Dpto. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
C/ María de Luna num. 1
E-50018 Zaragoza
Spain

Internal Report: 1995-V04

Using Direct Methods to Obtain Motion from 3D Lines[1]

J.J. Guerrero, C. Sagüés, F.J. Domínguez

If you want to cite this report, please use the following reference instead:
Using Direct Methods to Obtain Motion from 3D Lines, J.J. Guerrero, C. Sagüés, F.J. Domínguez *Seventh International Conference on Advanced Robotics*, pages 687-693, 1995.

[1]This work was partially supported by project TAP-94-0390 of the Comisión Interministerial de Ciencia y Tecnología (CICYT).

USING DIRECT METHODS TO OBTAIN MOTION FROM 3D LINES

J.J. Guerrero & C. Sagüés & F.J. Domínguez

Dpto. Informática e Ingeniería de Sistemas.
Centro Politécnico Superior
UNIVERSIDAD DE ZARAGOZA
María de Luna 3, E-50015 ZARAGOZA, SPAIN
Phone 34-76-517274, FAX 34-76-512932
email: jjguerrero@mcps.unizar.es

Abstract

In this paper a method to obtain the instantaneous 3D motion of a camera using the brightness information in image regions supporting straigth lines is presented. The 3D location of the lines must be provided to the algorithm, but with a little computation overhead with respect to the image line extraction and using a second image close to the first one, an estimation of the 3D motion is obtained when at least three lines are available. The algorithm does not need neither correspondence determination nor full optical flow computation and uses a direct formulation over line support regions.

Keywords: Dynamic vision, motion from structure, straight lines, direct motion computation.

1 INTRODUCTION

Methods for extracting shape and motion information from vision can be classified as optical flow-based, correspondence-based, and direct methods. In the first two approaches, after the computation of optical flow or the correspondence between features, the structure and motion are determined. Direct methods avoid both correspondence and optical flow computation, and the structure and motion are extracted without intermediate steps.

In [6] some special cases to extract structure from motion with direct methods are addressed. In [10] the direct approach by fixation of a small patch of approximately constant depth is presented. Direct methods are also used to obtain the depth of the scene when the motion is considered to be known. In [11] the accuracy of direct methods to determine the depth is evaluated.

Direct methods are based on the brightness change constraint equation which has been used over surfaces, but this constraint equation only can be considered in image regions where there are great brightness gradients. Thus we propose to use these direct methods over image regions that support straight contours.

It is well known that from line features can not be determined the camera motion and scene structure using only two images. Some experiments have been made using second derivatives of the flow along rectilinear edges but high order derivatives are very sensitive to noise. Locally discrete approaches have also been considered [4], but the image line orientation must be accurate and therefore the global line orientation and location are important. Besides that, some additional constraints (like intersection of lines) are needed. We do not assume additional constraints because we do not determine the motion and the structure simultaneously. We estimate, in a separated way, the location of lines from the camera motion [9] and the camera motion from 3D lines. The method introduces little computational load in addition to the extraction of the line in the first image and it works, at least as well as a correspondence-based approach, in images with small disparities.

In this paper we present a direct approach to extract the 3D instantaneous motion of a camera using the information from two close images, supposed the location of different 3D lines to be known. Only the information of the motion field in normal direction along lines, using the global line location and a direct formulation, is considered.

In §2 we discuss the validity of the brightness change constraint equation over contours and we present the background of direct formulation. In §3 the appropriate line representation that takes into account straight contour in the image and its 3D location, is presented. In §4 the motion equations of the line in image and in space are developed. In §5 the motion of the camera is obtained based on the brightness constraint equation. In §6 some experiments with real images are presented. Finally §7 is devoted to expose conclusions and future developments.

