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

Internal Report: 1998-V01

Direct method to obtain straight edges depth from motion[1]

Guerrero J.J., Sagüés C.

If you want to cite this report, please use the following reference instead:
Direct method to obtain straight edges depth from motion, Guerrero J.J., Sagüés C., *Optical Engineering*, Vol. 37(7), pages 2124-2132, 1998.

[1]This work was supported by projects TAP97-0992

Direct method to obtain straight edges depth from motion

J.J. Guerrero & C. Sagüés

Dpto. de Informática e Ingeniería de Sistemas
Centro Politécnico Superior, UNIVERSIDAD DE ZARAGOZA
María de Luna 3, E-50015 ZARAGOZA, SPAIN
Phone 34-976-761940, Fax 34-976-762111
email: jguerrer@posta.unizar.es, csagues@posta.unizar.es

Abstract

We develop a method to estimate the three-dimensional location of straight edges captured from a mobile camera. The location uncertainty is also computed. A new direct formulation is presented that avoids both full optical flow and correspondence computations. The algorithm uses image regions that support straight edges and can be applied to small, known camera motions. When the image disparity is small, coherent depth results are obtained in a way more efficient, than with a correspondence-based approach.

Keywords

Dynamic vision, structure from motion, straight edges, brightness constraint, direct methods.

1 Introduction

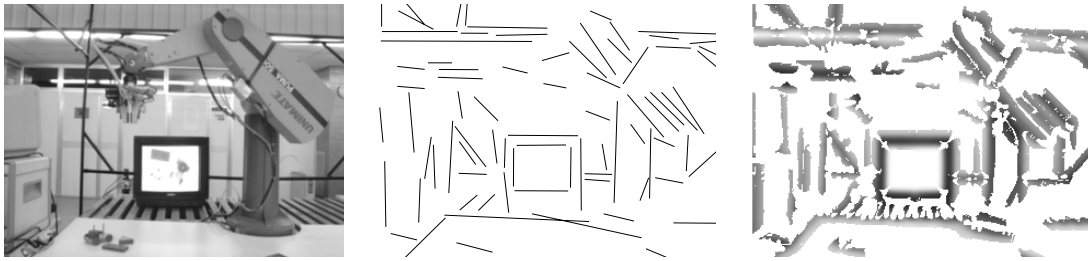
Methods to extract structure and motion information from vision can be classified as optical flow-based, correspondence-based, and direct methods. The first two approaches determine the structure and motion after the computation of the projected motion. On the other hand, direct methods enable us to extract 3-D information directly from image brightness, avoiding the computation of both correspondence and optical flow.

Direct methods presented by Negahdaripour and Horn [9] have been used to solve several specific problems of motion and/or structure from mobile vision [6]. Direct methods were also used to obtain scene depth when the camera motion is known [16]. Another direct approach consists of using the fixation of a small patch of approximately constant depth to determine motion and structure [14]. Recently, methods with a brightness model that enables time brightness variation were also proposed [10].

The former techniques employ a motion constraint equation based on the brightness continuity assumption ($\frac{dE}{dt} = 0$). This constraint has been questioned by some researches [3], but many works are based on it [5]. Brightness continuity depends largely on the lighting conditions and reflection properties of the observed objects. A general continuity equation cannot be established, but if the gray-value gradient is large, the influence of many of the additional terms is small [7]. Many authors conclude that the computation of the motion using the simple brightness constraint is reliable for steep gray edges, while it may be distorted in regions of small brightness gradients, even with complex models.

Direct methods have similarities with optical flow techniques, but they avoid certain computational difficulties in the computation of optical flow. In particular, it is not necessary to assume that the optical flow field is smooth. This usual assumption is often violated near object boundaries, where large brightness gradients are usually extracted.

We adopt the brightness continuity assumption only in regions supporting steep edges [13]. The line support region concept, used previously to obtain straight edges [1], provides the opportunity of mixing both feature based and flow based methods in an integrated way (Fig. 1). In this paper, we propose a



Gray level image | Image of lines | Line support regions

Figure 1: Classical perception methods can be improved using both the geometric representation of straight edges and the intensity of line support regions. In this way the potentiality of geometric features is maintained and all the information concerning the edge in the image is considered.

direct approach to extract the 3-D localization of straight lines by using images whose relative motion is known. The brightness information from regions supporting the image edges is evaluated, linking the different steps in the computation and avoiding rejecting information in intermediate steps. At the same time, a new displacement-based formulation to determine depth from motion is proposed.

A summary of the paper is as follows. Section 2 presents the displacement-based formulation to obtain directly depth from motion. In Sec. 3, we obtain the localization of straight edges by using this direct method. In Sec. 4, depth perception uncertainty is obtained. Section 5 presents some experimental results. Finally, Sec. 6 presents the conclusions.

