to represent kinematic screws. Consequently[MLS94, p. 427], there does not exist a bi-invariant Riemannian metric on the manifold of 3D lines.*

The Plücker coordinates (both projective and Euclidean) give trivially the image projection (ℓ) of $(\mathcal{L})$, since (ℓ) is simply the intersection of the interpretation plane and the image plane. One can verify that (4) defines an implicit orientation of the 2D space, and that (ℓ) being thus oriented can be defined uniquely by:

$$\ell = \underline{\mathbf{h}} \tag{5}$$

where $\underline{\mathbf{h}} = \mathbf{h}/\|\mathbf{h}\|$. With this new notation and introducing $h = \|\mathbf{h}\|$, the Euclidean Plücker coordinates of a line $(\mathcal{L})$ can be rewritten as:

$$\mathcal{L} = (\underline{\mathbf{u}}, h\underline{\mathbf{h}}) \;\; ; \;\; \underline{\mathbf{u}}^T\underline{\mathbf{h}} = 0 \; ; \; \underline{\mathbf{u}}^T\underline{\mathbf{u}} = 1$$

2.1.2 Binormalized Plücker coordinates

Line parameterization From the previous section, we are now ready to define the binormalized Plücker coordinates, as a way to parameterize 3D lines, and to give some properties of these coordinates.

Definition 1 *The binormalized Plücker coordinates of the pencil of all the 3D lines oriented by the unit vector* $\underline{\mathbf{u}}$ *and lying in the plane of unit normal vector* $\underline{\mathbf{h}}$ *are the couple* $(\underline{\mathbf{u}}, \underline{\mathbf{h}})$ *where*

$$\underline{\mathbf{u}}^T\underline{\mathbf{h}} = 0 \tag{6}$$

$$\underline{\mathbf{u}}^T\underline{\mathbf{u}} = \underline{\mathbf{h}}^T\underline{\mathbf{h}} = 1 \tag{7}$$

By extension, we will call binormalized Plücker coordinates of a 3D line the binormalized Plücker coordinates of the pencil it defines. With this definition, we can trivially state the following proposition:

Proposition 1 (3D line representation) *A 3D line* $(\mathcal{L})$ *is uniquely defined by its binormalized Plücker coordinates* $(\underline{\mathbf{u}}, \underline{\mathbf{h}})$ *and its depth* h*:*

$$\mathcal{L} = \begin{pmatrix} \underline{\mathbf{h}} \\ \underline{\mathbf{u}} \\ h \end{pmatrix} \tag{8}$$

Notice that, since we consider $(\mathcal{L})$ lying in a pencil of parallel lines, h is not only a distance but also becomes a depth.

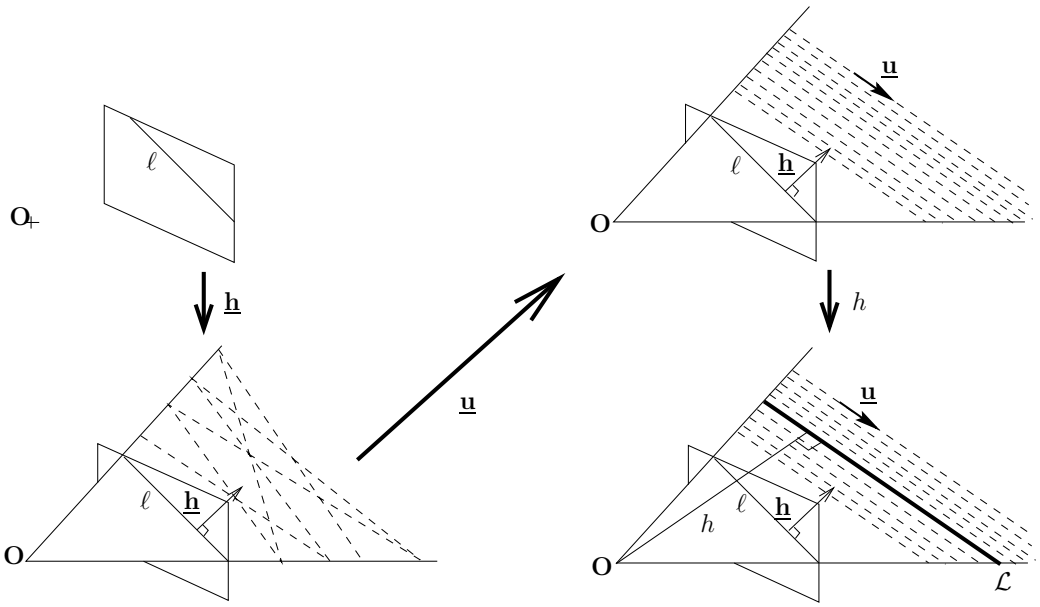

Figure 2: The binormalized Plücker coordinates have a sequential interpretation: from image measurements to the estimation of 3D components.

Geometrical interpretation Our representation yields a sequential interpretation suited to the case of vision (Figure 2): $(\mathcal{L})$ is the line in the interpretation plane of (ℓ) (defined by its normal vector $\underline{\mathbf{h}}$) with orientation $\underline{\mathbf{u}}$ and lying at depth h. This interpretation corresponds to the usual sequence followed in image analysis to determine a 3D line from one or several images: extraction of (ℓ) from the image, then computation of the orientation of $(\mathcal{L})$, and finally, computation of the 3D position of $(\mathcal{L})$.

Before going further let us notice that from the definition of the binormalized Plücker coordinates, we can make the following two remarks:

Remark 4 *The triple* $(\underline{\mathbf{u}}, \underline{\mathbf{h}}, \underline{\mathbf{u}} \times \underline{\mathbf{h}})$ *forms an orthonormal basis of the 3D space.*

Remark 5 *Let* $\mathbf{H}$ *be the closest point to the origin on a line* $\mathcal{L} = (\underline{\mathbf{h}}^T, \underline{\mathbf{u}}^T, h)$. *Then,* $\mathbf{H} = h\underline{\mathbf{u}} \times \underline{\mathbf{h}}$.

Finally, representing lines with binormalized Plücker coordinates and depth allows to generate some well-known geometrical objects from lines, by fixing one or several components to a given value. For instance, the lines with a given depth h generates a sphere of radius h. Figure 3 gathers the other cases.

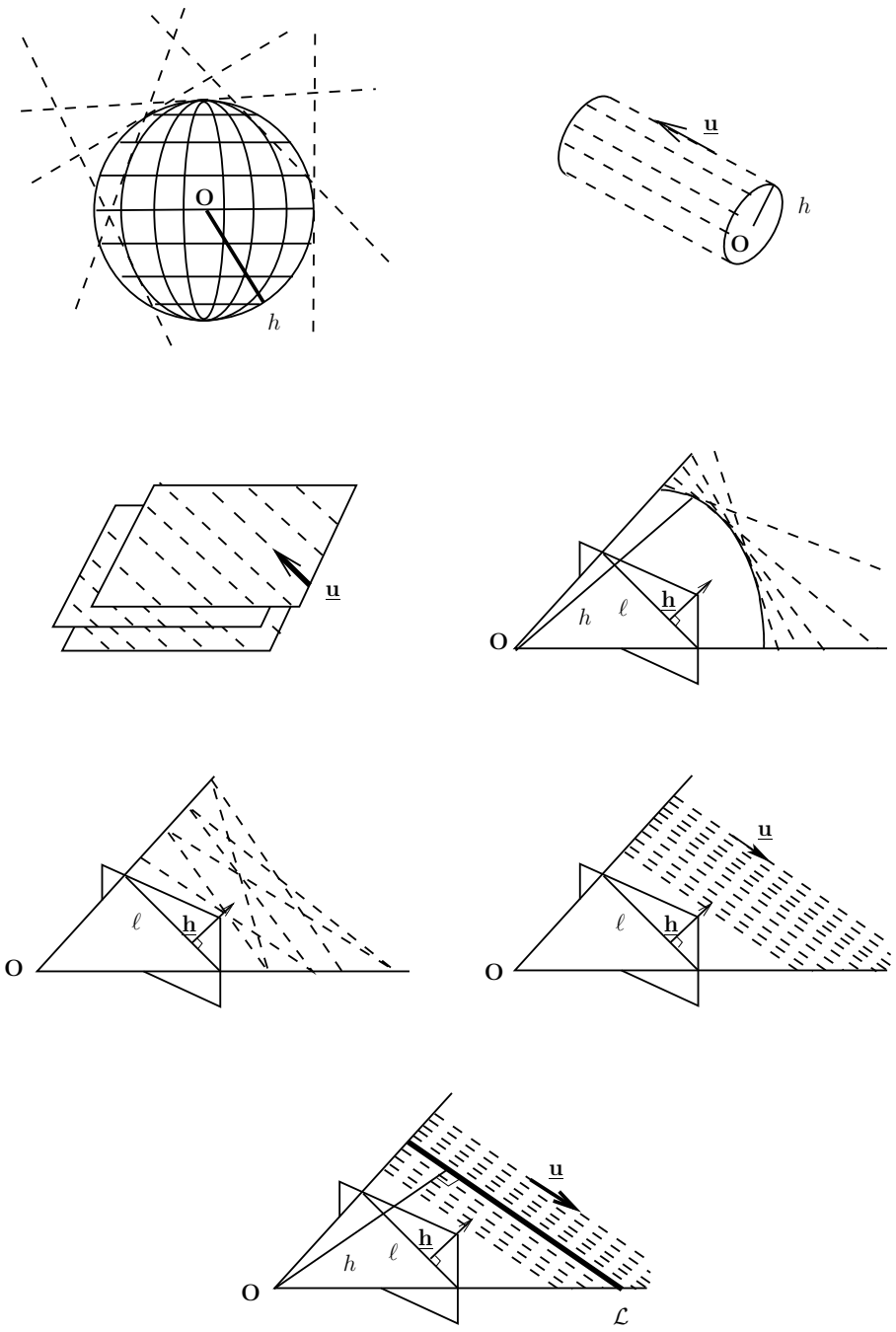

Figure 3: Geometrical objects generated by lines with fixed components of their binormalized Plücker coordinates

2.1.3 Image alignment of a 3D line

Among the previous cases, we are interested in those where the image projection $\underline{\mathbf{h}}$ is given. Indeed, visual servoing consists in defining the goal position of a moving system by an image, in our case, by the projection of a set of lines into the image. On the opposite to the case where visual servoing is based on points, here we do not have a simple answer to the question: "What are the configurations in 3D space that give the same image ?" As seen in Figure 3, there are four cases:

1. only $\underline{\mathbf{h}}$ is given;
2. $\underline{\mathbf{h}}$ and h are given;
3. $\underline{\mathbf{h}}$, $\underline{\mathbf{u}}$ and h are given;
4. $\underline{\mathbf{h}}$ and $\underline{\mathbf{u}}$ are given.

The first case corresponds to the requirement used by Chaumette *et al*[Cha90, ECR92, KMM+96, KRD+98]. The second and third ones are not very interesting since they require the estimation of the depth, always hard to accurately achieve.

In contrast, the fourth case is highly interesting. The 3D line is constrained by its image projection and its alignment on a given direction. Hence, we call this the *image alignment* of a 3D line. It corresponds to fixing the binormalized Plücker coordinates of the 3D line to a given value. As stated before, the only remaining degree of freedom is depth. Hence, this brings us back to the case of points, where the image of a point defines a 1-dimensional manifold in the 3D space.

In addition to this elegant similarity with the case of points, the image alignment matches the property of partial decoupling given below, which allows the definition of a control law with global convergence proof (see section 3).