2 BACKGROUND OF THE APPROACH

2.1 Brightness change constraint equation

The brightness change constraint equation ($\frac{dE}{dt} = 0$) has been considered as a reasonable assumption in many situations [5] and many works have been based on it. This constraint supposes that the total time differential of the image irradiance is zero. Sometimes does not work, for example when the illumination conditions change, when there are specular reflections, mutual illumination effects. However, it can be a very reasonable constraint on which to base visual motion computation. It has been said that even for a lambertian surface that is not purely translating, this constraint does not hold [3], but other alternate constraints ($\frac{d(|\nabla E|)}{dt} = 0$) also appears to be inexact. Moreover, it is not possible to judge whether there is much advantage using it, instead of the brightness change constraint equation [8]. They consider that in practice, all one can say is that com-

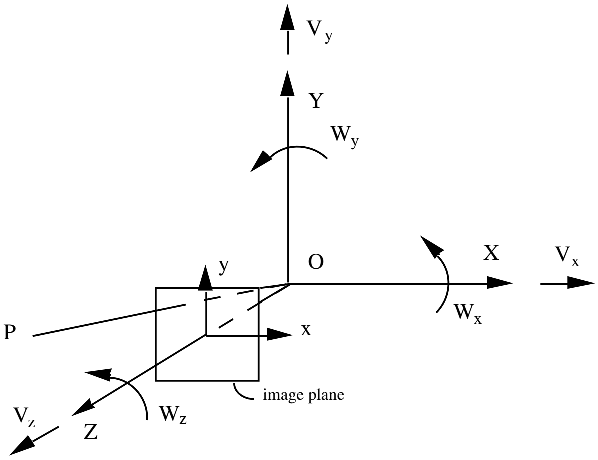

Figure 1: *Pinhole camera model*

puting visual motion at image *irradiance discontinuities* gives the best chance that the visual motion will equate to the projected motion.

In [7] different illumination and reflection conditions are discussed. The continuity of brightness depends to a great extent on the illumination conditions and reflection properties of the observed objects. A general continuity equation can not be established, but if the gray value gradient is large, the influence of many of the additional terms is small. He concludes that the determination of the velocity is most reliable for *steep gray edges* while it may be distorted in regions with only small brightness gradients. In many cases the illumination can be arranged in such a way that the simple brightness change constraint equation is valid where there is great gradient.

Besides that, the motion field extraction presents the known aperture problem. The visual motion field is not fully defined at each pixel since we have only one equation between its coordinates. Only the normal flow to the contours of iso-brightness can be recovered. In order to solve this new problem, researchers have tried to suppose a motion field, close to the real one when possible, by imposing smoothness constraints. Normally these smoothness constraints do not work well near contours, where the brightness change equation could be used in practice. In our approach, this problem is circumvented because the motion field is mixed with the structure which is supposed to be known.

We adopt the brightness change constraint equation in regions supporting steep contours of 3D lines with known structure. We use direct optical flow techniques that in practice can be considered valid in edges and they allow the motion computation when there is little disparity in the image.

2.2 Camera and motion models

The pinhole camera model with a planar screen is illustrated in figure 1. The origin of the camera coordinate system $OXYZ$ is on the projection center of the camera. The Z axis is aligned with the optical axis and the focal length is considered to be the unit. Without loss of generality, the image plane is in front of the pinhole to avoid dealing with inverted images. A point in the scene with (X, Y, Z) coordinates in that system is projected in the image with $(x, y, 1)$ coordinates, being:

$$x = \frac{X}{Z} \quad , \qquad y = \frac{Y}{Z} \tag{1}$$

Assuming that any given point on the 3D object appears in the successive image frames with a constant brightness, we can formulate the brightness change constraint equation as:

$$\frac{\delta E}{\delta x}\frac{dx}{dt} + \frac{\delta E}{\delta y}\frac{dy}{dt} + \frac{\delta E}{\delta t} = E_x\, u + E_y\, v + E_t = 0$$

where E denotes the brightness of a point in the image.

Motion field can be originated by object or camera motion. The camera is supposed to move with respect to the scene and its motion to be composed by a translation $\mathbf{t} = [V_x, V_y, V_z]^T$ and a rotation $\mathbf{w} = [W_x, W_y, W_z]^T$ in the camera reference system.

3 REPRESENTATIONS OF IMAGE LINES AND 3D LINES

We want a line representation combining the information of the line in the image with its 3D information in a similar way as considered in [4]. To locate the projection of a line in the image plane we need two parameters and to locate a 3D line we need at least four parameters. Thus, we use two parameters of its projection in the image and two additional depth parameters.

