Visual Servoing from Lines

Nicolas Andreff, Bernard Espiau, Radu Horaud

Proceedings of the 2000 IEEE
International Conference on Robotics & Automation
San Francisco, CA • April 2000

Visual Servoing from Lines

Nicolas Andreff, Bernard Espiau, Radu Horaud *
INRIA Rhône-Alpes/ GRAVIR-IMAG, 655, av. de l'Europe, 38330 Montbonnot, France
email: firstname.lastname@inrialpes.fr

Abstract

In this paper, we use an alternative formulation of the Euclidean Plücker coordinates to define the new of normalized Plücker coordinates alignment of lines. This concept is more relevant than usual image alignment to position a single calibrated camera with respect to a set of known 3D *lines.*

An explicit control law is derived which realizes such a task both for one or several lines. It is tested by positioning the camera with respect to an orthogonal trihedron. This application is all the challenging in that it requires depth to be observed. We do this by using the image projection of a laser pointer and adequately completing the control law.

1 Introduction

Our work is related to the control of one camera mounted onto a robot gripper. The camera is calibrated: its intrinsic parameters and its relative position with respect to the robot end-effector (also referred to as hand-eye transformation [17]) are fixed and known in advance.

The problem we address is to move the camera from a current position where it observes a static scene represented by a set of lines to a target position specified by a desired image projection of the scene (Figure 1).

To solve this problem, we will use the visual servoing approach [5]. Visual servoing is a sensor-based closed-loop control [16] with one or several cameras as sensors: the measurement of an error, computed from images, determines the next instantaneous motion to be taken by the camera (or the robot, since they are rigidly linked in our case) in order to drive the error to zero.

Most of the efforts in visual servoing are devoted to points (see [9] for an exhaustive reference list) but interestingly, only few works were devoted to visual servoing from lines. As far as we know there exists only one class of approach using monocular vision [5, 18] and another one using stereovision [7].

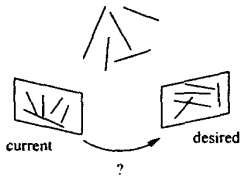


Figure 1: The problem is to move the camera such that it attains a position where the current image coincides with the desired one.

One reason to this lack of interest is the difficulty to find a good representation for lines while points are rather easy to handle. In particular, the monocular approach [5] used a parameterization of image lines by their angle and distance to origin which did not fit its representation of 3D lines as the kernel of two plane equations of the form $AX + BY + CZ + D = 0$. Moreover, this approach is based on image (or 2D) line alignment while, in practice, 3D line alignment is looked for.

The main contributions of this paper are: the use of a representation of 3D lines that is well suited to express the relation between the instantaneous motion of the camera and the apparent motion observed in the image (Section 2); the definition of a new line positioning task which we claim to be more relevant than the usual 2D alignment of lines (Section 3); the derivation of an explicit control law which realizes this task for any number of lines (Section 4); and, finally, the application of this control law to an industrial problem (Section 5), validated by experimental results (Section 6).

2 Line representation

2.1 3D line

The representation of lines we propose to use is based on the Euclidean Plücker coordinates of a 3D line [2]. They are a Euclidean variant of the projective Plücker coordinates [14, 6] and can be defined from the fact that a 3D point and a 3D orientation

*The authors acknowledge support from the European Community through the Esprit-IV reactive LTR project number 26247 (VIGOR).

- 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 ($\boldsymbol{P}$) while points in the image are written with lower-case bold characters ($\boldsymbol{p}$).
- 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 $\boldsymbol{\Omega}$ and linear velocity $\mathbf{V}$.
- 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: Notations used in this paper

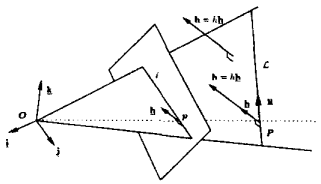


Figure 2: Euclidean Plücker coordinates of a 3D line

define a unique 3D line.