Nevertheless, in opposition to the case of points, the image alignment of a 3D line is not given by the only image measurements but requires to estimate somehow the orientation $\underline{\mathbf{u}}$ of the line.

2.1.4 Motion of a line vs. motion of the camera

Here, we relate the instantaneous camera motion to the 3D line motion (expressed in our formalism) and recall the apparent motion in the image of this line.

The instantaneous motion of a camera is defined by its velocity screw $\tau = (V, \Omega)$, where Ω is the instantaneous angular velocity and V the instantaneous

linear velocity of a given point. They are usually expressed in the camera frame.

Let $\mathcal{L} = (\underline{\mathbf{h}}^T, \underline{\mathbf{u}}^T, h)^T$ be a 3D line. Then its motion is the vector $(\dot{\underline{\mathbf{h}}}^T, \dot{\underline{\mathbf{u}}}^T, \dot{h})^T$. Rives *et al* [RE87] recall the derivative $(\dot{\underline{\mathbf{u}}}, \dot{\mathbf{h}})$ of the normalized Plücker coordinates $(\underline{\mathbf{u}}, \mathbf{h})$ of a 3D line:

$$\dot{\underline{\mathbf{u}}} = \Omega \times \underline{\mathbf{u}} \tag{9}$$

$$\dot{\mathbf{h}} = \Omega \times \mathbf{h} + V \times \underline{\mathbf{u}} \tag{10}$$

The null space of this mapping is given by

$$\Omega_{ns} = \lambda \underline{\mathbf{u}} \tag{11}$$

$$V_{ns} = \mu \underline{\mathbf{u}} + \lambda \mathbf{h} \tag{12}$$

where μ and λ are real numbers.

From the definition of h, it is now easy to show that:

$$\dot{h} = V^T(\underline{\mathbf{u}} \times \underline{\mathbf{h}}) \tag{13}$$

As for $\dot{\underline{\mathbf{h}}}$, which Navab [Nav93] calls the *line motion field equation* since it can be interpreted as the relation between the camera motion and the apparent motion of a 3D line in the image, it can be expressed either by:

$$\dot{\underline{\mathbf{h}}} = \frac{1}{h}(\mathbf{I}_3 - \underline{\mathbf{h}}\underline{\mathbf{h}}^T)(\Omega \times \mathbf{h} + V \times \underline{\mathbf{u}}) \tag{14}$$

or by the simpler form:

$$\dot{\underline{\mathbf{h}}} = \Omega \times \underline{\mathbf{h}} - \frac{V^T \underline{\mathbf{h}}}{h}(\underline{\mathbf{u}} \times \underline{\mathbf{h}}) \tag{15}$$

Notice that equations (9) and (15) exhibit a partial decoupling between rotation and translation.

2.2 Discussion

The parameterization we propose, based on binormalized Plücker coordinates, has some advantages, from the control point of view, over the parameterizations found in the literature:

- Our representation is continuous and covers the whole space, except the case of lines going through the optical center.
- It is coherent in the 3D space and the image.
- It is unique provided that some constraints, easy to satisfy, hold.
- It exhibits a partial decoupling between rotation and translation.

Details of the following discussion can be found in [AEH01].

2.2.1 In the 3D space

Apart from the Plücker coordinates, one usually finds three other representations of a 3D line.

A line is defined by a point and a direction – This representation is not unique and has no relation with the image. Hence, Hager[Hag97] represents lines with a point and a direction, but comes to the Plücker coordinates as soon as control is involved.

A line is the intersection of two planes – The main drawback of this representation is that it is implicit. Used in [Cha90, ECR92, KMM+96, KRD+98] together with the (ρ, θ) representation of an image line (see below), it yields moreover complex motion equations where rotation and translation are highly coupled.

The (a, b, p, q) representation [Fau93, Nav93, Plü65] – This representation represents the lines that are neither parallel to the optical axis, nor parallel to the image plane. One thus obtains either an incomplete representation or a non-unique and/or discontinuous representation which may have destabilizing effects on the control.

2.2.2 In the image

The usual representations of an image line, ours excepted, are the following.

The unnormalized triple (a, b, c) of its equation coefficients – This representation is not unique and, hence, never used for control.

The normalized triple $(\cos\theta, \sin\theta, \rho)$ – This parameterization, used by Hager[Hag97], corresponds to the normalization of the line equation coefficients (a, b, c) on a 2-dimensional subspace. Hence, $\mathbf{d} = (\cos\theta, \sin\theta, \rho)$ is not a unit vector in the 3D space. This results in an heterogeneity between image and space parameterizations. Moreover, the consequent motion line equation $\dot{\mathbf{d}} = (\mathbf{I}_3 - \mathbf{d}\mathbf{d}^T\mathbf{N})(\Omega \times \mathbf{d} + V \times \underline{\mathbf{u}})$ does not seem to have nice geometric interpretation nor to allow for any decoupling between rotation and translation.

The (ρ, θ) representation – Used in [Cha90, ECR92, KMM+96, KRD+98], this representation has the advantage of being minimal. However, it is periodic and the choice of the definition interval of θ is not generic since it depends on the initial (θ_0) and desired (θ^*) values of θ. Moreover, this *ad hoc* choice does not prevent from possible discontinuities in the control law.

3 Control

In a preliminary way, let us emphasize that, provided that the required measurements are available in real-time, the following developments are not specific of visual servoing. Nevertheless, they are particularly suitable to the use of a camera, and we will focus on this particular case in the following at every time it will be needed

3.1 Introduction

Let us consider a fixed reference frame F_r and a moving object with associated frame F_0. We define a goal frame F^* in F_r and the control problem we consider is to drive F_0 from its initial value $F_0(0)$ to F^* in a stable way. With our representation, and assuming we have enough independent lines to span all the displacement space, our goal will therefore be defined as the set $\{\underline{\mathbf{u}}_i^*, \underline{\mathbf{h}}_i^*, h_i^* \ ; \ i = 1 \dots n\}$.

Note: In the case of vision, F_0 will be the camera frame. The representation we use allows to extract directly 2 parameters from the image, i.e. $\underline{\mathbf{h}}$, the normal to the interpretation plane. The remaining 2 parameters may be considered as 3D ones: the depth h and the last component of $\underline{\mathbf{u}}$. We will see later how to take them into account.

We will firstly demonstrate the principle of the control scheme we propose by using a single line. We will then consider the more general case of the n lines as just defined, and finally focus on the particular situation of the orthogonal trihedron. Before, let us set the assumptions which will be used in all the following.

- According to non-linear control theory[SLBE91], the errors to be regulated are supposed to be small enough to ensure the validity of the approximation of the space by its tangent space and the inversion of the latter for control purposes.
- The set of goal lines $\mathcal{L}_i^*$ corresponds to a fixed achievable configuration of the system (i.e. the displacement between the current position and the goal one is a rigid motion).
- The velocity screw (Ω, V) of the moving frame is the control variable (i.e. dynamical effects are neglected).
- All 3D trajectories of the moving frame which are needed are feasible (no obstacles, no joint limits neither geometrical singularities are reached by the moving device).

- All the needed variables **but the depth** are exactly known or measured. In the case of vision, this means that, we assume that the $\underline{\mathbf{u}}_i$'s are estimated, either analytically in some simple cases[3] or by using generic pose estimation[Che91, CH99, DRLR89, LHF90] or Euclidean reconstruction algorithms[Fau93, VLF95, Har94, QK97, LH86, SA90]. Notice that these generic methods give more than the minimal information required by our method and that there may exist simplifications to them that would give us exactly what we need.

More, when using a camera, two further assumptions are required:

- During all the motion, no line ($\mathcal{L}_i$) goes through the optical center of the camera.
- The camera is fully calibrated and the hand-eye calibration was performed [AHE01].

3.2 The Example of a Single Line

Let us consider a single target line specified by the set $\{\underline{\mathbf{u}}^*, \underline{\mathbf{h}}^*, h^*\}$ and forget for the moment the depth h. The state equations of this subsystem are given by eqs (9) and (15) only. We may remark that the orientation part is independent from the translation part, which allows to consider that this system is in cascaded form. Since the manifold of lines does not exhibit natural metric we have to define convenient errors. By exploiting the above mentioned partial decoupling, we set

$$\mathbf{e_u} = \underline{\mathbf{u}} \times \underline{\mathbf{u}}^* \tag{16}$$

and

$$\mathbf{e_h} = \underline{\mathbf{u}} \times (\underline{\mathbf{h}} - \underline{\mathbf{h}}^*) \tag{17}$$

We can therefore set the following cascaded control:

$$\begin{cases} \Omega = \mu_u \mathbf{e_u} \\ V = -\mu_h \mathbf{e_h}|_{\mathbf{\Omega=0};\underline{\mathbf{u}}=\underline{\mathbf{u}}^*} \end{cases} \tag{18}$$

where $\mu_u > 0$, $\mu_h > 0$.

[3]For instance, when observing two parallel lines, the common direction $\underline{\mathbf{u}}$ is immediately given by their vanishing point. Alternately, when a line lies in a plane with a known orientation (line on the ground, vertical line), its direction is simply the direction orthogonal to both the known plane orientation and the interpretation plane associated to the line. An application of the use of the binormalized Plücker coordinates to stabilize an unmanned aerial vehicle with respect to a collection of parallel lines can thus be found in [MH01].

Remark 6 $\mathbf{e_u}$ *is the classical error on the sphere; it allows to move on a geodesics.*

Remark 7 $\mathbf{e_h}$ *is a better choice than simply* $(\underline{\mathbf{h}} - \underline{\mathbf{h}}^*)$ *since it can be shown that it ensures that* $\underline{\mathbf{h}} \times \underline{\mathbf{h}}^*$ *is minimum even if* Ω *vanishes without the right equilibrium reached* ($\underline{\mathbf{u}}_{eq} \neq \underline{\mathbf{u}}^*$) *(see [And99]). Moreover, since* $\underline{\mathbf{h}}$ *is a unit vector, it is constrained to move on a sphere. Thus,* $\mathbf{e_h}$ *allows to move on this sphere using translational degrees of freedom only (figure 4).*

Remark 8 *The evolution equation of (15) under the control (18) is:*

$$\dot{\underline{\mathbf{h}}} = \frac{\mu_h}{h} \mathbf{e_h}^T \underline{\mathbf{h}} (\underline{\mathbf{u}} \times \underline{\mathbf{h}}) \tag{19}$$

This shows that the unknown depth[4]*, h, acts only as a variable unknown positive gain, which has no consequence on the stability in continuous time.*

Remark 9 *Since we don't consider h, the null space of the system is 3-dimensional. Instead of (11), (12), it is given by*

$$\Omega_{ns} = \lambda \underline{\mathbf{u}} \tag{20}$$

$$V_{ns} = \mu \underline{\mathbf{u}} + \lambda h \underline{\mathbf{h}} + \nu (\underline{\mathbf{u}} \times \underline{\mathbf{h}}) \tag{21}$$

where μ*,* λ *and* ν *are real numbers. Any motion within this subspace will not affect the stabilization.*

Considering the latter remark, we notice that the translation control in (18) may have components in the two dimensions of the null space (μ,ν) that are independent from the rotation control. These motions are useless disturbances unless they are used for complementary tasks (see section 4.1). It is therefore pertinent to project the linear velocity on the only relevant component, i.e. $\underline{\mathbf{h}}$ when $\Omega = 0$. Hence, we obtain the final cascaded control for one line:

$$\begin{cases} \Omega = \mu_u \mathbf{e_u} \\ V = -\mu_h \epsilon_h \underline{\mathbf{h}}|_{\Omega=0;\underline{\mathbf{u}}=\underline{\mathbf{u}}^*} \end{cases} \tag{22}$$

where $\epsilon_h = (\underline{\mathbf{u}} \times \underline{\mathbf{h}})^T \underline{\mathbf{h}}^*$, $\mu_u > 0$, $\mu_h > 0$. We have the following result:

Proposition 2 *The control (22) is such that:*
1- if $\underline{\mathbf{u}}(t=0) \neq -\underline{\mathbf{u}}^*$*, then* $\underline{\mathbf{u}}(t)$ *asymptotically converges to* $\underline{\mathbf{u}}^*$*;*
2- if $\underline{\mathbf{h}}(t=0|_{\Omega=0;\underline{\mathbf{u}}=\underline{\mathbf{u}}^*}) \neq -\underline{\mathbf{h}}^*$*, then* $\underline{\mathbf{h}}(t)$ *asymptotically converges to* $\underline{\mathbf{h}}^*$*.*

The proof is easy (see [And99]) using $\frac{1}{2}||\underline{\mathbf{u}} - \underline{\mathbf{u}}^*||^2$ and $\frac{1}{2}||\underline{\mathbf{h}} - \underline{\mathbf{h}}^*||^2$ as Lyapunov functions respectively.