Firstly we attach a reference system to the projection plane of the line in the image by making two rotations ($Rot(z, \phi) Rot(y, \theta)$) in the camera reference system. Where ϕ is the orientation of the line in the image with respect to the y axis, and $tan\theta$ is the distance in the image from the origin to the line (figure 2). The x axis of the new reference system will be perpendicular to the plane of projection of the image line. If we name $\mathbf{n}$ the vector of this reference system in the x direction, it turns out that the image line has the equation:

$$(x, y, 1) \cdot \mathbf{n} = 0 \tag{2}$$

which can be rewritten as $x\, cos\phi + y\, sin\phi - tan\theta = 0$

We take ϕ in such a way that the $\mathbf{n}$ vector points in the direction of the spatial brightness gradient in image from dark to light ($-\pi < \phi \leq +\pi$). The angle θ takes values from $-\frac{\pi}{2}$ to $+\frac{\pi}{2}$. Normally using real cameras that have a small field of view, the angle θ will be small for all lines that appear in image. Thus, an image line which can have two alternate

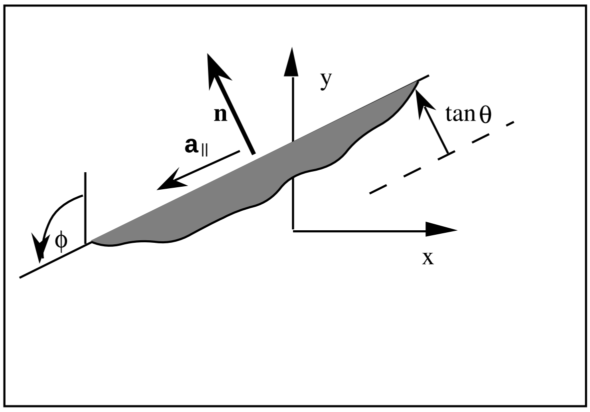

Figure 2: *Straight line in the image*

representations geometrically, when we take into account the sign of the contrast only one representation remains valid (figure 2).

A second reference system is attached to the 3D line. The two parameters named above are combined with two additional parameters, in order to obtain a representation of the 3D line. Thus, we can define a third rotation $Rot(x, \psi)$ $(0 \leq \psi < \pi)$ in such a way that the new z axis points in the direction of the line and the new y is perpendicular to the 3D line (from the camera to the scene) (figure 3).

Another parameter is defined as the distance from the origin of the camera reference system to the 3D line (d). This parameter is always greater than zero because the line is always in front of the camera ($d_z = d\ cos\theta\ sin\psi \geq 0$ since $cos\theta \geq 0$ and $sin\psi \geq 0$).

Thus the transformation from the camera reference system to the reference attached to the line with the previous considerations is $\mathbf{T}_{cl} = Rot(z, \phi)Rot(y, \theta)Rot(x, \psi)Trasl(y, d)$. And expressed as homogeneous matrix:

$$\mathbf{T}_{cl} = \begin{pmatrix} c\phi c\theta & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta c\psi + s\phi s\psi & d_x \\ s\phi c\theta & s\phi s\theta s\psi + c\phi c\psi & s\phi s\theta c\psi - c\phi c\psi & d_y \\ -s\theta & c\theta s\psi & c\theta c\psi & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & d\,\mathbf{o} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where c is cos and s is sin.

Some of the advantages of this representation are:

- Two out of the four parameters defining the 3D line are also used to define its projection in the image.
- We have not singularities for the lines that could appear in the image, because $|\theta|$ is always less than $\frac{\pi}{2}$ that is the sole singularity of the RPY angles.