2 Depth computation from a known motion

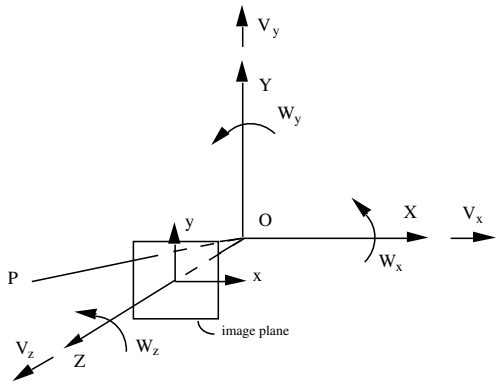


Figure 2: Pinhole camera model.

We adopt a pinhole camera model with a planar screen. The camera model is illustrated in Fig. 2. The origin of the camera reference 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. A point in the scene with (X, Y, Z) coordinates in that system is projected on the image with $(x, y, 1)$ coordinates

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

Let us suppose that the camera translates following $\mathbf{t}$ and rotates according to $\mathbf{R}$, both expressed in the camera reference system. Thus, a 3-D point appears in the first image with $\mathbf{p} = (x, y, 1)^T$ coordinates and with $(x', y', 1)$ coordinates in the second image. These are

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \frac{1}{Z'} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \frac{Z\,\mathbf{R}^T\,\mathbf{p} - \mathbf{R}^T\mathbf{t}}{\hat{\mathbf{z}} \cdot (Z\,\mathbf{R}^T\,\mathbf{p} - \mathbf{R}^T\mathbf{t})} \tag{2}$$

where $\hat{\mathbf{z}}$ is the unit vector in the focal axis direction.

Therefore, the displacement in the image is

$$\begin{bmatrix} x'-x \\ y'-y \end{bmatrix} = \frac{1}{Z\ z_{dr} - t_z^r} \begin{bmatrix} Z\ x_{dr} - Z\ x\ z_{dr} - t_x^r + x\ t_z^r \\ Z\ y_{dr} - Z\ y\ z_{dr} - t_y^r + y\ t_z^r \end{bmatrix} \tag{3}$$

where $(x_{dr}, y_{dr}, z_{dr})^T = \mathbf{R}^T\ \mathbf{p}$; $(t_x^r, t_y^r, t_z^r)^T = \mathbf{R}^T\ \mathbf{t}$.

Assuming that any given point on the 3-D object appears in the successive image frames with the same brightness, we can formulate the brightness constraint equation [5]. Assuming that a time unit has elapsed between the two images, it turns out

$$\frac{\delta E}{\delta x}\ \frac{dx}{dt}\ +\ \frac{\delta E}{\delta y}\ \frac{dy}{dt}\ +\ \frac{\delta E}{\delta t}\ =\ E_x\ (x'-x) + E_y\ (y'-y) + E_t = 0 \tag{4}$$

Instead of computing image displacements, the brightness constraint equation (4) and the displacement field equation (3) are combined and one equation that links image brightness gradients and camera motion with scene depth is obtained. Therefore, after some manipulations we arrive at

$$\frac{\mathbf{s}_d^T\ \mathbf{R}^T\ \mathbf{t}}{Z} - \mathbf{s}_d^T\ \mathbf{R}^T\ \mathbf{p}\ \ =\ \ 0 \tag{5}$$

where $\mathbf{s}_d = (-E_x, -E_y, x\ E_x + y\ E_y - E_t)^T$ is extracted from the brightness of the images.

Using this expression, the depth Z can be obtained directly from the images as a function of $\mathbf{s}_d$ and the camera motion as follows

$$Z = \frac{\mathbf{s}_d^T\ \mathbf{R}^T\ \mathbf{t}}{\mathbf{s}_d^T\ \mathbf{R}^T\ \mathbf{p}} \tag{6}$$

This can be considered as the displacement-based version of the motion constraint equation [9]. Due to the discrete nature of the image acquisition system, specially in the temporal dimension, the proposed expression turns out more suitable than the classical equation. At the same time, this formulation avoids some limitations of the continuous formulation, related to the field of view and to the translation in the Z direction[4].

To obtain reasonable values of the temporal gradient, the image disparities must be small. It implies that, in a general situation the camera translation must be small. However, large translations could be used if a control procedure of fixation in the image were available [12, 14]. On the other hand, the camera rotation is not so critical because its effect can be predicted without knowledge of the scene depth.

The proposed formulation (6) enables us to consider a general camera rotation. However, when the camera rotation is large and the translation is small, it is necessary to compensate the rotation before obtaining the temporal gradient. Thus, in this case, the depth could be obtained as

$$Z = \frac{\mathbf{s}_d^T\ \mathbf{t}}{\mathbf{s}_d^T\ \mathbf{p}} = \frac{\mathbf{s}_d^T\ \mathbf{t}}{-E_t}$$

where the temporal gradient can be calculated from the intensity in two consecutive images as, $E_t(x,y) = E_2(\frac{x_{dr}}{z_{dr}}, \frac{y_{dr}}{z_{dr}}) - E_1(x,y)$.