[4]Recall that h can be considered as a depth § 2.1.2.

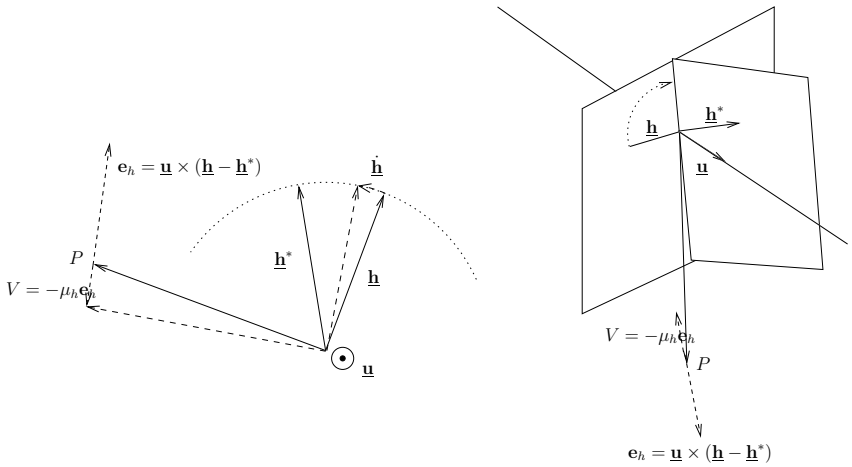

Figure 4: The action of the translation control $V = -\mu_h \underline{\mathbf{u}} \times (\underline{\mathbf{h}} - \underline{\mathbf{h}}^*)$: in the case where $\underline{\mathbf{u}} = \underline{\mathbf{u}}^*$ (left) and in the general case (right).

In practice, the stabilization in orientation can be considered as "sufficiently" achieved after a finite time T_Ω. The translation control then writes as

$$V = -\mu_h s(t - T_\Omega) \epsilon_h \underline{\mathbf{h}} \tag{23}$$

where $s(t)$ is the Heaviside step function, and the convergence condition becomes $\underline{\mathbf{h}}(t = T_\Omega) \neq -\underline{\mathbf{h}}^*$.

Remark 10 *In practice, one may challenge this control law by asking for regulating an initial rotation error out of the small error assumption. This would then cause trouble to the cascaded control scheme since the line may leave the field of view*[5]*. To cope with this, the two control vectors may not be fully cascaded: the control in orientation should be activated all the time, because of unmodelled coupling effects between translational and rotational motions; the control in translation can be progressively activated using a transition function smoother than the step one, in order to avoid unexpected large motions. This is the reason why we wrote* $\underline{\mathbf{u}}$ *instead of* $\underline{\mathbf{u}}^*$ *in the proposition..*

Remark 11 *The forbidden initial conditions in the proposition are unstable equilibriums. However, since they correspond to the case where the line is in the correct position but reverse orientation,* $(\underline{\mathbf{u}}, \underline{\mathbf{h}}) = (-\underline{\mathbf{u}}^*, -\underline{\mathbf{h}}^*)$ *(see Remark 1), they are easy to detect and avoid.*

[5]Notice that panoramic cameras do not suffer from this and can be used in conjunction with the sequential control.

3.3 General Case

We now consider the case of a set of $n > 1$ distinct lines, rigidly linked, and for the moment we do not assume further properties on their configuration. Let us recall that the equations of their motion are, for $i = 1 \dots n$:

$$\dot{\underline{\mathbf{u}}}_i = \Omega \times \underline{\mathbf{u}}_i \tag{24}$$

$$\dot{\underline{\mathbf{h}}}_i = \Omega \times \underline{\mathbf{h}}_i - \frac{V^T \underline{\mathbf{h}}_i}{h_i}(\underline{\mathbf{u}}_i \times \underline{\mathbf{h}}_i) \tag{25}$$

Again, we left the h_i's, the dynamics of which are given by eq (15), free to evolve. Like previously, we define an ideal cascaded control, allowing us therefore to split it in two parts: the rotational, then the translational, provided that the former is stabilized.

3.3.1 Control in Rotation

It is a straightforward generalization of the single line case which results in

Theorem 1 *(proof in appendix A.1)*
If $\underline{\mathbf{u}}_i(t=0) \neq -\underline{\mathbf{u}}_i^*, \ \forall i = 1 \dots n,$
Then, the control

$$\Omega = \mu_u \sum_{i=1}^{n} \underline{\mathbf{u}}_i \times \underline{\mathbf{u}}_i^*, \quad \mu_u > 0 \tag{26}$$

asymptotically stabilizes the equilibrium of the system (24), $i = 1 \dots n$*:*

$$\underline{\mathbf{u}}_i = \underline{\mathbf{u}}_i^*, \quad i = 1 \dots n \tag{27}$$

3.3.2 Control in Translation

The translational control presented for a single line can be generalized in a similar way. The following result holds:

Theorem 2 *(proof in appendix A.2)*
If for all $i = 1..n$*, the vectors* $\underline{\mathbf{u}}_i$ *are constant (but not necessarily equal to the vectors* $\underline{\mathbf{u}}_i^*$*),*
Then:

1. *the control*

$$V = -\mu_h \sum_{i=1}^{n} \mathbf{B}_i \epsilon_i \underline{\mathbf{h}}_i \mid_{\Omega=0;\underline{\mathbf{u}}_i=\underline{\mathbf{u}}_i^* \forall i}, \quad \mu_h > 0 \tag{28}$$

$\mathbf{B}_i$ *being* (3×3) *weighting matrices stabilizes equilibrium points under the condition:*

$$\left(\sum_{i=1}^{n} \mathbf{B}_i \epsilon_i \underline{\mathbf{h}}_i \right)^T \left(\sum_{i=1}^{n} \frac{\epsilon_i \underline{\mathbf{h}}_i}{h_i} \right) > 0 \tag{29}$$

2. *the choice* $\mathbf{B}_i = \frac{1}{h_i}\mathbf{I}_3, \forall i = 1..n$*, which corresponds to the control:*

$$V = -\mu_h \sum_{i=1}^{n} \frac{1}{h_i} \epsilon_i \underline{\mathbf{h}}_i \mid_{\Omega=0;\underline{\mathbf{u}}_i=\underline{\mathbf{u}}_i^* \forall i}, \quad \mu_h > 0, \tag{30}$$

stabilizes the stationnary points:

$$\sum_{i=1}^{n} \frac{\epsilon_i \underline{\mathbf{h}}_i}{h_i} = 0 \tag{31}$$

whatever the line configuration.

However, the analytical expression of the stationnary points cannot have an easy geometrical interpretation in any line configuration. Additionnal knowledge on the system geometry is therefore required. We will see in the following that the choice of a particular object, the orthogonal trihedron, will allow us to obtain results stronger than in the general case.

3.4 Case of the Orthogonal Trihedron

The trihedral configuration of lines is of particular interest since it is very often present in structured environments like in buildings or in manufactured systems. However, from the image point of view, it has been shown (see [DRLR89]) that any pencil of lines (i.e. with a common intersection) is degenerated for pose computation. Indeed, for example, it is intuitive that observing an ideal infinite trihedron with a single camera doesn't allow to recover its depth: imagine that the projected trihedron is centered in the image; then any translation along the optical axis leaves the projection unchanged. In the next section, we will propose a practical solution to this problem. For the moment, let us study the control without taking the depth into account. The final equilibrium will therefore have a single degree of freedom left.

With respect to the general case, the following constraints have to be considered for the orthogonal trihedron:

- $n = 3$
- $\underline{\mathbf{u}}_i^T \underline{\mathbf{u}}_j = 0 \ \forall \ i \neq j$ (orthogonality)
- $\mathbf{h}_i^T \underline{\mathbf{u}}_j + \mathbf{h}_j^T \underline{\mathbf{u}}_i = 0 \ \forall \ i, j$ (intersection)

Taking these constraints into account allows to set the following results

- Rotation: theorem 1 is still true.
- Translation: It can be shown (cf [And99]) that the stationnary points of the control (30) are the configurations where a plane of the trihedron intersects the origin of the moving frame (here the optical center of the camera). As a consequence, the control is asymptotically stable provided that the trajectory of the moving frame stays in a single octant.

However, it can be shown that the control (30) is inversely proportionnal to the depth of the trihedron center. Since this depth can not be observed, we can set it to an arbitrary value (for instance, the desired depth after convergence). But, in the particular case of the orthogonal trihedron a simpler control law can be used:

Theorem 3 *In the case of the orthogonal trihedron and if* $\underline{\mathbf{h}}_i(t = 0) \neq -\underline{\mathbf{h}}_i^*$, *then the following control*

$$V = -\mu_h \sum_{i=1}^{n} \epsilon_i \underline{\mathbf{h}}_i \ |_{\Omega=0; \underline{\mathbf{u}}_i = \underline{\mathbf{u}}_i^* \forall i}, \quad \mu_h > 0 \tag{32}$$

is independent from the depth and asymptotically stabilizes the equilibrium of the system (25)
The proof of this theorem is given in Appendix A.3.

Let us mention that, like in the case of a single line, the practical implementation of these control schemes can be achieved using some overlapping of the rotational and translational parts. Simulations of section 4 will provide some results concerning this matter.

4 Practical Achievements

Let us now consider the practical application presented in the introduction: positioning a robot with respect to an orthogonal trihedron. With respect to the theoretical control schemes previously presented, the following problems had to be solved in order to achieve the implementation:

1. how to cope with the missing depth?
2. how to extract and track the lines in the image in a fast and robust way?
3. how to compute the orientations $\underline{\mathbf{u}}_i$?

We briefly present in the following the solutions we choose for all these issues, before giving some simulation and experimental results.

4.1 Adding Depth Control

Dhome *et al.*[DRLR89] showed that a line junction is a degenerated case of pose estimation in the sense that it is impossible to estimate the distance from the camera center point to the *junction center* (i.e. the intersection of the 3D lines) from a single image. Thus, one can only estimate a *partial pose* of the camera: the 3D orientation of the lines and the line of sight passing through the junction center.

From the control point of view, this degeneracy means that the distance from the robot to the ship part cannot be measured neither controlled. Consequently, we need to add some other sensor to the camera in order to observe the depth of the junction center.

The original solution we chose was to use a commercial laser pointer (cheap and simple) that we rigidly linked to the camera (figure 5). In this configuration the image of the laser spot moves on a given straight line whatever the observed scene and the motion of the set {camera-laser}. This can be easily understood if we consider the laser beam as the optical axis of a virtual camera. The line in the image is therefore an epipolar line of this particular stereo rig.