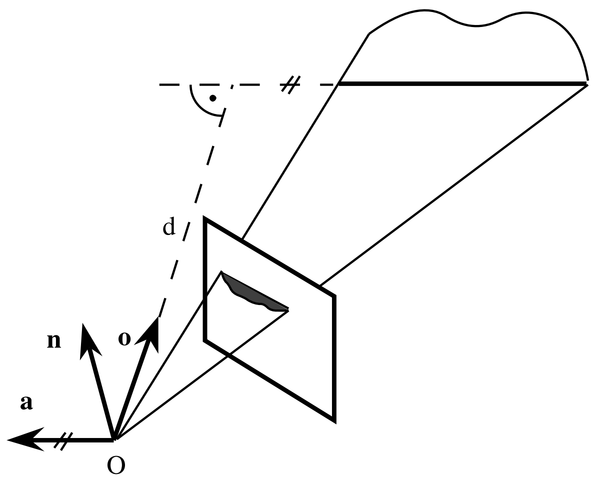

Figure 3: *Representation of the 3D line using its plane of projection*

- The cyclic symmetries of the plane and the line are avoided using the contrast sign of the line in the image and making that the **o** vector (perpendicular to the line from the origin) points from the image to the scene.

4 STRAIGHT EDGE MOTION

We propose a direct method to extract the motion parameters of a straight edge, supposed its location in the camera reference system to be known. The aim of this approach is to obtain a complete measure of the camera motion using two images.

It is well known that at least three lines are needed in order to determine the location of the camera, supposed the line location to be known. In a similar way, from the motion field of at least three lines we can recover the motion of the camera. We use from each line only the 3D motion information it can supply, as a function of its location. Thus firstly we analyze the motion field of a line and its location with relation to the 3D motion of the camera.

4.1 Normal motion of an edge in the image

Let us name u_n the normal flow of a point of the edge in the image. Assuming the brightness change constraint equation, the normal flow can be formulated as:

$$u_n = \frac{-E_t}{+\sqrt{{E_x}^2 + {E_y}^2}} \tag{3}$$

being E_x, E_y, E_t the spatial and temporal brightness gradient in the image.

Taking into account the representation adopted, the normal flow u_n of a generic point (x, y) of the line in the image, can be written as $u_n = u\ cos\phi + v\ sin\phi$. Deriving the image line equation $x\ cos\phi + y\ sin\phi - tan\theta = 0$, we obtain:

$$u\ cos\phi - x\ sin\phi\ \dot{\phi} + v\ sin\phi + y\ cos\phi\ \dot{\phi} - (1 + tan^2\theta)\dot{\theta} = 0$$

that is:

$$\begin{aligned} u_n &= (1 + tan^2\theta)\ \dot{\theta} - (y\ cos\phi - x\ sin\phi)\ \dot{\phi} \\ &= (1 + tan^2\theta)\ \dot{\theta} - r_{\|}\ \dot{\phi} \end{aligned} \tag{4}$$

being $r_{\|}$ the distance from the generic point in the line (x, y) to the point in the line closest to the image center (figure 2).

4.2 3D motion with relation to line motion in the image

It has been shown [2] the relationship that exists between the three-dimensional structure and the kinematics of a line moving rigidly in space, with the two dimensional structure and kinematics (motion field) of its projection in the image. This relation can be summarized in the *line motion field equation* [2]. This equation, when the camera moves rigidly in space and considering our line representation can be expressed as follows:

$$\dot{\mathbf{n}} = -\mathbf{w} \times \mathbf{n} + \frac{\mathbf{t} \cdot \mathbf{n}}{d}\ \mathbf{o} \tag{5}$$

being $\times$ the cross product between vectors and $\cdot$ the dot product.

This equation (5) gives the information we can obtain from a 3D line, its image and its motion field in the image. Thus, we can recover **w** in the direction of the perpendicular from the origin to the line (**o**) without knowledge of depth and translational motion. Moreover, the rotation in the camera reference system around an axis parallel to the 3D line (**a**, figure 3) is coupled with translational motion in the direction of the normal to the plane of projection (**n**).

Therefore, from the motion field of a line it is only possible to recover the rotation around one direction and the translation in other direction. The rotation can not be obtained neither around the direction of the 3D line (symmetry of the feature), nor around **n** (symmetry of the image line observation). The translation can not be obtained neither in the direction of the 3D line (symmetry of the line), nor in the direction of its perpendicular which is contained in the plane of projection (it does not modify the projected line).