Let $\boldsymbol{P}$ be a 3D point and $\underline{\mathbf{u}}$ a unit vector expressed in the camera frame $(\boldsymbol{O},\mathbf{i},\mathbf{j},\mathbf{k})$ and $\mathcal{L}$ be the 3D line they define.

Define $\mathbf{h}=\overrightarrow{\boldsymbol{OP}}\times\underline{\mathbf{u}}$ and remark that this vector is independent of the point we choose on the line. Thus, the Euclidean Plücker coordinates are defined as $\mathcal{L}=(\underline{\mathbf{u}},\mathbf{h})$ with $\|\underline{\mathbf{u}}\|=1$ and $\mathbf{h}^T\underline{\mathbf{u}}=0$ (Figure 2).

Notice that $\mathbf{h}$ is orthogonal to $\mathcal{L}$ and to the plane defined by the camera center and $\mathcal{L}$ (hence, $\mathbf{h}^T\boldsymbol{P}=\mathbf{h}^T\underline{\mathbf{u}}=0$). This plane is often called the interpretation plane.

The case where $\underline{\mathbf{h}}=0$ happens when the 3D line contains the camera center and hence projects as a point in the image. In the following, we assume that it never occurs.

Then, the usual interpretation of the Euclidean Plücker coordinates is that $\mathcal{L}$ is the 3D line oriented by $\underline{\mathbf{u}}$ passing through the point $\mathbf{h}\times\underline{\mathbf{u}}$. We prefer an alternative sequential interpretation which isolates the depth: $\mathcal{L}$ is the line lying in the plane orthogonal to $\underline{\mathbf{h}}=\mathbf{h}/\|\mathbf{h}\|$ and parallel to $\underline{\mathbf{u}}$, and at a depth $h=\|\mathbf{h}\|$ from the origin.

Thus, we represent a 3D line $\mathcal{L}$ by:

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

Note that Hager *et al* [7] represent $\mathcal{L}$ as the tuple $(\boldsymbol{P},\underline{\mathbf{u}})$ but, in fact, implicitly use the representation above.

We call the couple $(\underline{\mathbf{h}}^T,\underline{\mathbf{u}})$ the *normalized Plücker coordinates* of the line. Remark that they represent the pencil of all the lines oriented by $\underline{\mathbf{u}}$ and lying in the interpretation plane $\underline{\mathbf{h}}$. It is a one-dimensional (depth) manifold.

2.2 2D line

The reason for the representation (1) is that the 2D line ℓ, projection of a 3D line $\mathcal{L}=(\underline{\mathbf{h}}^T,\underline{\mathbf{u}}^T,h)^T$ onto the image plane is completely defined by $\underline{\mathbf{h}}$. Indeed, consider a point $\boldsymbol{P}=(xZ,yZ,Z)^T$ of $\mathcal{L}$ projecting in the point $\boldsymbol{p}=(x,y,1)^T$ of ℓ (Figure 2). The equation of ℓ is of the form $ax+by+c=0$ and we take the convention that $a^2+b^2+c^2=1$. Rewriting now $\mathbf{h}^T\boldsymbol{P}=0$ as $(h\underline{\mathbf{h}})^T(Z\boldsymbol{p})=0$ and noticing that neither h nor Z can be equal to 0, yields $\underline{\mathbf{h}}^T\boldsymbol{p}=0$ which is precisely the equation of ℓ. Hence, $\underline{\mathbf{h}}$ is the vector formed with the coefficients of the equation of ℓ.

Finally, we recall that ℓ is the intersection of the interpretation plane and the image plane and that, reciprocally, ℓ defines the interpretation plane.

2.3 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 the instantaneous angular velocity Ω and the instantaneous linear velocity V of a given point expressed in the camera frame. From them, one defines the velocity screw $\tau=(V,\Omega)$.