Remark 12 *We will* not *consider that the laser-camera system is calibrated. We will only need to know the laser epipolar line, which is straightforward to obtain by recording the image spot position while moving, whatever the target.*

Let us now address the problem of depth control. In fact, we have, up to now, one dimension free. We therefore need to add to the control scheme one

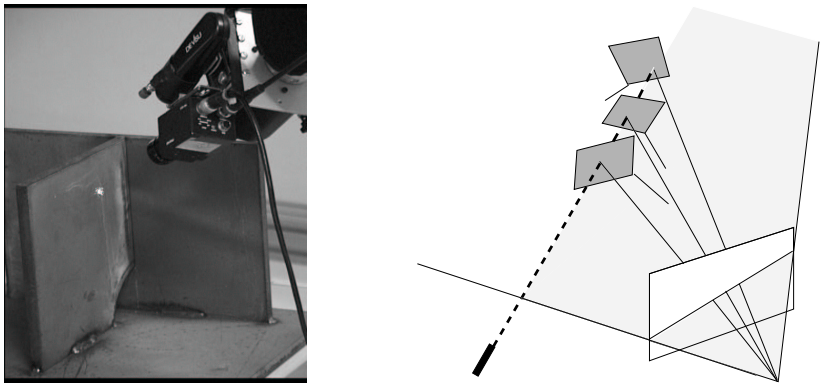

Figure 5: A laser pointer rigidly linked to the camera generates an epipolar line in the image.

variable only, which should be an image of the depth. Since the image laser spot moves on a line, it looks natural to choose the signed distance of the spot from a given origin of the line, denoted as s, as the variable to control. When the target is a plane, s is a monotonic function of the true depth. In order to design the control, we now need to know the variation of s, i.e its interaction screw. We have the following results:

Proposition 3 *If the laser spot moves in a single 3D plane, then the variation of s is:*

$$\dot{s} = -\frac{h_L \cos\theta_z}{z_L^2} \begin{pmatrix} \tan\alpha_x & \tan\alpha_y & 1 & -\frac{z_L}{\cos\theta_z}\tan\alpha_y & \frac{z_L}{\cos\theta_z}\tan\alpha_x & 0 \end{pmatrix} \Theta_L \quad \tau \tag{33}$$

where τ is the velocity screw (V, Ω), Θ_L is the adjoint operator involving the rigid transformation R_L, t_L between the camera frame and the laser frame, z_L is the depth of the laser spot along its view line, α_x and α_y are two angles defining the normal to the 3D plane in the laser frame, h_L is the distance between the camera center of projection to the intersection between the laser beam and the image plane, and θ_z is the angle between the laser beam and the optical axis.

Notice that this equation can be derived from eq. (7.1.48) in [SLBE91, pp 273–274], obtained while defining the interaction screw of general thin-field range-sensors. An alternative proof of those results is given in [And99], with explicit references to the present case.

Proposition 4 *In addition, if the rotational velocity is zero, and if the translational velocity is such that $V_l = v\underline{\mathbf{u}}_l$ where $\underline{\mathbf{u}}_l$ is a constant unit vector, then*

$$\dot{s} = \frac{v}{z^2}\left(-h_L k_z (\tan\alpha_x \quad \tan\alpha_y \quad 1 \quad) R_l \underline{\mathbf{u}}_l\right) \tag{34}$$

where the term in brackets is a constant, noted k.

From this result, it is now straightforward to design the depth control. In order to take benefit of the last proposition, the depth control will be also cascaded with the rotational one. Let us therefore assume that the desired orientation has been reached. Then, $\Omega = 0$, even though it is highly advisable to keep this control activated in order to ensure that the orientation remains constant all the time despite disturbances. Let us choose as $\underline{\mathbf{u}}_l$ the direction of the view line going through the trihedron center. An image of the depth error is $e_s = s - s^*$ where s^* is the desired image spot position on the laser epipolar line, and with, from (34), $\dot{e}_s = kv/z^2$. An ideal exponential behavior would be reached by choosing v as

$$v = -\lambda \frac{z^2}{k}(s - s^*), \quad \lambda > 0 \tag{35}$$

Since z is unknown, but is always nonzero, it is possible to fix its value to an arbitrary one. Stability will not be affected, but the convergence will no more be exponential. The depth control part is finally:

$$V_l = -\frac{\mu_l}{k}(s - s^*)\underline{\mathbf{u}}_l, \quad \mu_l > 0 \tag{36}$$

It remains only now to integrate this control in the translational one. Recall that we had the following control:

$$V_h = -\mu_h \sum_{i=1}^{3}((\underline{\mathbf{u}}_i \times \underline{\mathbf{h}}_i)^T \underline{\mathbf{h}}_i^*)\underline{\mathbf{h}}_i, \quad \mu_h > 0 \tag{37}$$

The final translational velocity control is simply $V = V_h + V_l$. It can indeed be seen that the control V_h is a combination of the $\underline{\mathbf{h}}_i$'s, which are all orthogonal to $\underline{\mathbf{u}}_l$ owing to the trihedron geometry (intersection). Therefore V_l is in the null space of V_h and can run simultaneously with it.

4.2 Line Extraction and Tracking

To extract the lines and track them in the image along the motion, we used an adapted version of the 2D-3D model-based approach proposed in [MBCM99] and which is summarized by their authors as: "*in a first step, the object image motion is represented by a 2D affine motion model, and is estimated, using a robust statistical method, from the computation of the normal displacements evaluated along the projected model contours. [...] The 2D affine motion model does not always match the real displacement of the object. A second step that consists in fitting the projection of the object model on the*

intensity gradients in the image is necessary. This is achieved using an iterative minimization of a non-linear energy function with respect to 3D pose parameters."

Notice that our control law does not require to compute a full pose and was developed independently from this tracker. Hence, we intentionally did not take advantage of the pose given by the tracker. Any line tracker (even only image-based) can thus be used in our practical application. For instance, we also used successfully, in a preliminar extension of our control scheme to a more complex line configuration, the tracker developped by Martin and Horaud[MH02].

A question which arises concerns the combinatorics of the initialization process. Even though the initialization is made by hand, by clicking on the three lines in the right order, it is easy to see how it can be automated. In our application, only physical half-lines are visible in the image, thus giving a natural orientation to the $\underline{\mathbf{u}}_i$'s from the vertex outwards. This yields 6 combinations. Then, knowing whether the trihedron is convex or concave, only three of these cases are such that the three lines form a direct trihedron. The correspondence is thus established by choosing the correct first line. Some *a priori* knowledge (recall the small error assumption) or low-level image processing such as correlation should do the latter.

4.3 Computation of the Line Orientations

To compute the line orientations, we based ourselves on the analytic solution to the perspective 4 point problem proposed in [HCLL89]. Indeed, the orthogonal trihedron is a particular case of the 4 point problem where 3 of the points lie at the infinite.

Thus, using $\underline{\mathbf{u}}_l$ the unit vector of the view line going through the trihedron center, we define $\underline{\mathbf{v}}_i = \underline{\mathbf{u}}_l \times \underline{\mathbf{h}}_i$. Recall that the $\underline{\mathbf{h}}_i$'s can be expressed as:

$$\underline{\mathbf{h}}_i = \frac{\underline{\mathbf{u}}_l \times \underline{\mathbf{u}}_i}{\|\underline{\mathbf{u}}_l \times \underline{\mathbf{u}}_i\|}, \quad i = 1..3 \tag{38}$$

Consequently, if we give the same orientation to the 3D lines and to their projections, we obtain the following system:

$$\underline{\mathbf{u}}_i = \cos\theta_i \underline{\mathbf{u}}_l + \sin\theta_i \underline{\mathbf{v}}_i, \quad i = 1..3, \quad \sin\theta_i > 0$$

where the unknowns are the angles $\theta_i, i = 1..3$. This system is similar to the

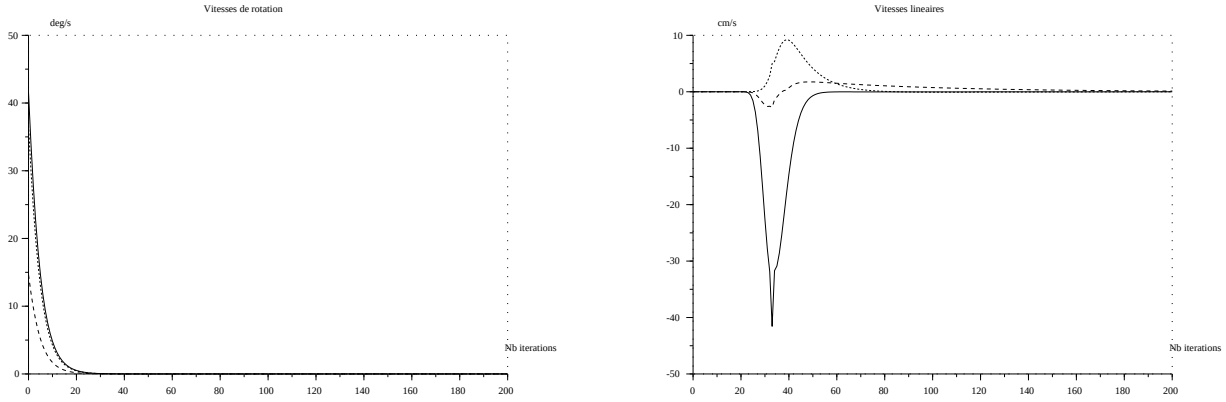

Figure 6: Sequential activation of the control in rotation and then in translation: Rotation (left) and translation (right) control.

one obtained in [HCLL89] and its solution is of the form:

$$\cos\theta_i = \epsilon_i \sqrt{\frac{\underline{\mathbf{v}}_{i-1}^T \underline{\mathbf{v}}_{i+1}}{\underline{\mathbf{v}}_{i-1}^T \underline{\mathbf{v}}_i \underline{\mathbf{v}}_i^T \underline{\mathbf{v}}_{i+1} + \underline{\mathbf{v}}_{i-1}^T \underline{\mathbf{v}}_{i+1}}}$$

$$\sin\theta_i = \sqrt{\frac{\underline{\mathbf{v}}_{i-1}^T \underline{\mathbf{v}}_i \underline{\mathbf{v}}_i^T \underline{\mathbf{v}}_{i+1}}{\underline{\mathbf{v}}_{i-1}^T \underline{\mathbf{v}}_i \underline{\mathbf{v}}_i^T \underline{\mathbf{v}}_{i+1} + \underline{\mathbf{v}}_{i-1}^T \underline{\mathbf{v}}_{i+1}}}$$

where $\epsilon_i = \pm 1$ depending on whether the visible half of the 3D line points inwards or outwards. In the case we deal with, where the trihedron faces are opaque, this sign is identical for all the 3 lines (positive for a concave trihedron, negative for a convex one). Notice finally that to simplify the notation, the operations $i+1$ and $i-1$ represent the addition and subtraction modulo 3.

5 Simulation Results

We present here some simulation results to show the behavior of the control law we proposed with both the use of lines and the depth control. We compare the sequential activation of the control in rotation and then in translation (Figures 6, 7 and 8) to their simultaneous activation (Figure 9, 10 and 11). In both cases, depth control is always activated after the desired partial pose is obtained.