From (5), making the dot product of vector $\dot{\mathbf{n}}$ with the unit vector $\mathbf{a}$ in direction of the line, and with the unit vector in the direction perpendicular to the line from the origin $\mathbf{o}$ ($\dot{\mathbf{n}} \cdot \mathbf{n} = 0$), we have the two motion parameters obtained from each line :

$$\dot{\mathbf{n}} \cdot \mathbf{a} = (-\mathbf{w} \times \mathbf{n}) \cdot \mathbf{a} + \frac{\mathbf{t} \cdot \mathbf{n}}{d}\, \mathbf{o} \cdot \mathbf{a} = \mathbf{w} \cdot \mathbf{o} \tag{6}$$

$$\dot{\mathbf{n}} \cdot \mathbf{o} = (-\mathbf{w} \times \mathbf{n}) \cdot \mathbf{o} + \frac{\mathbf{t} \cdot \mathbf{n}}{d}\, \mathbf{o} \cdot \mathbf{o} = -\mathbf{w} \cdot \mathbf{a} + \frac{\mathbf{t} \cdot \mathbf{n}}{d} \tag{7}$$

We rename these two values for the $(l-th)$ line respectively as w_{ol} for the rotation in the direction of the $\mathbf{o}$ vector (6), and t_{nl} for translation in the $\mathbf{n}$ direction coupled with the rotation around $\mathbf{a}$ (7).

On the other hand we can compute the line motion field in a similar way to [4]:

$$\begin{aligned} \dot{\mathbf{n}} &= \begin{bmatrix} -cos\theta\ sin\phi\ \dot{\phi} - cos\phi\ sin\theta\ \dot{\theta} \\ cos\theta\ cos\phi\ \dot{\phi} - sin\phi\ sin\theta\ \dot{\theta} \\ -cos\theta\ \dot{\theta} \end{bmatrix} \\ &= cos\theta\ \dot{\phi}\, \mathbf{l} - \dot{\theta}\, \mathbf{n} \times \mathbf{l} \end{aligned}$$

being $\mathbf{l} = (-sin\phi, cos\phi, 0)$ the direction of the line in the image.

Since, as can be deduced, $\mathbf{l} \cdot \mathbf{a} = -sin\psi$ and $\mathbf{l} \cdot (\mathbf{a} \times \mathbf{n}) = cos\psi$, and making some algebraic manipulations, we arrive at:

$$\begin{aligned} \dot{\mathbf{n}} \cdot \mathbf{a} &= cos\theta\ \dot{\phi}\, \mathbf{l} \cdot \mathbf{a} - \dot{\theta}\, (\mathbf{n} \times \mathbf{l}) \cdot \mathbf{a} = \\ &= -cos\theta\ \dot{\phi}\ sin\psi - \dot{\theta}\ cos\psi \end{aligned} \tag{8}$$

$$\begin{aligned} \dot{\mathbf{n}} \cdot \mathbf{o} &= cos\theta\ \dot{\phi}\, \mathbf{l} \cdot (\mathbf{a} \times \mathbf{n}) - \dot{\theta}\, (\mathbf{n} \times \mathbf{l}) \cdot (\mathbf{a} \times \mathbf{n}) = \\ &= cos\theta\ \dot{\phi}\ cos\psi - \dot{\theta}\ sin\psi \end{aligned} \tag{9}$$

From (8) and (9) equations, we can compute both $\dot{\phi}$ and $\dot{\theta}$ as:

$$\dot{\phi} = \frac{-w_{ol}\ sin\psi + t_{nl}\ cos\psi}{cos\theta} \tag{10}$$

$$\dot{\theta} = -t_{nl}\ sin\psi - w_{ol}\ cos\psi \tag{11}$$

Substituting these expressions in (4), we arrive at an expression that connects directly normal flow of a point of the line with the three-dimensional structure and kinematics of the line when the camera moves rigidly in space:

$$\begin{aligned} u_n\ cos^2\theta + t_{nl}\ (sin\psi + r_{\|}\ cos\theta\ cos\psi) + \\ + w_{ol}\ (cos\psi - r_{\|}\ cos\theta\ sin\psi) = \quad 0 \end{aligned} \tag{12}$$