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* [15] recall the derivative $(\dot{\underline{\mathbf{u}}},\mathbf{h})$ of the Euclidean Plücker coordinates $(\underline{\mathbf{u}},\mathbf{h})$ of a 3D line (2)-(3) and Navab [12] derives from these equations the *line motion field equation*. It can be interpreted as the relation between the camera motion and the apparent motion of a 3D line in the image. It is therefore the derivative $\dot{\underline{\mathbf{h}}}$ of the image projection $\underline{\mathbf{h}}$ of the 3D line, which is given in (4). As for the derivative of the depth $\dot{h}$ (5), it is obtained by differentiating the expression defining h. Finally:

$$\dot{\underline{\mathbf{u}}}=\Omega\times\underline{\mathbf{u}} \quad (2)$$

$$\dot{\mathbf{h}}=\Omega\times\mathbf{h}+V\times\underline{\mathbf{u}} \quad (3)$$

$$\dot{\underline{\mathbf{h}}}=\Omega\times\underline{\mathbf{h}}-\frac{V^T\underline{\mathbf{h}}}{h}(\underline{\mathbf{u}}\times\underline{\mathbf{h}}) \quad (4)$$

$$\dot{h}=\dot{\sqrt{\mathbf{h}^T\mathbf{h}}}=V^T(\underline{\mathbf{u}}\times\underline{\mathbf{h}}) \quad (5)$$

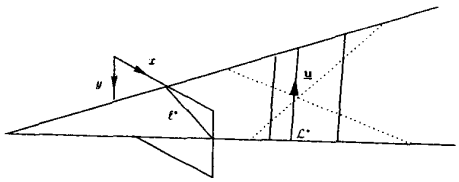


Figure 3: The solid lines are in normalized Plücker coordinates alignment with $\mathcal{L}^*$ while the dotted lines are not, even though they project onto ℓ^*.

3 Normalized Plücker coordinates alignment

We define now a normalized Plücker coordinates alignment task inspired by two ideas. The first one concerns the projection $\boldsymbol{p}^*$ in the image of a 3D point $\boldsymbol{P}^*$: The alignment of a point $\boldsymbol{p}$ with $\boldsymbol{p}^*$ in the image is the alignment of their two lines of sight. It corresponds to bringing the physical 3D point $\boldsymbol{P}$, whose projection is $\boldsymbol{p}$, into the manifold of all the points equivalent to $\boldsymbol{P}^*$ up to depth. The second idea comes from [11]. This work defines a 2D 1/2 visual servoing control law for a set of points, where orientation is controlled by driving 3D errors to zero and simultaneously position is controlled by driving errors in the image to zero.

From the first idea we argue that the image alignment of 3D lines should not consist in the alignment of their interpretation planes as it is done in [5, 18] but in bringing them into a manifold of equivalent 3D lines up to depth.

It is trivial to see that the set of the lines equivalent to a 3D line $\mathcal{L} = (\underline{\mathbf{h}}, \underline{\mathbf{u}}, h)$ up to depth is defined by $\{(\underline{\mathbf{h}}, \underline{\mathbf{u}}, Z), Z \in \mathbb{R}\}$. It is geometrically interpreted as the pencil of all the parallel lines, oriented by $\underline{\mathbf{u}}$ and lying in the interpretation plane orthogonal to $\underline{\mathbf{h}}$. It is thus represented by normalized Plücker coordinates.

This implies, as in [10], a mixed 2D-3D information. We thus define the new alignment of lines as the alignment of the pencil based on the current 3D line position with respect to the camera onto the pencil based on its desired position (see Figure 3). With our notations, it writes:

$$\underline{\mathbf{u}} = \underline{\mathbf{u}}^* \quad \underline{\mathbf{h}} = \underline{\mathbf{h}}^* \tag{6}$$

With this proper definition of line alignment, we can now define a control scheme with nice properties.