Let us first consider the case of the sequential activation. In a first time, only rotation control is activated (Figure 6). It results in an exponential

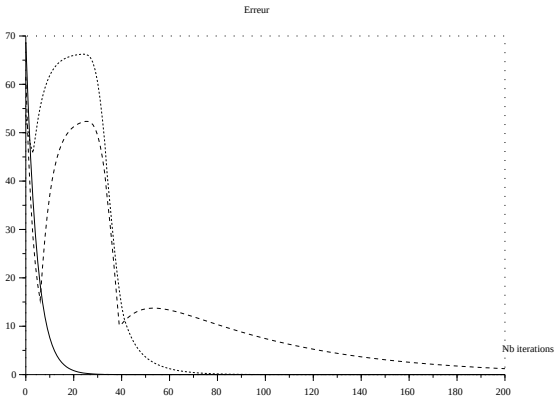

Figure 7: Sequential activation of the control in rotation and then in translation: Errors in orientation (solid line), on the image line projections (dotted line) and on the laser spot position (dashed line).

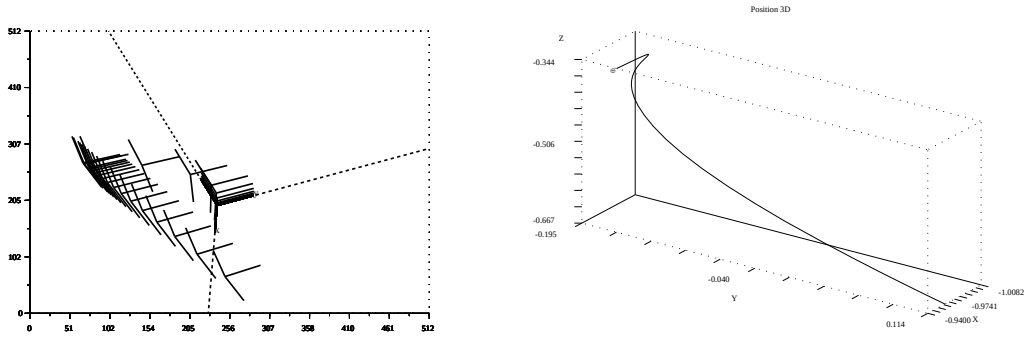

Figure 8: Sequential activation of the control in rotation and then in translation: Trajectory of the trihedron in the image (left) and in the 3D space (right).

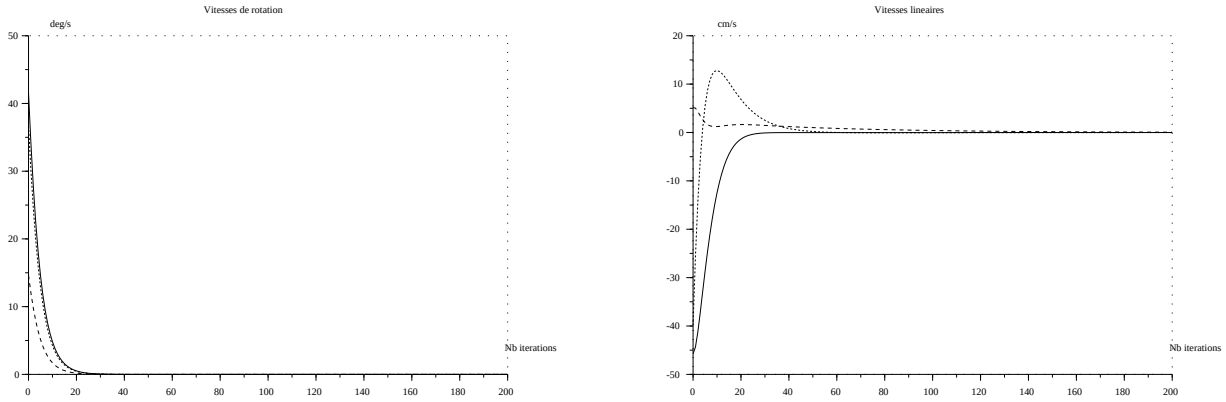

Figure 9: Simultaneous activation of the control in rotation and then in translation: Rotation (left) and translation (right) control.

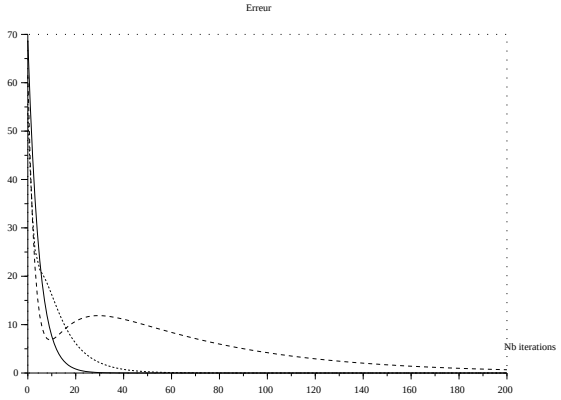

Figure 10: Simultaneous activation of the control in rotation and then in translation: Errors in orientation (solid line), on the image line projections (dotted line) and on the laser spot position (dashed line).

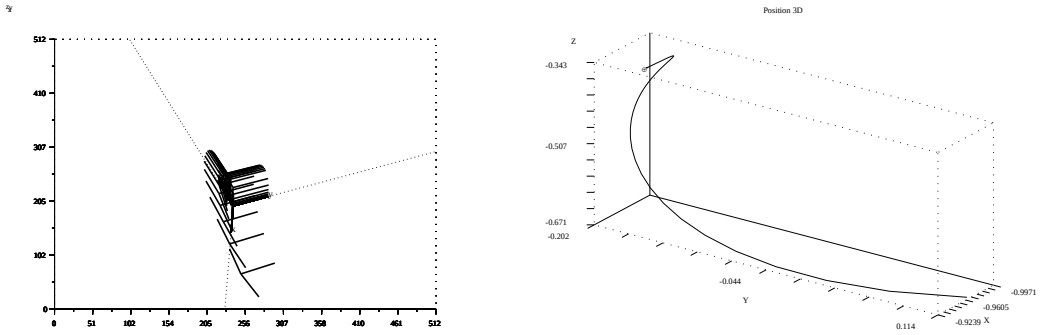

Figure 11: Simultaneous activation of the control in rotation and then in translation: Trajectory of the trihedron in the image (left) and in the 3D space (right).

decay of the error in orientation together with an increase of the errors on the image line projections and on the laser spot positions (Figure 7). Then, control in translation is activated, yielding the convergence of the image line projection errors towards 0, without any influence on the orientation. This can be seen in the image (Figure 8): the trihedron drifts to the left while orientation control is activated and then is brought back to its desired position (large trihedron) by the control in translation. Finally, depth control is activated, bringing the laser spot to its desired position in the image (Figure 7), without the partial pose to be modified. One can verify (Figure 8) that this depth control yields a straight line in the 3D space.

Simultaneous activation of the control in rotation and translation can be easily seen in Figure 9. The partial decoupling of the control law makes that the convergence in orientation is unchanged. On the opposite, the simultaneous activation suppresses the drifting of the trihedron (Figure 10). This results in a smaller amplitude of the image motion but in a larger one in the 3D space (Figure 11). Finally, one can see that the desired partial pose is obtained sooner than in the sequential case and that the depth control is thus activated sooner (Figure 10).

6 Experimental Results

In this section, we focus on the experimental validation of the control law bases on lines. Thus, we disabled the depth control using the laser beam and we simplified the visual tracking by using a trihedron with sharp contrast rather than the ship part.

In all the experiments we led, the task was to bring back the camera to a prerecorded position.

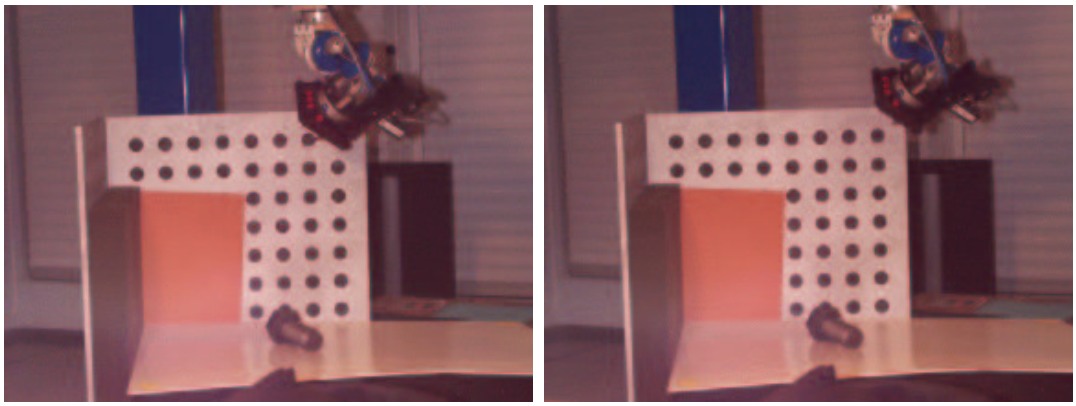

Figure 12: Pure translation: Desired (left) and initial (right) positions of the robot.

6.1 Pure translation

In the first experiment, we took the camera some ten centimeters away from its goal position (Figure 12). Notice first that during the realization of the task, a slight tracking error occurred (Figure 13, left), which yields perturbations on the following curves. Notice also that the computation of the 3D line orientations is stable since the tracking error does not affect it much (Figure 13, right). Moreover, these orientations are constant, yielding a zero control in orientation (Figure 14, right). Hence, the controlled motion is a pure translation as expected. This exhibits the partial decoupling of our control law between translation and rotation. The convergence in translation is here exponential (Figure 14, left). As expected the error in the image ($\sum_{i=1}^{3} \|\underline{\mathbf{h}}_i - \underline{\mathbf{h}}_i^*\|$) decreases and the orientation error is zero up to noise (Figure 15, left). Finally, notice the almost straight image trajectory of the trihedron center (Figure 15, right).

6.2 Pure rotation

In the same experimental conditions, we rotated the camera 100deg along its optical axis away from its desired position. One can notice that the orientation control operates, as expected, solely along the optical axis (Figure 16, left) and yields an asymptotically stable translation control (Figure 16, right), which keeps the trihedron visible (Figure 17, right).

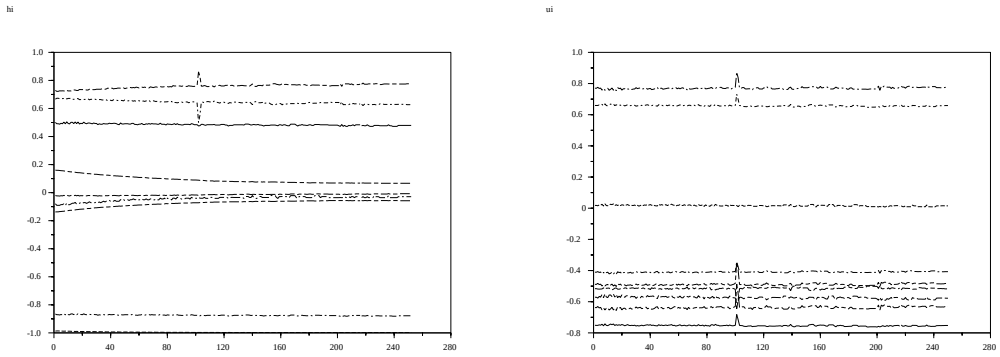

Figure 13: Pure translation: coefficients of the vectors $\underline{\mathbf{h}}_i$ (left) and $\underline{\mathbf{u}}_i$ (right) along the control.