4.3 Brightness change constraint equation of the line

We can now, instead of computing the normal optical flow along the line in the image, take the normal flow operator (3) and the *line normal motion field equation* (12) and obtain an equation involving the 3D location parameters of the line, with the motion parameters of the camera associated to each line (only these ones that could be computed from its motion field, t_{nl}, w_{ol}) and with some measurable quantities of the image related to its brightness gradients. By mixing these equations we have for each pixel (x, y) of the line in the image:

$$\begin{aligned} E_t\ cos^2\theta = t_{nl} \sqrt{{E_x}^2 + {E_y}^2}\ (sin\psi + cos\theta\ cos\psi\ r_{\|})+ \\ +w_{ol}\ \sqrt{{E_x}^2 + {E_y}^2}\ (cos\psi - cos\theta\ sin\psi\ r_{\|}) \end{aligned} \tag{13}$$

where $r_{\|} = y\ cos\phi - x\ sin\phi$.

This is our fundamental equation to determine the camera motion from brightness information of pixels that support a straight line in the image.

5 MOTION DETERMINATION

To extract the motion information from straight lines we fully process one image, and we use a second one with a little processing. The first step in the procedure proposed is the extraction of spatial gradients to segmentate the image. Pixels are grouped into regions of similar direction of brightness gradient [1], with the gradient magnitude larger than a threshold. Two overlapping sets of partitions, with a post grouping process, are used in order to avoid the problems related to the arbitrary boundary of fixed partitions. Thus we have line-support regions (LSR), containing all information available of the straight contours in one image that will be the basis of the method. The LSR concept provides also the opportunity of mixing both feature based and flow based methods of motion analysis in an integrated way.

5.1 Direct approach

When we have the LSR of a line and its 3D location (ϕ, θ, ψ, d) in the camera reference system, we can evaluate directly the (13) equation and with a linear least-squares approach along each LSR, the two motion parameters t_{nl}, w_{ol} of each line can be obtained. In this minimization a weighting factor (N_w) proportional to the gradient magnitude is used, in such a way that, only pixels on the ridge of a LSR make large contributions to the motion determination.

Thus the minimization along the LSR is the following:

$$J_d = \sum_{x,y}^{LSR} [-E_t \, cos^2\theta + t_{nl} \, FT(x,y) + w_{ol} \, FW(x,y)]^2 N_w \tag{14}$$

being:

$$\begin{aligned} FT(x,y) &= \sqrt{{E_x}^2 + {E_y}^2} \; (sin\psi + cos\theta \; cos\psi \; r_{\|}) \\ FW(x,y) &= \sqrt{{E_x}^2 + {E_y}^2} \; (cos\psi - cos\theta \; sin\psi \; r_{\|}) \end{aligned}$$

Deriving with respect to t_{nl} and w_{ol} and equating to zero we arrive at a linear set of equations that is easily solved:

$$\begin{bmatrix} S_{t2} & S_{tw} \\ S_{tw} & S_{w2} \end{bmatrix} \begin{bmatrix} t_{nl} \\ w_{ol} \end{bmatrix} = \begin{bmatrix} S_{et} \\ S_{ew} \end{bmatrix} \tag{15}$$

where the integral factors are shown in appendix 1.

With this direct method only a little computation cost is added to the computation made on the first image. Besides that, the search of correspondences that would be needed using a classical approach is avoided.

5.2 Global camera motion determination

From the direct method proposed we have determined two motion parameters t_{nl}, w_{ol} in each line by a minimization in the image region that supports the line. Using at least the motion parameters of three straight edges, whose 3D location is known, we can recover the 3D camera motion velocity. Thus, from each line $(l - th)$ we have a linear equation in terms of the rotation velocity:

$$\mathbf{w} \cdot \mathbf{o}_l = w_{ol}$$

Using three lines, the rotation velocity can be obtained by solving a linear set of equations. When the rotation velocity is known we can recover from each line the contribution to the rotation ($w_{al} = \mathbf{w} \cdot \mathbf{a}_l$) of the parameter t_{nl}. Therefore we have an equation from each line that allows to determine the translation velocity ($\mathbf{t}$) solving a linear set of equations like the following:

$$\mathbf{t} \cdot \mathbf{n}_l = (t_{nl} + w_{al}) \; d_l$$

When more than three lines are available a least-squares approach enables a more robust determination of the camera motion. In appendix 2 the least-squares formulation used is shown.