3 Extraction of 3-D lines with a mobile camera

The aim of the proposed method is to obtain the 3-D structure of straight lines from a mobile camera, estimating also the uncertainty of their locations. The camera motion is supposed to be known. To extract 3-D straight lines we segment one image into line support regions, and we use at least a second image to obtain the temporal brightness gradient.

The image segmentation algorithm employed here is based on the work of Burns [1]. The first step in the procedure is the extraction of spatial gradients. Afterwards, pixels having gradient magnitude

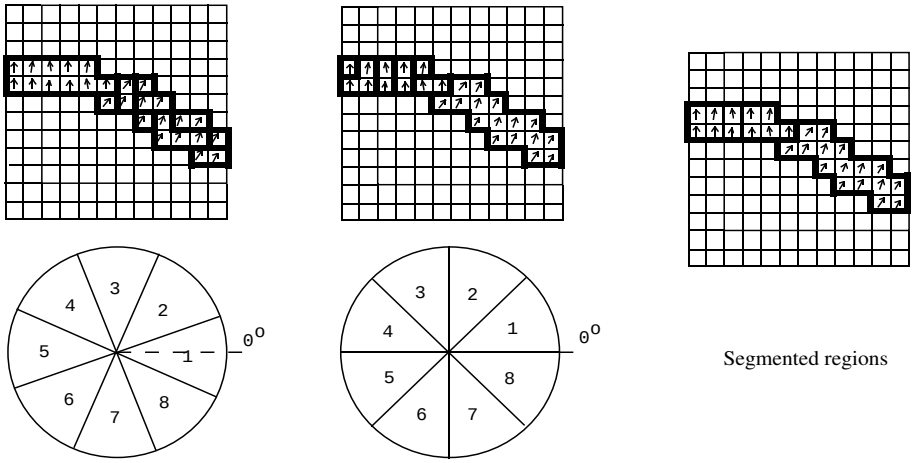


Figure 3: Image segmentation into LSR by using two fixed and overlapped partitions of the gradient orientation.

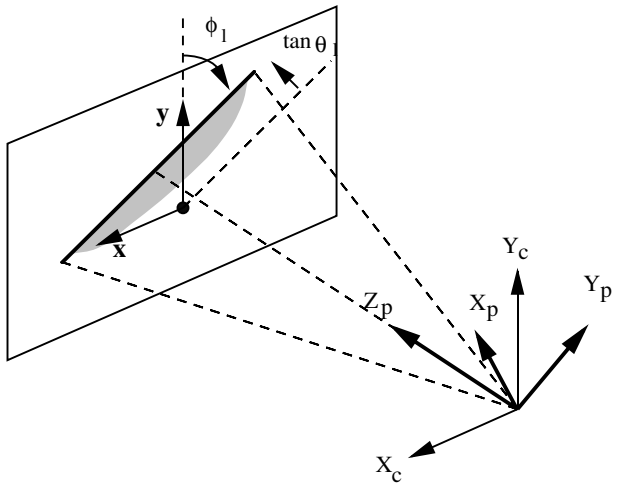


Figure 4: Projected line representation.

larger than a threshold are grouped into regions of similar direction of brightness gradient. Segmentation is globally made using fixed partitions of gradient orientation. Two overlapping sets of partitions, with a post selecting process are used in order to avoid the problems related to the arbitrary boundary of fixed partitions (Fig. 3). From both segmentations, support regions giving a longer interpretation are selected in a subsequent process. In this way, we have the image segmented into line support regions (LSR). Each LSR (consisting of points with similar gradient direction in the neighborhood of an straight edge) contains all available information in the image about the straight edges (Fig. 1).

After image segmentation, we may obtain the 3-D depth Z of each point within the line support region from equation (6). However, such estimates can be erroneous because of various sources of noise (quantization, finite difference approximations, etc.) and use of very local information[16]. Typically, depth uncertainty is much larger than the uncertainty in the 2-D projected position. Thus, we propose to first extract the 2-D straight line in one image from the LSR, which we can do with good accuracy. In a second step, we then calculate the parameters of the 3-D line through a least-square minimization along the projection plane of the line (Fig. 4). This also enables us to estimate the line localization uncertainty due to depth estimation.

3.1 The 2-D straight line extraction

A straight line can be obtained from its LSR as in Burns's work [1]. To do that, the planar model of the brightness on the projected edge is assumed. This model is consistent with the brightness

constraint used in direct methods, where a linear variation of the image brightness is considered.