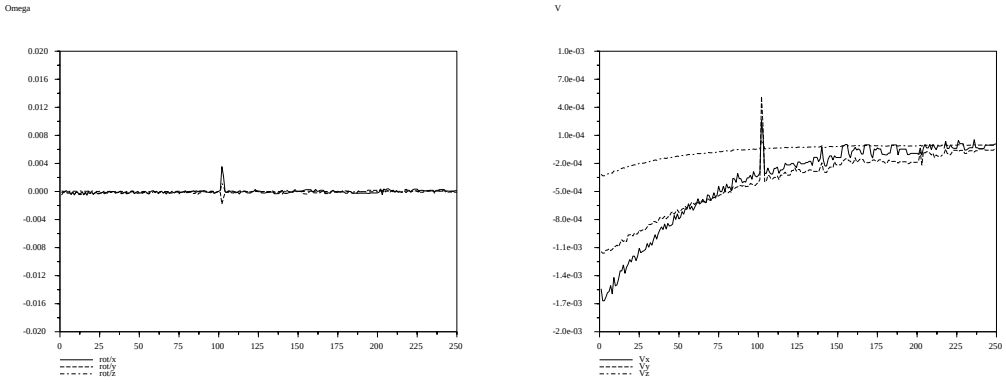

Figure 14: Pure translation: Rotation (left) and translation (right) control.

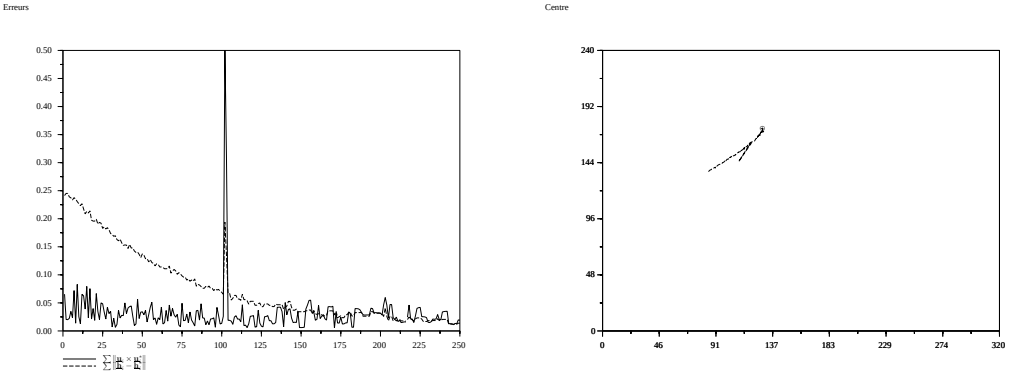

Figure 15: Pure translation: Errors (left) and image trajectory of the trihedron center (right).

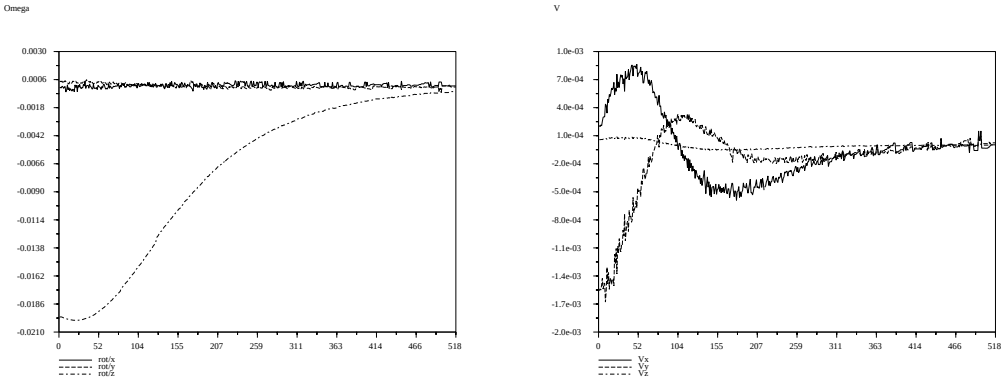

Figure 16: Pure rotation: Rotation (left) and translation (right) control.

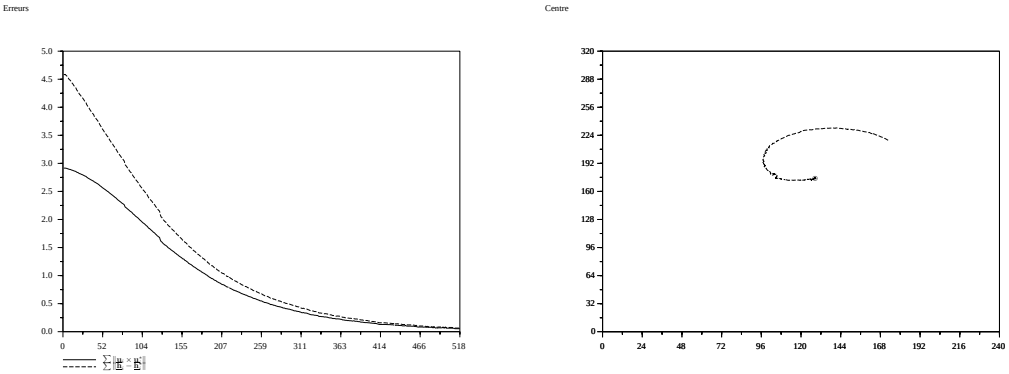

Figure 17: Pure rotation: Errors (left) and image trajectory of the trihedron center (right).

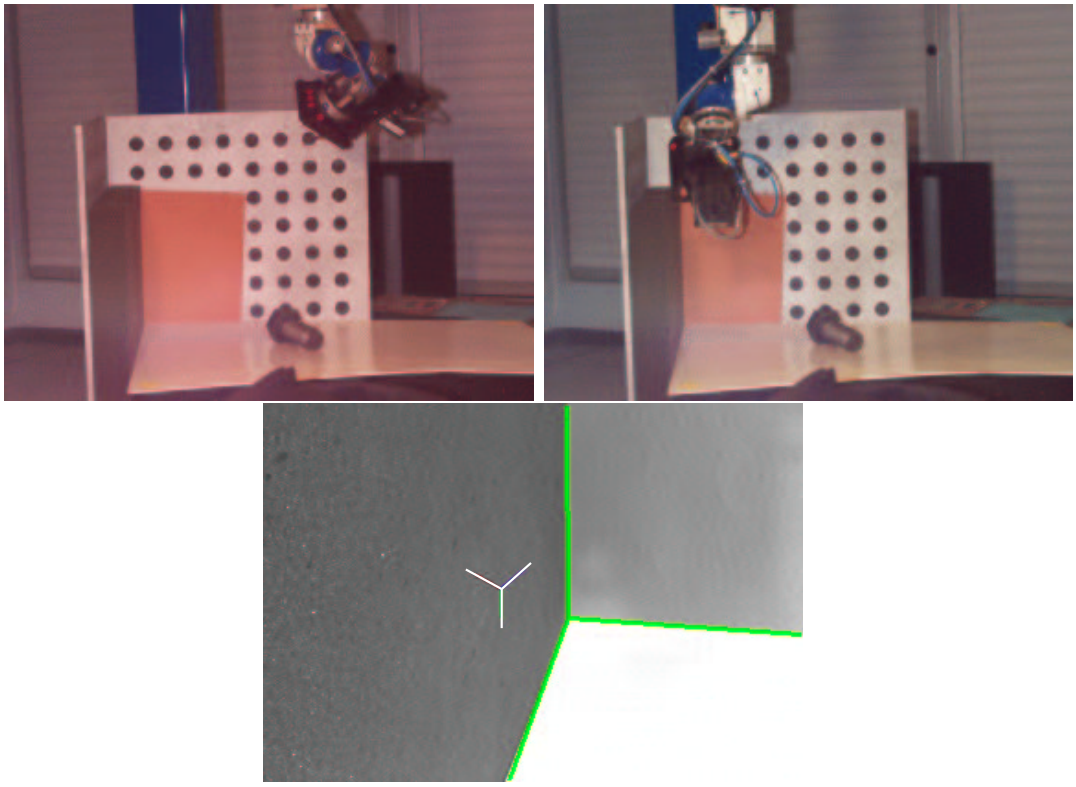

Figure 18: Complex motion. Top left: robot desired position. Top right: robot initial position. Bottom: initial image (superimposed: desired image of the trihedron).

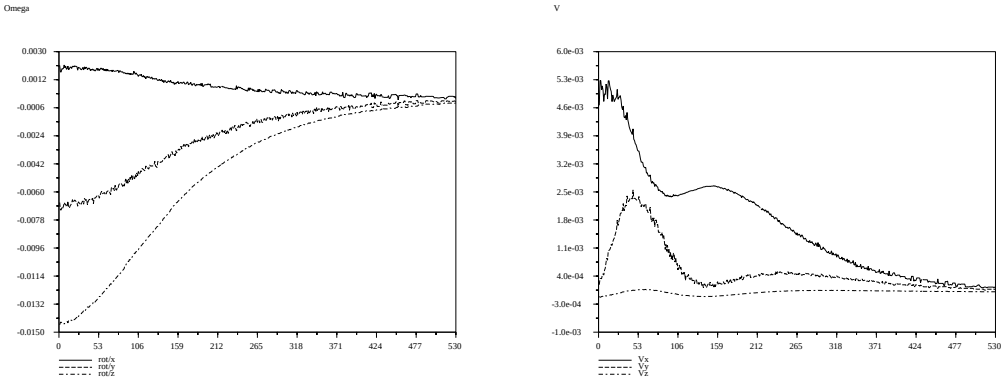

Figure 19: Complex motion: Rotation (left) and translation (right) control.

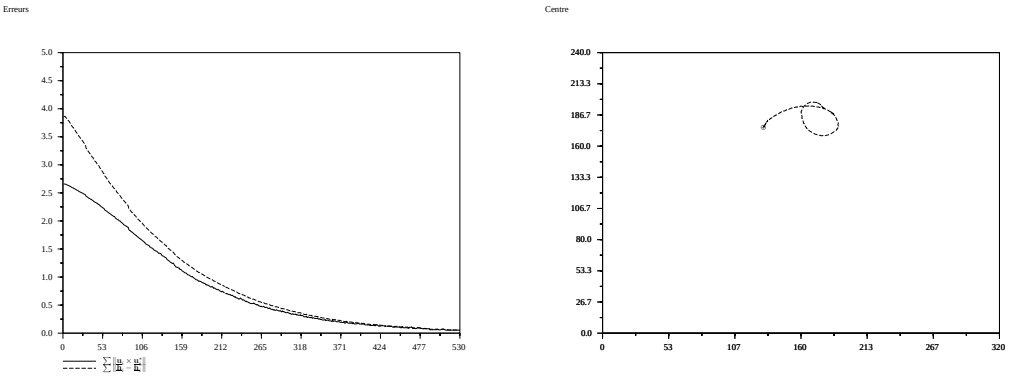

Figure 20: Complex motion: Errors (left) and image trajectory of the trihedron center (right).

It is noticeable that the rotation control does not immediately converge exponentially but starts with a slight acceleration (Figure 16, left). Nevertheless, both orientation error and error in the image decrease (Figure 17, left). Indeed, the residual angle between the initial and desired orientations is larger than $\pi/2$. Hence, the orientation control, which is of the form $\underline{\mathbf{u}} \times \underline{\mathbf{u}}^*$ and thus proportional to the sine of the residual angle, will increase in norm until the residual angle reaches $\pi/2$ and then will decrease exponentially.

6.3 Complex motion

We end up this experimental section with a complex motion, in the sense that it is composed of a translation of some 50cm and a rotation of about 80deg (Figure 18). Since the initial camera position is close to one the trihedron edges, it is challenging both for the tracking and the control algorithms. Indeed, the tracking algorithm must avoid confusing the two image lines that are close to each other. As for the control point of view, the difficulty comes from the fact that this initial position is close to a singularity (§ 3.4).