To make the method to work the lines must not be parallel in 3D. Besides that, it is better to take the largest lines and as separated as possible in the image in order to have better direction determination of rotation velocity.

	TEST 1			*TEST 2*		
	C	M	σ	C	M	σ
V_x	0.5	4.77E-1	3.13E-2	0.0	-1.25E-2	1.23E-1
V_y	0.0	-2.50E-2	1.52E-2	0.0	1.58E-1	7.09E-2
V_z	0.0	2.45E-1	4.47E-2	2.0	2.05	2.26E-1
W_x	0.0	-7.11E-5	1.57E-4	0.0	1.57E-4	1.42E-4
W_y	0.0	-1.58E-4	9.32E-5	0.0	1.08E-4	3.91E-4
W_z	0.0	1.04E-4	9.40E-5	0.0	-5.16E-4	2.81E-4

Table 1: *Commanded by the robot (C), Mean (M) and standard deviation (σ) of the motion parameters calculated (mm. and rad. per frame)*

		Direct method		*Correspondence*	
	C	M	σ	M	σ
V_x	0.0	-4.60E-2	1.61E-2	-2.30E-1	2.53E-2
V_y	0.0	-6.67E-3	4.21E-2	2.96E-2	3.67E-2
V_z	1.0	9.12E-1	7.47E-3	1.15	4.71E-2
W_x	0.0	-5.38E-4	1.02E-4	-3.81E-4	1.36E-4
W_y	0.0	1.18E-4	2.47E-5	8.52E-4	1.85E-4
W_z	0.0	2.59E-5	1.79E-4	-1.30E-3	4.28E-4

Table 2: *Commanded by the robot (C), Mean (M) and standard deviation (σ) of the motion parameters calculated (mm. and rad. per frame) with the proposed direct method compared with a correspondence based approach over the same images*

6 EXPERIMENTAL RESULTS

Some experiments with real images have been made using a camera of 12 *mm.* focal length and an image size of (370,256). The illumination of the scenes was mainly natural (it arrived through the ceiling and the windows) complemented with some artificial light sources. The motion has been made by means of a PUMA-560 robot with a camera in the hand. The robot controller is not accurate enough to verify exactly the algorithm (specially with rotations), but could be considered acceptable. We present results with pure translations because the checking of results is easier than with other motions.

The first scene used corresponds with a pyramid on a white table observed from 300 *mm.* on the top. In table 1 it can be seen the mean and standard deviation of the translation and rotation velocity obtained from 9 experiments with each commanded motion. The number of lines per frame was 7. The disparity in the image is around one pixel. It can be observed using similar image disparity that the results are better carrying out translation in Z direction than doing translation parallel to the image plane. It has been observed when the disparity raises in the order of 2 pixels, the results get worse and the velocity obtained was always smaller than the expected. Usually the worst translation estimation is obtained in the Z direction, and it often appears the effect of coupling between translation and rotation parallel to the image plane. Table 2 provides a comparison between the proposed direct method and a correspondence-based approach, when the maximum disparity is close to 0.5 pixels. It can be observed that the direct methods appears a little more stable and accurate than the correspondence method.

Figure 4: *First image of the laboratory scene and the lines used. The second image has a little disparity with this*

Some experiments with an scene of our laboratory (figure 4) have been made. The idea was to make collaboration between this motion algorithm and a stereo. The stereo provides the 3D location of lines and the direct method obtains the camera velocities. At the moment, we have only qualitatively satisfactory results. More experiments in this direction are expected in the future.

7 CONCLUSIONS

In this paper we have presented an algorithm to extract the camera motion using straight edges by direct methods, from two close monocular images. The proposed approach needs an estimation of the 3D location of lines, but it determines the motion using directly brightness information of the images.

Feature support regions in the image are used and with a least squares approach the motion is directly extracted when at least three 3D lines are available. The method takes only a little additional computational cost over the line extraction in the first image, and allows to estimate the instantaneous velocities.