4 Control

Before deriving our control, let us recall some elements of the control law in the previous monocular visual servoing from lines approach [5]. A 3D line was represented as the intersection of two planes:

$$\begin{cases} a_1 x + b_1 y + c_1 z + d_1 = 0 \\ a_2 x + b_2 y + c_2 z + d_2 = 0 \end{cases} \tag{7}$$

and a 2D line by an angle θ and a distance to the origin ρ: $\cos\theta x + \sin\theta y - \rho = 0$. This yielded the following motion equations for a 2D line:

$$\begin{pmatrix} \dot{\theta} \\ \dot{\rho} \end{pmatrix} = \begin{pmatrix} \lambda_\theta c\theta & \lambda_\theta s\theta & -\lambda_\theta \rho & -\rho c\theta & -\rho s\theta & -1 \\ \lambda_\rho c\theta & \lambda_\rho s\theta & -\lambda_\rho \rho & (1+\rho^2)s\theta & -(1+\rho^2)c\theta & 0 \end{pmatrix} \begin{pmatrix} \mathbf{V} \\ \mathbf{\Omega} \end{pmatrix} \tag{8}$$

where λ_θ and λ_ρ depend on the plane equations coefficients as well as on θ. By numerically inverting these equations, provided enough lines are available, one obtains the camera velocity screw which realizes a desired motion $(\dot{\theta}, \dot{\rho})$ of the 2D line. Then, by choosing that this motion should be aligned at each iteration of the control-loop with the error $\begin{pmatrix} \theta - \theta^* \\ \rho - \rho^* \end{pmatrix}$ to be servoed to zero, exponential convergence was proven. However, one could not distinguish what contributed to control the orientation of the camera from what helped in controlling its position.

Moreover, as these motion equations are complicated, one is reluctant to compute all the coefficients in the matrices above at each iteration. Therefore, taking advantage of continuity properties of the control, an estimate of each matrix was computed in the desired position of the camera and the control was restricted to a neighborhood of this position.

On the opposite, our representation of lines yields much simpler motion equations.

4.1 Case of a single line

The motion equations (2) and (4) show a partial decoupling. Indeed, translations of the camera do not modify the 3D line orientation. Therefore, we can take advantage of this property to provide a control law which makes the rotation control Ω and the translation control V explicit, is globally convergent and does not require any depth estimation.

To benefit from the decoupling, and assuming that $\underline{\mathbf{u}}$ can be recovered, we define the following control scheme which works in two phases. First, align the orientation $\underline{\mathbf{u}}$ of the line with the desired one, $\underline{\mathbf{u}}^*$, using the control defined by:

$$\Omega = \mu_\omega \, \underline{\mathbf{u}} \times \underline{\mathbf{u}}^* \quad \mu_\omega > 0 \tag{9}$$

$$V = 0 \tag{10}$$

Then, after convergence in orientation align the interpretation plane, i.e. control $\underline{\mathbf{h}}$ to its desired value $\underline{\mathbf{h}}^*$ with:

$$\Omega = \mu_\omega \, \underline{\mathbf{u}} \times \underline{\mathbf{u}}^* \qquad \mu_\omega > 0 \tag{11}$$

$$V = -\mu_v \, \underline{\mathbf{u}} \times (\underline{\mathbf{h}} - \underline{\mathbf{h}}^*) \qquad \mu_v > 0 \tag{12}$$

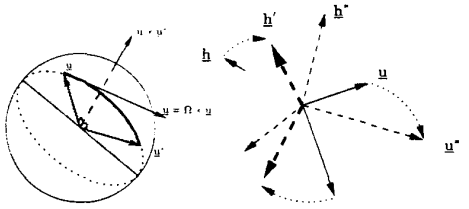


Figure 4: The rotational control brings $\underline{\mathbf{u}}$ over $\underline{\mathbf{u}}^*$ and $\underline{\mathbf{h}}$ onto $\underline{\mathbf{h}}'$, somewhere in the plane orthogonal to $\underline{\mathbf{u}}^*$.

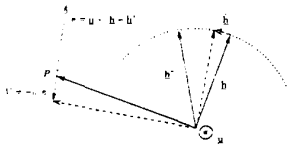


Figure 5: Controlling the interpretation plane normal vector when $\underline{\mathbf{u}} = \underline{\mathbf{u}}^*$.