The orientation control is now again strictly decreasing (Figure 19, left) since the residual angle is lower than $\pi/2$. On the opposite, the translation control is disturbed by the orientation control (Figure 19, right) but still converges. Both orientation errors and errors in the image are strictly decreasing (Figure 20, left). As for the image trajectory of the trihedron center, it is not as nice as in the previous cases (Figure 20, right) and depends on the relative choice of the control gains, μ_u and μ_h. Indeed, this trajectory was obtained with $\mu_u = 0.008$ and $\mu_h = 0.01$ while the choice $\mu_u = \mu_h = 0.01$ did not keep the trihedron visible.

7 Conclusion and Perspectives

In this paper, we achieved a theoretical and geometrical study of several line representations in order to determine the one which is best suited for visual servoing. This led us to define a new representation, the so-called binormalized Plücker coordinates, which has two advantages from the control point of view. The first one is that it allows us to define the concept of image line alignment, which gathers both 2D and 3D information. The second one is that we are able to define a control law which realizes such an image line alignment with some very interesting properties. First, the proposed law is an analytic inversion of the motion equations and does not need any numerical inversion of a Jacobian. Second, it shows a partial decoupling between translation and rotation. Third, global asymptotic stability conditions could be found, with a geometrical interpretation in the particular case of the orthogonal trihedron. This control law was validated both with simulations and experimentally.

However, there still remain several issues to be developed. In a first time, stability conditions should be extended to the case where orientation and translation controls are activated simultaneously. This may be eased by recent results in the field of cascaded systems[PL98]. In parallel, one could think of studying the robustness of the proposed control law to measurement and/or calibration errors. Another point to be addressed consists of the geometrical determination of the stationary points in the general case, but this will certainly be tightly coupled to the 3D line configuration. One could also investigate the pose estimation algorithms or the Euclidean reconstruction algorithms based on lines to derive from them a simplified method which could only estimate the line directions without performing an unneeded full pose computation. Another research direction is visual servoing from projective lines, following the recent work by Ruf and Horaud[RH99] on projective visual servoing. As for the practical side, we still need to reduce drastically the cycle time of the control loop.

Acknowledgments

This work was supported by the European Community through the Esprit-IV reactive LTR project number 26247 (VIGOR). During this work, Nicolas Andreff was staying at INRIA Rhône-Alpes as a Ph.D. student.

A Proofs

A.1 Theorem 1

Firstly, let us state the following lemma:

Lemma 1 *For all configurations of rigidly linked lines:*

$$\sum_{i=1}^{n} \underline{\mathbf{u}}_i \times \underline{\mathbf{u}}_i^* = 0 \Rightarrow \underline{\mathbf{u}}_i = \pm \underline{\mathbf{u}}_i^*, i = 1..n \tag{39}$$

Proof : Recall the Rodrigues formula, which gives the image $\mathbf{v}'$ of a vector $\mathbf{v}$ by a rotation of axis $\underline{\mathbf{n}}$ and angle θ:

$$\mathbf{v}' = \mathbf{v} + \sin\theta(\underline{\mathbf{n}} \times \mathbf{v}) + (1 - \cos\theta)\underline{\mathbf{n}} \times (\underline{\mathbf{n}} \times \mathbf{v}) \tag{40}$$

Since we consider a set of rigidly linked lines, the $\underline{\mathbf{u}}_i$'s differ from the $\underline{\mathbf{u}}_i^*$'s by a unique rotation. Hence, we can apply (40) for all $i = 1..n$:

$$\underline{\mathbf{u}}_i = \underline{\mathbf{u}}_i^* + \sin\theta(\underline{\mathbf{n}} \times \underline{\mathbf{u}}_i^*) + (1 - \cos\theta)\underline{\mathbf{n}} \times (\underline{\mathbf{n}} \times \underline{\mathbf{u}}_i^*), \quad i = 1..n \tag{41}$$

Building for all $i = 1..n$, the crossproduct $\underline{\mathbf{u}}_i \times \underline{\mathbf{u}}_i^*$ and expressing $\underline{\mathbf{u}}_i^*$ on the orthogonal basis $(\underline{\mathbf{n}}, \underline{\mathbf{n}} \times \underline{\mathbf{u}}_i^*, \underline{\mathbf{n}} \times (\underline{\mathbf{n}} \times \underline{\mathbf{u}}_i^*))$ yields:

$$\begin{aligned}
\underline{\mathbf{u}}_i \times \underline{\mathbf{u}}_i^* &= \sin\theta\Big[(\underline{\mathbf{n}}^T \underline{\mathbf{u}}_i^*)^2 - 1\Big]\underline{\mathbf{n}} \\
&+ \Big[(1 - \cos\theta)\underline{\mathbf{n}}^T \underline{\mathbf{u}}_i^* + \alpha_i \underline{\mathbf{n}}^T \underline{\mathbf{u}}_i^*\Big] \underbrace{(\underline{\mathbf{n}} \times \underline{\mathbf{u}}_i^*)}_{\perp \underline{\mathbf{n}}} \\
&+ \Big[\beta_i \underline{\mathbf{n}}^T \underline{\mathbf{u}}_i^*\Big] \underbrace{\underline{\mathbf{n}} \times (\underline{\mathbf{n}} \times \underline{\mathbf{u}}_i^*)}_{\perp \underline{\mathbf{n}}}
\end{aligned}$$

Consequently, if the sum of these crossproducts is zero, then

$$\sum_{i=1}^{n} \sin\theta\Big[(\underline{\mathbf{n}}^T \underline{\mathbf{u}}_i^*)^2 - 1\Big] = 0 \tag{42}$$

From this equation, we deduce immediately that either $\sin\theta = 0$ or $\underline{\mathbf{u}}_i^* = \pm\underline{\mathbf{n}}, \forall i = 1..n$. Discarding the second case which can be shown to be a degenerate case of the first one, we obtain the final result as an interpretation of the first case. □

Now, let us prove Theorem 1.

Consider the Lyapunov function $L = \frac{1}{2}\sum_{i=1}^{n} \|\underline{\mathbf{u}}_i - \underline{\mathbf{u}}_i^*\|^2$. Its derivative is:

$$\dot{L}_u = -\sum_{j=1}^{n} \dot{\underline{\mathbf{u}}}_j^T \underline{\mathbf{u}}_j^* \tag{43}$$

From the dynamical equation (24) and the control definition (26), we can write (43) as:

$$\dot{L}_u = -\mu_u \sum_{j=1}^{n} \left(\left(\sum_{i=1}^{n} \underline{\mathbf{u}}_i \times \underline{\mathbf{u}}_i^* \right) \times \underline{\mathbf{u}}_j \right)^T \underline{\mathbf{u}}_j^* \tag{44}$$

Using the following property of the mixt product:

$$(\mathbf{u} \times \mathbf{v})^T \mathbf{w} = (\mathbf{v} \times \mathbf{w})^T \mathbf{u} \tag{45}$$

we obtain:

$$\dot{L}_u = -\mu_u \left(\sum_{j=1}^{n} \underline{\mathbf{u}}_j \times \underline{\mathbf{u}}_j^* \right)^T \left(\sum_{i=1}^{n} \underline{\mathbf{u}}_i \times \underline{\mathbf{u}}_i^* \right) \tag{46}$$

Consequently, $\dot{L}$ is strictly negative unless $\sum_{i=1}^{n} \underline{\mathbf{u}}_i \times \underline{\mathbf{u}}_i^* = 0$. From Lemma 1, this means that $\dot{L}$ is strictly negative unless $\underline{\mathbf{u}}_i \neq \pm \underline{\mathbf{u}}_i^*$. We find here again the same unstable initial stationnary point as in the single line case. Thus, the control (26) is asymptotically stable provided that $\underline{\mathbf{u}}_i(t=0) \neq -\underline{\mathbf{u}}_i^*$, $\forall i = 1..n$. □

A.2 Theorem 2

Consider the Lyapunov function defined by:

$$L_h = \frac{1}{2}\sum_{j=1}^{n} \|\underline{\mathbf{h}}_j - \underline{\mathbf{h}}_j^*\|^2 \tag{47}$$

the derivative of which is:

$$\dot{L}_h = -\sum_{j=1}^{n} \dot{\underline{\mathbf{h}}}_j^T \underline{\mathbf{h}}_j^* \tag{48}$$

From (25), where $\Omega = 0$, we have:

$$\dot{L}_h = -\sum_{j=1}^{n} \left(-\frac{V^T \underline{\mathbf{h}}_j}{h_j} (\underline{\mathbf{u}}_j \times \underline{\mathbf{h}}_j) \right)^T \underline{\mathbf{h}}_j^* \tag{49}$$

which rewrites:

$$\dot{L}_h = \sum_{j=1}^{n} \frac{V^T \underline{\mathbf{h}}_j}{h_j} \epsilon_j \tag{50}$$

Inserting the control (28) yields:

$$\dot{L}_h = -\mu_h \sum_{j=1}^{n} \frac{\left(\sum_{i=1}^{n} \mathbf{B}_i \epsilon_i \underline{\mathbf{h}}_i\right)^T \underline{\mathbf{h}}_j}{h_j} \epsilon_j \tag{51}$$

This expression also rewrites under the following form which gives the analytic condition for stationnary points to exist (i.e. when $\dot{L}_h = 0$):

$$\dot{L}_h = -\mu_h \left(\sum_{i=1}^{n} \mathbf{B}_i \epsilon_i \underline{\mathbf{h}}_i\right)^T \left(\sum_{i=1}^{n} \frac{\epsilon_i \underline{\mathbf{h}}_i}{h_i}\right) \tag{52}$$

□

A.3 Theorem 3

Consider the same Lyapunov function L as in the previous proof. Then its derivative is given by (50) independently from the translation control (provided that rotation control has converged). Inserting the control (32) in (50) gives

$$\dot{L}_h = -\mu_h \sum_{j=1}^{3} \frac{\left(\sum_{i=1}^{3} \epsilon_i \underline{\mathbf{h}}_i\right)^T \underline{\mathbf{h}}_j}{h_j} \epsilon_j \tag{53}$$

Noting $\mathbf{P}_j = \mathbf{I}_3 - \underline{\mathbf{h}}_j \underline{\mathbf{h}}_j^T$, the projection operator orthogonal to $\underline{\mathbf{h}}_j$, and $\mathbf{e}_j = \underline{\mathbf{h}}_j - \underline{\mathbf{h}}_j^*$, we remark that:

$$\epsilon_j \underline{\mathbf{h}}_j = \underline{\mathbf{u}}_j \times (\mathbf{P}_j \mathbf{e}_j), \ \forall j \tag{54}$$

Then, using this equation, we can rewrite the expression of $\dot{L}$ as

$$\dot{L}_h = -\mu_h \sum_{j=1}^{3} \frac{1}{h_j} \left(\sum_{i=1}^{3} \underline{\mathbf{u}}_i \times (\mathbf{P}_i \mathbf{e}_i)\right)^T \left(\underline{\mathbf{u}}_j \times (\mathbf{P}_j \mathbf{e}_j)\right) \tag{55}$$

$$= -\mu_h \sum_{j=1}^{3} \frac{1}{h_j} \sum_{i=1}^{3} \left(\underline{\mathbf{u}}_i \times (\mathbf{P}_i \mathbf{e}_i)\right)^T \left(\underline{\mathbf{u}}_j \times (\mathbf{P}_j \mathbf{e}_j)\right) \tag{56}$$

$$= -\mu_h \sum_{j=1}^{3} \frac{1}{h_j} \sum_{i=1}^{3} \left[(\underline{\mathbf{u}}_i \times (\mathbf{P}_i \mathbf{e}_i)) \times \underline{\mathbf{u}}_j\right]^T (\mathbf{P}_j \mathbf{e}_j) \tag{57}$$

$$= -\mu_h \sum_{j=1}^{3} \frac{1}{h_j} \sum_{i=1}^{3} \left[(\underline{\mathbf{u}}_i^T \underline{\mathbf{u}}_j)(\mathbf{P}_i \mathbf{e}_i) - \underline{\mathbf{u}}_j^T (\mathbf{P}_i \mathbf{e}_i) \underline{\mathbf{u}}_i\right]^T (\mathbf{P}_j \mathbf{e}_j) \tag{58}$$

which finally gives:

$$\dot{L}_h = -\mu_h \sum_{j=1}^{3} \frac{1}{h_j} \sum_{i=1}^{3} \left[(\underline{\mathbf{u}}_i^T \underline{\mathbf{u}}_j - \underline{\mathbf{u}}_i \underline{\mathbf{u}}_j^T)(\mathbf{P}_i \mathbf{e}_i) \right]^T (\mathbf{P}_j \mathbf{e}_j) \tag{59}$$