Experimental results using real images are presented. A comparison of the results (when the image disparity is about a half pixel) of this method and an equivalent motion determination algorithm based on line correspondences has shown the direct method works a little better being the computation time of our method shorter.

More experiments in this direction and the collaboration of our method with a mobile stereo when the image disparity between frames is small, are the works we expect to do in the future.

Acknowledgements

This work was partially supported by project TAP-94-0390 of the Comisión Interministerial de Ciencia y Tecnología (CICYT).

Appendix 1

Integral factors used in the minimization, obtained from the LSR in a sequential way as:

$$
\begin{aligned}
S_{t2} &= \sum_{x,y}^{LSR} [FT(x,y)]^2 \; N_w \\
S_{w2} &= \sum_{x,y}^{LSR} [FW(x,y)]^2 \; N_w \\
S_{tw} &= \sum_{x,y}^{LSR} [FT(x,y) \; FW(x,y)] \; N_w \\
S_{et} &= \sum_{x,y}^{LSR} [FT(x,y)] \left[-E_t \; cos^2\theta\right] \; N_w \\
S_{ew} &= \sum_{x,y}^{LSR} [FW(x,y)] \left[-E_t \; cos^2\theta\right] \; N_w
\end{aligned}
$$

Appendix 2

When the motion parameters of more than three lines are available a least-squares approach allows to obtain the camera motion as (l represents each line):

$$
\begin{aligned}
\mathbf{x}^T &= \left[\mathbf{w}^T, \mathbf{t}^T\right] \\
\mathbf{z}_l^T &= [w_{ol}, t_{nl}] \\
\mathbf{z}^T &= \left[\mathbf{z}_1^T, \mathbf{z}_2^T, ..., \mathbf{z}_l^T, ...\right]
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{A}_l &= \begin{bmatrix} \mathbf{o}_l^T & , & 0, & 0, & 0 \\ -\mathbf{a}_l^T & , & & \frac{\mathbf{n}_l^T}{d_l} & \end{bmatrix} \\
\mathbf{A}^T &= \left[\mathbf{A}_1^T, \mathbf{A}_2^T, ..., \mathbf{A}_l^T, ...\right]
\end{aligned}
$$

Thus the motion parameters that minimize the norm $||\mathbf{A}\ \mathbf{x} - \mathbf{z}||$ are:

$$\mathbf{x} = (\mathbf{A}^T\ \mathbf{A})^{-1}\ \mathbf{A}^T\ \mathbf{z}$$

References

[1] J.B. Burns, A.R. Hanson, and E.M. Riseman. Extracting straight lines. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(4):425–455, 1986.

[2] O. Faugeras, N. Navab, and R. Deriche. Information contained in the motion field of lines and the cooperation between motion and stereo. *International Journal of Imaging Systems and Technology*, 2:356–370, 1991.

[3] Olivier Faugeras. *Three-Dimensional Computer Vision. A Geometric Viewpoint.* The MIT Press, Massachusetts, 1993.

[4] B. Giai-Checa and T. Viéville. 3d-vision for active visual loops using locally rectilinear edges. In *The 7th IEEE Symposium on Intelligent Control*, pages 341–348, Glasgow, 1992.

[5] B.K.P. Horn. *Robot Vision.* MIT Press, Cambridge, Mass., 1986.

[6] B.K.P. Horn and E.J. Weldon. Direct methods for recovering motion. *International Journal of Computer Vision*, (2):51–76, 1988.

[7] Bernd Jähne. *Digital Image Processing.* Springer-Verlag, Berlin-Heidelberg, 1993.

[8] D.W. Murray and B.F. Buxton. *Experiments in the Machine Interpretation of Visual Motion.* The MIT Press, Massachusetts, 1990.

[9] C. Sagüés and J.J. Guerrero. Locating 3d edges by direct methods in motion based vision. In *1993 IEEE International Conference on Systems, Man and Cybernetics*, pages 511–516, Le Touquet-France, October 1993.

[10] M.A. Taalebinezhaad. Direct robot motion vision by fixation. In *IEEE International Conference on Robotics and Automation*, pages 626–631, Sacramento, California, April 1991.

[11] E.J. Weldon and H. Liu. How accurately can direct motion vision determine depth. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 613–618, 1991.