To obtain a line in the image, a planar brightness surface is fitted to the LSR by a least-square approach, predicting the brightness E as a function of image coordinates. In this fitting, a weighting norm $N_w(x, y)$ proportional to the gradient magnitude is considered, so that larger changes in brightness have a greater influence on the adjustment. The minimizing function along the LSR as a function of the brightness surface parameters (A_e, B_e, C_e) is expressed as

$$\sum_{x,y}^{LSR} [A_e\ x + B_e\ y + C_e - E]^2\ N_w(x, y)$$

The straight line is obtained as the intersection of this brightness plane and the horizontal plane of mean brightness E_m in the LSR (weighted with the gradient magnitude)

$$E_m\ = \frac{\sum_{x,y}^{LSR} E\ N_w(x, y)}{\sum_{x,y}^{LSR}\ N_w(x, y)}$$

To define the representation of the projected line we attach a reference system to the projection plane of the line by making two rotations $(Rot(z, \phi_l)Rot(y, \theta_l))$ from the camera reference system. The angle ϕ_l describes the orientation of the line with respect to y axis. As the focal length is the unit, the distance in the image from the origin to the line can be expressed as $tan\theta_l$ (Fig. 4).

Rigid transformations can be interpreted in direct or reverse order, depending on the reference system considered [11]. Thus, these two rotations can be easily interpreted as a rotation ϕ_l along the camera Z axis to place the new Y axis parallel to the 2-D line, followed by a rotation θ_l along this new Y axis to place the X axis normal to the projection plane of the line. We take ϕ_l in the 2π range because the gradient direction is also considered. Thus, the projected line has the equation

$$x\ cos\phi_l + y\ sin\phi_l - tan\theta_l = 0$$

And therefore,

$$\phi_l = atan2(B_e, A_e)$$

$$\theta_l = atan\frac{E_m - C_e}{\sqrt{{A_e}^2 + {B_e}^2}}$$

Thus, the rotation from the camera reference system (C) to the reference system attached to the line projection plane (P) is

$$\mathbf{R}_{CP} = Rot(z, \phi_l)\ Rot(y, \theta_l) = \begin{pmatrix} cos\phi_l cos\theta_l & -sin\phi_l & cos\phi_l sin\theta_l \\ sin\phi_l cos\theta_l & cos\phi_l & sin\phi_l sin\theta_l \\ -sin\theta_l & 0 & cos\theta_l \end{pmatrix}$$

3.2 Depth computation

To obtain the 3-D edge localization, the depth remains to be determined. We compute the depth by fitting a line to a set of points on the line projection plane. This is accomplished by taking the image points in the LSR multiplied by their depth, which is obtained from the brightness information (6). These points, with X_p, Y_p, Z_p coordinates in the reference system attached to the line projection plane, are

$$\begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix} = {\mathbf{R}_{CP}}^T \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \frac{\mathbf{s}_d^T\ \mathbf{R}^T\ \mathbf{t}}{\mathbf{s}_d^T\ \mathbf{R}^T\ \mathbf{p}} \tag{7}$$

We propose a linear regression model to fit Y_p and Z_p, since X_p is the coordinate in the normal direction to the line projection plane. We exclude in the parameterization the line which has the direction of the Z_p axis (this appears in the image as a point). Naming Z_0 and m the parameters that define the line in its projection plane, the line may be fitted according to $Z_p = Z_0 + m\ Y_p$. We refer

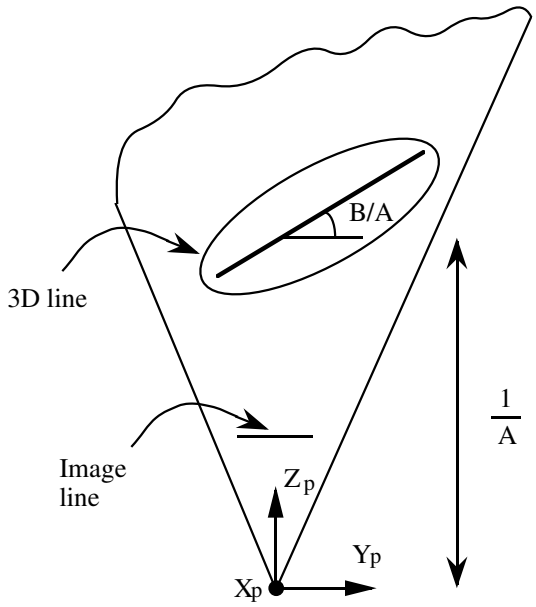


Figure 5: Straight line in its projection plane.

to Z_p and Y_p as the response and prediction variables, respectively. To estimate error statistics, we want the prediction variable to be independent of measurement noise[2]. However, we note in equation (7) that both variables depend on brightness information (expressed in terms of $\mathbf{s}_d$). We employ a transformation to define two new response and prediction variables, where the latter is independent of measurement noise. We transform the previous model in the following way:

$$\frac{1}{Z_p} = \frac{1}{Z_0} - \frac{m}{Z_0}\,\frac{Y_p}{Z_p} = A - B\,\frac{Y_p}{Z_p}$$