Ideally, Ω should always be 0 in the second stage. We do not make this simplification to cope with robot calibration errors. Then,

Theorem 1 *The first stage (rotation control only) is asymptotically stable provided that in the initial position* $\underline{\mathbf{u}}(t=0) \neq -\underline{\mathbf{u}}^*$. *Then, the second stage (ideally, translation control only) is also asymptotically stable provided that* $\underline{\mathbf{u}} = \underline{\mathbf{u}}^*$ *and* $\underline{\mathbf{h}}(t=0) \neq -\underline{\mathbf{h}}^*$.

Details of the proof, based on the study of the Lyapunov functions $L_u = \|\underline{\mathbf{u}} - \underline{\mathbf{u}}^*\|^2$ and $L_h = \|\underline{\mathbf{h}} - \underline{\mathbf{h}}^*\|^2$ are available in [1], but we will give here only a geometrical insight of it.

The first control law generates a rotation in the plane $(O, \underline{\mathbf{u}}, \underline{\mathbf{u}}^*)$ until $\underline{\mathbf{u}} = \underline{\mathbf{u}}^*$. Its effect on the interpretation plane is to bring $\underline{\mathbf{h}}$ into the plane orthogonal to $\underline{\mathbf{u}}^*$ (Figure 4). Then, the translation control rotates $\underline{\mathbf{h}}$ around the now fixed orientation $\underline{\mathbf{u}} = \underline{\mathbf{u}}^*$ until $\underline{\mathbf{h}}$ lies in the plane defined by $\underline{\mathbf{u}}$ and $\underline{\mathbf{h}}^*$ (Figure 5). Indeed, reporting (12) into (4) yields

$$\dot{\underline{\mathbf{h}}} = \frac{\mu_h}{h}[(\underline{\mathbf{u}} \times \underline{\mathbf{h}})^T \underline{\mathbf{h}}^*](\underline{\mathbf{u}} \times \underline{\mathbf{h}}). \qquad (13)$$

Notice now that this control takes the shortest path towards the configuration which minimizes the angle between $\underline{\mathbf{h}}$ and $\underline{\mathbf{h}}^*$ even if $\underline{\mathbf{u}} \neq \underline{\mathbf{u}}^*$ (Figure 6). However, there is a singular case when, at initial time, $\underline{\mathbf{u}}$, $\underline{\mathbf{h}}$ and $\underline{\mathbf{h}}^*$ lie in the same plane with a non minimal angle; then, the control vanishes. However, this singular case is unstable (Figure 6).

Consequently the translation control is highly satisfying since, if $\underline{\mathbf{u}} = \underline{\mathbf{u}}^*$, then $\underline{\mathbf{h}}$ is parallel to $\underline{\mathbf{h}}^*$, with minimal angle; hence, they are equal.

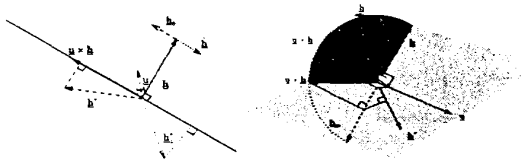


Figure 6: The translation control brings $\underline{\mathbf{h}}$ to $\underline{\mathbf{h}}^*$ along the shortest path (Left). Unstable singular case (Right): perturbating $\underline{\mathbf{h}}$ into $\underline{\mathbf{h}}'$ brings back $\underline{\mathbf{h}}$ to the configuration with minimal deviation from $\underline{\mathbf{h}}^*$.