Define now

$$E = \begin{pmatrix} \mathbf{P}_1 \mathbf{e}_1 \\ \mathbf{P}_2 \mathbf{e}_2 \\ \mathbf{P}_3 \mathbf{e}_3 \end{pmatrix}, \quad U = \begin{pmatrix} \underline{\mathbf{u}}_1 \\ \underline{\mathbf{u}}_2 \\ \underline{\mathbf{u}}_3 \end{pmatrix}$$

and the operator:

$$\mathbf{P}_U = \frac{1}{2}(\mathbf{I}_9 - U \otimes U^T) = \begin{pmatrix} \mathbf{I}_3 - \underline{\mathbf{u}}_1 \underline{\mathbf{u}}_1^T & -\underline{\mathbf{u}}_2 \underline{\mathbf{u}}_1^T & -\underline{\mathbf{u}}_3 \underline{\mathbf{u}}_1^T \\ -\underline{\mathbf{u}}_1 \underline{\mathbf{u}}_2^T & \mathbf{I}_3 - \underline{\mathbf{u}}_2 \underline{\mathbf{u}}_2^T & -\underline{\mathbf{u}}_3 \underline{\mathbf{u}}_2^T \\ -\underline{\mathbf{u}}_1 \underline{\mathbf{u}}_3^T & -\underline{\mathbf{u}}_2 \underline{\mathbf{u}}_3^T & \mathbf{I}_3 - \underline{\mathbf{u}}_3 \underline{\mathbf{u}}_3^T \end{pmatrix}$$

One can easily prove that $\mathbf{P}_U$ is an orthogonal projection onto the kernel of U^T, hence on the hyperplane Π orthogonal to U.

After convergence of the rotational control, we have

$$\underline{\mathbf{u}}_j = \underline{\mathbf{u}}_j^*, \quad j = 1..3$$

from which we deduce that

$$\underline{\mathbf{u}}_j^T (\mathbf{P}_j \mathbf{e}_j) = 0, \quad j = 1..3 \tag{60}$$

Using this result and the trihedron orthogonality, we obtain

$$\dot{L} = -2\mu_h \tilde{E}^T \mathbf{P}_U E \tag{61}$$

where $\tilde{E}$ is the vector obtained by concatenating the $\tilde{\mathbf{e}}_j = \frac{1}{h_j} \mathbf{P}_j \mathbf{e}_j$ for all $j = 1..3$.

From (60), we find that E and U are orthogonal, hence $E \in \Pi$ and $\mathbf{P}_U E = E$. This simplifies the expression of $\dot{L}$ into:

$$\dot{L} = -2\mu_h \tilde{E}^T E = -2\mu_h \sum_{j=1}^{3} \frac{1}{h_j} \mathbf{e'}_j^T \mathbf{e}'_j$$

where $\mathbf{e}'_j = \mathbf{P}_j \mathbf{e}_j$.

Consequently, this sum is negative and only vanishes when all $\mathbf{e}'_j$ are zero. After the convergence of the rotation control, this is only possible when $\underline{\mathbf{h}}_j = \pm \underline{\mathbf{h}}_j^*$.

□

References

[AE02] N. Andreff and B. Espiau. Revisiting plücker coordinates in vision-based control. In *8th International Symposium on Advances in Robot Kinematics (ARK2002)*, Caldes de Malavella, Spain, June 24–28 2002.

[AEH01] N. Andreff, B. Espiau, and R. Horaud. Visual servoing from lines. Technical Report RR–4226, INRIA, 2001.

[AHE01] N. Andreff, R. Horaud, and B. Espiau. Robot hand-eye calibration using structure from motion. *International Journal of Robotics Research*, 20(3):228–248, March 2001.

[And99] N. Andreff. *Asservissement visuel à partir de droites et auto-étalonnage pince-caméra*. Thèse de doctorat, Institut National Polytechnique de Grenoble, Grenoble, November 1999.

[AZ95] M. Armstrong and A. Zisserman. Robust object tracking. In *Proc. Asian Conference on Computer Vision*, volume I, pages 58–61, 1995.

[BR79] O. Bottema and B. Roth. *Theoretical Kinematics*. Dover Publications, 1979.

[CEG+96] B. Chazelle, H. Edelsbrunner, L. J. Guibas, M. Sharir, and J. Stolfi. Lines in space : Combinatorics and algorithms. *Algorithmica*, 15:428–447, 1996.

[CH99] S. Christy and R. Horaud. Iterative pose computation from line correspondences. *Computer Vision and Image Understanding*, 73(1):137–144, January 1999.

[Cha90] F. Chaumette. *La relation vision-commande : théorie et application à des tâches robotques* . Thèse, Université de Rennes I, 1990.

[Che91] H. Chen. Pose determination from line-to-plane correspondences: Existence solutions and closed-form solutions. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 13(6):530–541, June 1991.

[Cor93] P. I. Corke. Visual control of robot manipulators — a review. In K. Hashimoto, editor, *Visual Servoing — Real-Time Control*

of Robot Manipulators Based on Visual Sensory Feedback, pages 33–70. World Scientific, 1993.

[DC00] T. Drummond and R. Cipolla. Real-time tracking of multiple articulated structures in multiple views. In *6th European Conference on Computer Vision*, June 2000.

[DRLR89] M. Dhome, M. Richetin, J.T. Lapresté, and G. Rives. Determination of the attitude of 3D objects from a single perspective view. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1265–1278, December 1989.

[ECR92] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. IEEE *Transactions on Robotics and Automation*, 8(3):313–326, June 1992.

[Fau93] O. Faugeras. *Three-Dimensional Computer Vision - A Geometric Viewpoint*. Artificial intelligence. The MIT Press, Cambridge, MA, USA, Cambridge, MA, 1993.

[GO97] J. Goodman and J. O'Rourke, editors. *Handbook of Discrete and Computational Geometry*. CRC Press, 1997.

[Hag97] G. D. Hager. A modular system for robust positioning using feedback from stereo vision. IEEE *Transactions on Robotics and Automation*, 13(4):582–595, August 1997.

[Har94] R.I. Hartley. Projective reconstruction from line correspondences. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Maui, Hawaii, USA*, pages 903–907, Seattle, USA, 1994.

[HCLL89] R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle. An analytic solution for the perspective 4-point problem. *Computer Vision, Graphics and Image Processing*, 47:33–44, 1989.

[HHC96] S. Hutchinson, G. D. Hager, and P. I. Corke. A Tutorial on Visual Servo Control. IEEE *Transactions on Robotics and Automation*, 12(5):651–670, October 1996.

[KMM+96] D. Khadraoui, G. Motyl, P. Martinet, J. Gallice, and F. Chaumette. Visual servoing in robotics scheme using a camera/laser-stripe sensor. *IEEE Transactions on Robotics and Automation*, 12(5):743–750, October 1996.

[KRD+98] D. Khadraoui, R. Rouveure, C. Debain, P. Martinet, P. Bonton, and J. Gallice. Vision based control in driving assistance of agricultural vehicles. *International Journal of Robotics Research*, 17(10):1040–1054, October 1998.

[LD98] M. Lützeler and E.D. Dickmanns. Road Recognition with MarVEye. In *Int. Conf. on Intelligent Vehicles*, 1998.

[LH86] Y. Liu and T.S. Huang. Estimation of rigid bidy motion using straight line correspondences, further results. In *Proc. 8th Int. Conf. on Pattern Recognition*, pages 306–307, Paris, France, October 1986.

[LHF90] Y. Liu, T.S. Huang, and O.D. Faugeras. Determination of camera location from 2D to 3D lines and points. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 12(1):28–37, January 1990.

[Low91] D. Lowe. Fitting parameterized three-dimensional models to images. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

[Mac90] J. M. MacCarthy. *Introduction to Theoretical Kinematics*. MIT Press, 1990.

[MBCM99] E. Marchand, P. Bouthémy, F. Chaumette, and V. Moreau. Robust real-time visual tracking using a 2D-3D model-based approach. In *International Conference on Computer Vision*, volume 1, pages 262–268, Kerkira, Greece, September 1999.

[MH01] R. Mahony and T. Hamel. Visual servoing using linear features for under-actuated rigid-body dynamics. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hawaii, USA, 2001.

[MH02] F. Martin and R. Horaud. Multiple camera tracking of rigid objects. *International Journal of Robotics Research*, 21(2), February 2002.

[MLS94] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[Nav93] N. Navab. *Visual motion of lines and cooperation between motion and stereo*. Thèse de doctorat, Université Paris-Sud, 1993.

[PL98] E. Panteley and A. Loria. On global uniform asymptotic stability of nonlinear time-varying systems in cascade. *Systems and Control Letters*, 33(2):131, 1998.

[Plü65] J. Plücker. On a new geometry of space. *Philosophical Transactions of the Royal Society of London*, 155:725–791, 1865.

[PPR98] H. Pottmann, M. Peternell, and B. Ravani. Approximation in line space – applications in robot kinematics and surface reconstruction. In J. Lenarčič and M. L. Husty, editors, *Advances in Robot Kinematics: Analysis and Control*, pages 403–412. Kluwer Academic Publishers, 1998.

[QK97] L. Quan and T. Kanade. Affine structure from line correspondences with uncalibrated affine cameras. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 19(8):834–845, August 1997.

[RE87] Patrick Rives and Bernard Espiau. Estimation recursive de primitives 3D au moyen d'une camera mobile. *Traitement du Signal*, 4:259–272, 1987.

[RH99] A. Ruf and R. Horaud. Visual Servoing of Robot Manipulators Part I: Projective Kinematics. *International Journal of Robotics Research*, 18(11):1101–1118, November 1999.

[SA90] M. Spetsakis and J. Aloimonos. Structure from motion using line correspondences. *International Journal on Computer Vision*, 4:171–183, 1990.

[SK52] J.G. Semple and G.T. Kneebone. *Algebraic Projective Geometry*. Oxford Science Publication, 1952.

[SLBE91] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control : The Task Function Approach* . Clarendon Press, Oxford, 1991.

[TN00] M. Tonko and H.-H. Nagel. Model-based stereo-tracking of non-polyhedral objects for automatic disassembly experiments. *International Journal on Computer Vision*, 37(1):99–118, 2000.

[VIG01] VIGOR. Visually Guided Robots Using Uncalibrated Cameras, ESPRIT–IV reactive LTR project number 26247, 1998–2001. `http://www.inrialpes.fr/VIGOR`.

[VLF95] T. Viéville, Q.T. Luong, and O.D. Faugeras. Motion of points and lines in the uncalibrated case. *International Journal on Computer Vision*, 17(1), 1995.