The inverse of A is the distance from the focal center to the line in the direction of Z_p, and the slope of the line in the projection plane is $\frac{B}{A}$ (Fig. 5). Denoting unit vectors along camera axes by $\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$, we redefine de prediction variable as

$$\frac{Y_p}{Z_p} = \frac{\hat{\mathbf{y}}^T\,{\mathbf{R}_{CP}}^T\,\mathbf{p}}{\hat{\mathbf{z}}^T\,{\mathbf{R}_{CP}}^T\,\mathbf{p}}$$

and the response variable

$$\frac{1}{Z_p} = \frac{1}{\hat{\mathbf{z}}^T\,{\mathbf{R}_{CP}}^T\,\mathbf{p}}\,\frac{\mathbf{s}_d^T\,\mathbf{R}^T\,\mathbf{p}}{\mathbf{s}_d^T\,\mathbf{R}^T\,\mathbf{t}}$$

Combining the projected line equation and the expressions for Y_p and Z_p of equation (7), the prediction variable turns out to be

$$\frac{Y_p}{Z_p} = \frac{-x\sin\phi_l + y\cos\phi_l}{x\,\cos\phi_l\sin\theta_l + y\,\sin\phi_l\sin\theta_l + \cos\theta_l} = \cos\theta_l(-x\sin\phi_l + y\cos\phi_l)$$

and the response variable,

$$\frac{1}{Z_p} = \cos\theta_l\,\frac{\mathbf{s}_d^T\,\mathbf{R}^T\,\mathbf{p}}{\mathbf{s}_d^T\,\mathbf{R}^T\,\mathbf{t}}$$

As desired, only the response variable is in terms of quantities that are estimated from the spatial and temporal derivatives of image brightness. Therefore, the parameters A and B can be estimated by a least-square approach along the line support region. Here the approximation has been made that pixels of a line support region are coincident to the projected line, while in fact they are close ($x\cos\phi_l + y\sin\phi_l - \tan\theta_l \approx 0$). This approximation works well because the projected line is at the center of the LSR, and the width of the LSR is usually small (three or four pixels).

Thus, the expression minimized along the line support region is

$$J_d = \sum_{x,y}^{LSR} \left[\frac{1}{Z_p} - \left(A - B \, \frac{Y_p}{Z_p} \right) \right]^2 \tag{8}$$

Taking the derivative with respect to A and B and equating to zero, we arrive at a linear set of equations as a function of some integral factors. These factors depend on the brightness information and can be sequentially obtained along the line support region (Appendix 6).

$$\begin{aligned} A &= \overline{\left(\frac{1}{Z_p}\right)} + B\overline{\left(\frac{Y_p}{Z_p}\right)} &&= \frac{S_e}{S} + B\,\frac{S_p}{S} \\ B &= -\frac{\sum_{x,y}^{LSR}\left[\frac{1}{Z_p} - \overline{\left(\frac{1}{Z_p}\right)}\right]\left[\frac{Y_p}{Z_p} - \overline{\left(\frac{Y_p}{Z_p}\right)}\right]}{\sum_{x,y}^{LSR}\left[\frac{Y_p}{Z_p} - \overline{\left(\frac{Y_p}{Z_p}\right)}\right]^2} &&= -\frac{S_{pem}}{S_{p2m}} \end{aligned}$$

From A and B, the 3-D localization of the line in the camera reference system can be recovered. The unit vector in the direction of the 3-D line is

$$\mathbf{R}_{CP} \cdot Rot(x, (atan2(B, A) + \frac{\pi}{2})) \cdot \hat{\mathbf{z}} = \mathbf{R}_{CP} \cdot Rot(x, atan2(A, B)) \cdot \hat{\mathbf{z}}$$

and the distance from the optical center to the 3-D line can be obtained as

$$d = \frac{1}{A}\cos\left(atan2(B, A)\right) = \frac{1}{+\sqrt{A^2 + B^2}}$$

4 Uncertainty estimation

It is well known that in a perception process, it is of the utmost importance to obtain an estimation of the uncertainty. In order to fuse geometric information with uncertainty, we use the *Symmetries and Perturbation Model* [15]. This model associates a reference (F) to every geometric feature. Its location is given by the transformation $\mathbf{T}_{WF}$ relative to a global reference system (W). These transformations are usually represented with a *location vector* $\mathbf{x}_{WF} = (x, y, z, \psi_x, \theta_y, \phi_z)^T$, composed of three cartesian coordinates and three Roll-Pitch-Yaw angles, the order of rigid transformations being

$$\mathbf{T}_{WF} = Trans(x, y, z) \cdot Rot(z, \phi_z) \cdot Rot(y, \theta_y) \cdot Rot(x, \psi_x)$$

The location estimate of a geometric feature is denoted by $\hat{\mathbf{x}}_{WF}$, and the estimation error is represented by a *differential location vector* $\mathbf{d}_F = (dx, dy, dz, d\psi_x, d\theta_y, d\phi_z)^T$ relative to the reference system attached to the feature. Thus, the true location of the feature is

$$\mathbf{x}_{WF} = \hat{\mathbf{x}}_{WF} \oplus \mathbf{d}_F$$

where $\oplus$ represents the composition of location vectors.