4.2 Case of several lines

In the case of several lines, we make the following assumptions : the desired configuration of lines is reachable, a one-to-one correspondence can be established between the current and desired lines and the orientations $\underline{\mathbf{u}}_i$ can be recovered. Then, we simply sum up the influence of each line (represented by its subscript $i = 1..n$) into a compound control scheme: First, align the orientation $\underline{\mathbf{u}}_i$ of each line with the desired one $\underline{\mathbf{u}}_i^*$:

$$\Omega = \mu_\omega \sum_{i=1}^{n} \underline{\mathbf{u}}_i \times \underline{\mathbf{u}}_i^* \quad \mu_\omega > 0 \qquad (14)$$

$$V = 0 \qquad (15)$$

Then, after convergence in orientation align the interpretation planes, i.e. control each $\underline{\mathbf{h}}_i$ to its desired value $\underline{\mathbf{h}}_i^*$ by:

$$\Omega = \mu_\omega \sum_{i=1}^{n} \underline{\mathbf{u}}_i \times \underline{\mathbf{u}}_i^* \qquad \mu_\omega > 0 \qquad (16)$$

$$V = -\mu_v \sum_{i=1}^{n} \underline{\mathbf{u}}_i \times (\underline{\mathbf{h}}_i - \underline{\mathbf{h}}_i^*) \quad \mu_v > 0 \qquad (17)$$

Theorem 2 *The first stage (rotation control) is asymptotically stable provided that* $\forall i, \underline{\mathbf{u}}_i(t=0) \neq -\underline{\mathbf{u}}_i^*$. *Then, the second stage (ideally, translation control) reduces non-strictly the errors* $\|\underline{\mathbf{h}}_i - \underline{\mathbf{h}}_i^*\|$ *provided that* $\forall i, \underline{\mathbf{h}}_i(t=0) \neq -\underline{\mathbf{h}}_i^*$ *and* $\forall i, \underline{\mathbf{u}}_i = \underline{\mathbf{u}}_i^*$.

This theorem is a direct application of the results in [1] and extends Theorem 1. Indeed, the translation control can not be proven asymptotically stable in the general case. However, it can in the application described in the next section.

5 Application

In this section, we focus on the application of the general control law defined in (14)-(17) to the special configuration of the orthogonal trihedron. This

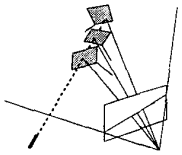


Figure 7: A laser pointer can be considered as a second camera reduced to its optical axis.

configuration appears in the image as a junction of three 2D lines. This has shown to be degenerate for the pose algorithms from lines [3, 4] since depth along the line of sight passing through the center of the trihedron is unobservable.We note $\underline{\mathbf{u}}_c$ the orientation of this line. However, one can still find from the extracted 2D lines (i.e. the $\underline{\mathbf{h}}_i$'s) the 3D orientation of each line (i.e. the $\underline{\mathbf{u}}_i$'s) using the closed-form solution given in [8]. Hence, the previous control law can be applied but when converging does not have any constraint on depth along $\underline{\mathbf{u}}_c$, thus risking to crash the robot against the trihedron.

To observe the depth a simple laser pointer is fixed onto the camera. Thus, the laser spot projects onto a fixed line in the image (Figure 7). Hence, the laser spot in the image can be addressed by a single abscissa s without any kind of laser/camera calibration. It can be seen [1] that s varies linearly when the camera center moves along $\underline{\mathbf{u}}_c$ and the laser point remains on a fixed plane among those of the trihedron. It means that the relative depth between the current and desired position of the camera with respect to the trihedron center is directly observable from the difference between the current abscissa and its value s^* at desired position.

Consequently, we add a third step to the general control law:

$$\Omega = \mu_\omega \sum_{i=1}^{n} \underline{\mathbf{u}}_i \times \underline{\mathbf{u}}_i^* \quad \mu_\omega > 0 \tag{18}$$

$$V = -\mu_v \sum_{i=1}^{n} \underline{\mathbf{u}}_i \times (\underline{\mathbf{h}}_i - \underline{\mathbf{h}}_i^*) + \mu_s (s - s^*) \underline{\mathbf{u}}_c \quad \mu_v > 0 \tag{19}$$

where the terms coming from (14)–(17) ideally vanish and the sign of the scalar gain μ_s is determined by the orientation of the plane hit by the laser beam.