Some degrees of freedom of this differential location vector are coincident with the symmetries of the geometric feature and therefore need not be considered as uncertainty elements. For example, the symmetries of a 3-D line are the set of continuous translations along and rotations about the line. Making systematically null the degrees of freedom of $\mathbf{d}_F$ corresponding to the symmetries of each element [15], the uncertainty terms of each feature can be selected in a perturbation vector $\mathbf{e}_F$.

Thus, the location of a geometric element is represented by three elements: $\hat{\mathbf{x}}_{WF}$ (which represents the estimated location of the reference system for the perturbations), $\bar{\mathbf{e}}_F$ (which is the estimated value of the perturbation vector) and $Cov(\mathbf{e}_F)$ (which is its covariance matrix). When $\bar{\mathbf{e}}_F = 0$ we say that the estimation is centered.

This representation model has many advantages to fuse geometric information with uncertainty, because it is general to every geometric feature. Furthermore, the uncertainty representation is not overparametrized, it is independent of the base reference system and it has no singularities.

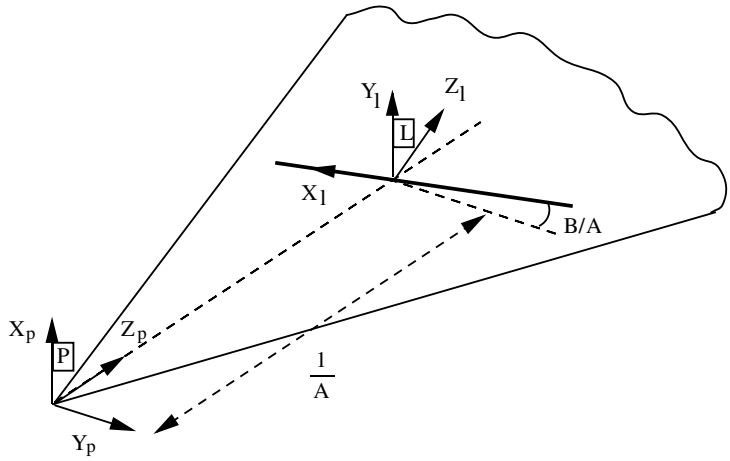


Figure 6: Reference systems attached to the 3-D line (L) and to the projection plane (P) by using the *Symmetries and Perturbation Model*.

The reference system attached to a line has the X axis in the direction of the line and the origin on any of its points. Therefore, the estimated location vector of the line with respect to its projection plane may be (Fig. 6):

$$\hat{\mathbf{x}}_{PL} = \left(0, 0, \frac{1}{A}, 0, \arctan\left(\frac{B}{A}\right), \frac{-\pi}{2}\right)^T$$

Two sources of location error of the 3-D line with respect to the camera reference system can be considered:

- The error of the 2-D line extraction from the image.
- The error in the depth computation by this direct method.

Therefore, the matrix covariance of the differential location vector is

$$Cov(\mathbf{d}_L) = \mathbf{J}_{PL} Cov(\mathbf{d}_P) \mathbf{J}_{PL}^T + Cov(\mathbf{d}_{L_d}) \tag{9}$$

Here, $\mathbf{d}_P$ is the differential location vector of the line projection plane, expressed in its own reference system. $\mathbf{J}_{PL}$ is the Jacobian that transforms location errors between the P and L reference systems [15]. And $\mathbf{d}_{L_d}$ is the depth uncertainty expressed in the reference system attached to the 3-D line.

Due to the symmetries of 3-D lines, two degrees of freedom of the differential location vector are eliminated. They correspond with the translation along and rotation about the line (X axis). Therefore, the perturbation vector of the line $\mathbf{e}_L$ is

$$\mathbf{e}_L = (dy, dz, d\theta_y, d\phi_z)^T$$

We concentrate here on the depth error estimate $Cov(\mathbf{d}_{L_d})$ with the proposed direct method. The other term of line uncertainty, $\mathbf{d}_P$, is obtained from a previous uncertainty model of the projected line[8]. As the computation of the global uncertainty is a basic step to integrate the information, the uncertainty due to the location error of the camera must be added when the feature location is integrated in a static reference system. However, this is outside the scope of the paper.

4.1 Depth error estimate

To obtain a depth error estimate we evaluate the error in the computation of the regression line. An unbiased estimator $\widehat{\sigma^2}$ of the error variance σ^2 can be extracted from the residual (8) according to the following expression:

$$\widehat{\sigma^2} = \frac{J_d}{S-2} = \frac{S_{e2m} - \frac{S_{pem}^2}{S_{p2m}}}{S-2}$$

where S, S_{e2m}, S_{pem} and S_{p2m} are defined in Appendix 6. The factor $S-2$ is introduced because two degrees of freedom are lost in the estimation of depth parameters.

The estimator of the covariance matrix of the depth parameters, $Cov(A,B)$, can be obtained from $\widehat{\sigma^2}$ as

$$Cov(A,B) = \widehat{\sigma^2} \begin{bmatrix} \frac{S_{p2}}{S\ S_{p2m}} & \frac{S_p}{S\ S_{p2m}} \\ \frac{S_p}{S\ S_{p2m}} & \frac{1}{S_{p2m}} \end{bmatrix}$$

On the other hand, this matrix can be related with the error covariance matrix in the reference system attached to the line, by using a linear approximation. In this case we have (see Appendix 6)

$$\begin{aligned} dx &= -\sin(\arctan(\frac{B}{A}))\frac{-1}{A^2}\,dA = \frac{1}{A^2}\,\frac{B}{\sqrt{A^2+B^2}}\,dA \\ dy &= 0 \\ dz &= \cos(\arctan(\frac{B}{A}))\frac{-1}{A^2}\,dA = -\frac{1}{A}\,\frac{1}{\sqrt{A^2+B^2}}\,dA \\ d\psi_x &= 0 \\ d\theta_y &= \frac{-B}{A^2+B^2}\,dA + \frac{A}{A^2+B^2}\,dB \\ d\phi_z &= 0 \end{aligned}$$

Due to the symmetries of the line, the dx error is not relevant. Error in depth exists only along two degrees of freedom $(dz, d\theta_y)$. Thus, the covariance matrix of $(dz, d\theta_y)$ due to depth errors is

$$Cov(dz,d\theta_y) = \begin{bmatrix} -\frac{1}{A}\,\frac{1}{\sqrt{A^2+B^2}} & 0 \\ \frac{-B}{A^2+B^2} & \frac{A}{A^2+B^2} \end{bmatrix} Cov(A,B) \begin{bmatrix} -\frac{1}{A}\,\frac{1}{\sqrt{A^2+B^2}} & 0 \\ \frac{-B}{A^2+B^2} & \frac{A}{A^2+B^2} \end{bmatrix}^T$$

5 Experimental results

Several experiments were carried out with real images to evaluate the proposed method. A camera is attached to a PUMA robot, that provides the motion data to the algorithm. Several masks to extract the gradient were tested. The best results were obtained with simple masks after the application of a gaussian filter to smooth the images. We observed that the method works well when the disparity in the image is small, typically less than two pixels.

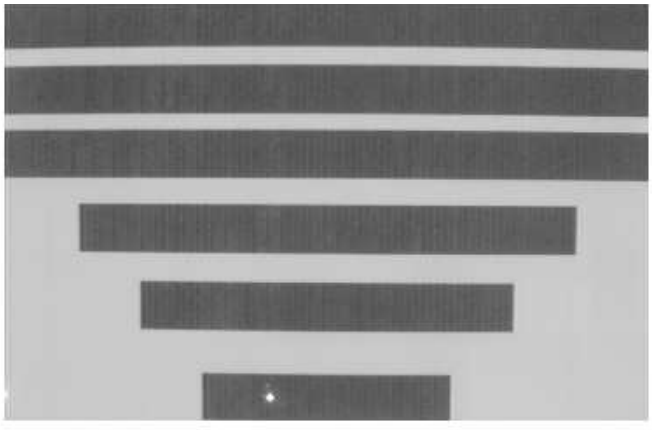


Figure 7: Image of the test scene. Real images of simple scenes make it easier to evaluate the goodness of vision methods. The second image is quite similar.

Small translations and rotations of the camera have been carried out to test the validity of the method. In the first experiments, simple scenes on a work table were used to easily check the computed

Table 1: Orientation errors $d\theta_y$ and position errors dz, comparing the measured values with the expected depth (0 degrees in orientation and 45 focal units in position) corresponding to the lines in Fig. 7. Uncertainty estimate of orientation $\sigma_{d\theta_y}$ and uncertainty estimate of position σ_{dz}.

	Estimated errors			
	$d\theta_y$ (°)	dz (f)	$\sigma_{d\theta_y}$ (°)	σ_{dz} (f)
1	0.26	6.94	17.18	7.60
2	0.47	8.01	21.27	9.72
3	0.82	7.92	15.20	7.89
4	6.48	9.63	16.84	8.37
5	2.40	8.42	14.51	7.34
6	1.80	8.15	16.08	7.35
7	8.04	10.13	14.32	6.67
8	1.40	10.35	21.02	7.80
9	10.31	11.83	20.30	8.23
10	13.17	11.16	28.17	10.31

depth. We used a planar pattern with several lines of equal aspect but different lengths (Fig. 7). The pattern is located parallel to the image plane, at a distance of approximately 45 focal units from the camera (camera has a focal length of 12 *mm.*). The camera motion is a translation of 1 *mm.* along an axis parallel to the image plane and perpendicular to the lines.