6 Results

In the context of the VIGOR project[1], we plan to apply this three step control scheme (14)–(19) to the positioning of a welding torch with respect to a ship part (Figure 8). We present in Figure 9 simulation results to show the behaviour of the control law. The first step drives the orientation error (i.e. on the $\underline{\mathbf{u}}_i$'s) to zero with no effect on the trajectory but with drifts of the other errors (a) and displacement to the right in the image (c). Then, translation control occurs (b), reducing the error in terms of image alignment (a,c) and approaching the goal position (d). Finally, the laser spot is driven to its goal position (a) with no effect in the image and a straight line approach in space (d).

[1] http://www.inrialpes.fr/VIGOR

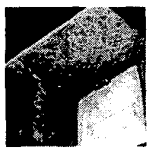
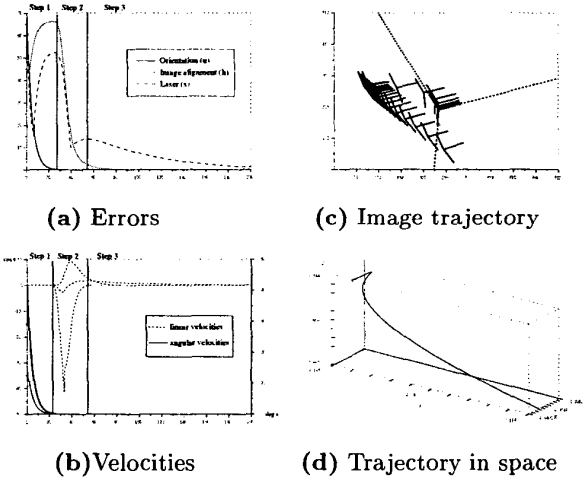


Figure 8: The kind of image to be dealt with.



(a) Errors (c) Image trajectory

(b)Velocities (d) Trajectory in space

Figure 9: Behaviour of the proposed control scheme.

7 Conclusion

In this work, we represented 3D lines with an alternative formulation of their Euclidean Plücker coordinates. This allowed a simple definition of a new alignment of lines. This formulation also provided a control law which realizes this alignment and is nevertheless simpler than previous control laws realizing the sole 2D alignment. The soundness of the approach is illustrated in simulation by the positioning of a camera with respect to an orthogonal trihedron

The control law, mixing 2D and 3D information, is

defined in three cascaded steps with orientation control first. This may lead the lines to leave the field of view. Therefore, it may be valuable to run the three steps simultaneously. The extension of the convergence theorems will then require the use of cascaded system analysis [13].

The present work did not analyze the robustness of the control law with respect to calibration errors and other perturbations, but this is a crucial point which also needs to be studied.

Finally, from the practical point of view, one should try to extend this work to the case where the scene structure is unknown, as it is done for points. This means that Euclidean reconstruction from lines [] should be used. As only orientation is required, it may lead to a simplification of such algorithms.

Acknowledgments

The authors wish to thank E. Marchand from IRISA, Rennes (France) for providing us with his line tracking programs.

References

[1] Nicolas Andreff. *Asservissement visuel.* PhD thesis, Institut National Polytechnique de Grenoble, 1999. To appear.

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

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

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

[10] E. Malis. *Contributions à la modélisation et à la commande en asservissement visuel.* PhD thesis, Université de Rennes I, 1998.

[11] E. Malis, F. Chaumette, and S. Boudet. 2-1/2-D visual servoing. *IEEE Trans. on Robotics and Automation*, 15(2):238–250, 1999.

[12] N. Navab. *Visual Motion of Lines and Cooperation Between Motion and Stereo.* PhD thesis, Université de Paris-Sud, Centre d'Orsay, 1993.

[13] 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.

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

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

[16] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control: the Task Function Approach.* Clarendron Press, Oxford University Press, Oxford, UK, 1990.

[17] R.Y. Tsai and R.K. Lenz. A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. IEEE *Transactions on Robotics and Automation*, 5(3):345–358, 1989.

[18] J. P. Urban, G. Motyl, and J. Gallice. Real-time visual servoing using controlled illumination. *International Journal of Robotics Research*, 13(1):93–100, February 1994.