In Table 1 we present an example of depth results (line numbers correspond to horizontal lines from top to bottom). The first column presents the error in orientation ($d\theta_y$), and the second column has the error in position (dz), comparing the measured values with the expected depth. The third and fourth columns present the square root of diagonal elements of the matrix $Cov(dz, d\theta_y)$. The error in position has turned out to be coherent for all lines and experiments. The values are comparable for all lines, being greater in shorter lines. The obtained uncertainties in position are well adjusted to the differences between the expected and measured depth. The estimation of the orientation is good, but the uncertainty is large. Therefore, the 3-D orientation turns out to be less reliable than the 3-D position.

In this test the position errors are about 20% of the depth, which is reasonable according to the conditions of the experiment. To evaluate these results we compare them with theoretical results in an ideal stereo system (geometrically equivalent) whose only source of error is the location of the lines in the image. Thus, our errors are similar to those of this equivalent stereo system in which the lines were extracted in the images with a location error of about 0.2 pixels.

Image 1 Image 2

Figure 8: Two consecutive images of the laboratory scene. The disparity of about 0.7 pixels turns out unappreciable on sight.

Experiments with complex scenes of the laboratory have also been performed (see Fig. 8). Two types of motion were used. In both cases, the estimated covariances of the error in orientation are high because the lines are far from the camera and many of them are short. Consequently the 3-D orientation of these lines cannot be considered valid. However, the 3-D position of high-contrast lines is reasonably accurate. The first motion is a translation parallel to the image plane having a disparity of about one pixel between images. Selecting the largest lines, the errors in position are

Table 2: Mean and standard deviation of the depth of ten lines (Fig. 7) at different tests. The expected depth is 45 focal units for all lines. The method has a bias, but estimates correctly that all the lines have the same depth.

	Test 1	Test 2	Test 3	Test 4
$Mean$	51.23	47.93	63.54	41.26
$S.D.$	1.81	1.86	2.45	1.62

about 30%, except for lines nearly parallel to the projected motion. The depth of these lines cannot be obtained due to the aperture problem. The second motion is a pure translation parallel to the focal axis. The maximum disparity is about 0.7 pixels. In this case, depth results are worse than in the previous situation, but large lines far from the image center provide qualitatively valid estimations of 3-D position. The estimated depth is erroneous near the focus of expansion (in this case, the image center), because the image disparity due to translation is negligible, and small unknown rotations can introduce large estimation errors.

An experiment with a large motion has been carried out. Using the pattern in Fig. 7, we have made a translation of 120 *mm.* parallel to the image plane and a rotation of 12.73° about the other axis parallel to the image plane to compensate the disparity of the central line. This motion can be considered as a manual version of a control procedure that fixates a line in the image. In this case the depth errors of the central line are very small, about 1% of the depth, because the 3-D triangulation is much better conditioned. The 3-D position has an error of 0.39 focal units, and the 3-D orientation has an error of 0.73 degrees. The square root of diagonal elements of the matrix $Cov(dz, d\theta_y)$ are $0.28\,f$ and 1.51° respectively. Unfortunately, when performing large translations, the method can be applied only near the fixated region because the temporal correspondence is lost in other pixels.

5.1 Influence of motion errors

Two sources of motion errors can be considered in our experimental environment; the inherent inaccuracy of the robot arm and the location uncertainty of the camera in the robot hand (obtained from the calibration of the camera). The proposed method works well with small disparities and it assumes the motion to be known. In a general situation, a small motion is required to have small disparities, but small motion can be overridden by mechanical errors or calibration errors. We have used an angular robot to move the camera, which rotates (with angles that are too small to be corrected by the robot controller), although the commanded motion is a pure translation. Even so, the method works well and the results can be compared to those obtained with other approaches. We cannot make more precise motions, but we believe that the technique would work best in applications were the camera can be moved with high precision.

To evaluate the effect of motion errors we have used the planar pattern scene (Fig. 7). Depth information has been obtained repeating the motion over the same scene. To simplify the exposition, we concentrate here in the position, without considering the orientation. The method estimates correctly that all the lines have the same depth, but each time, a different bias is obtained for all lines. Table 2 shows the mean and standard deviation of the depth of the ten lines in Fig 7, obtained from four different motions. This bias is probably due to errors in camera motion (0.03° of camera rotation error could originate this bias).

5.2 Analysis of performance

The results confirm that the proposed method is valid to compute depth on lines. To apply the method, the illumination and reflection conditions must be stable. The straight edges must be steep: a minimum spatial gradient of about 12 gray level units per pixel ($\frac{glu}{pix}$) is recommended. Assuming typical errors of 2 $\frac{glu}{pix}$ and 3 $\frac{glu}{frame}$ in the computation of spatial and temporal gradients respectively, a spatial gradient of about 12 $\frac{glu}{pix}$ is required to have an accuracy of 0.3 pixels in the projected motion.

The 3-D position is estimated better than the 3-D orientation, and large lines provide better results than short ones. Due to the aperture problem, the results are erroneous when the translation is parallel to the projection plane of the line.

The method needs small disparity (typically less than two pixels) and therefore, in general situations, small motions must be carried out. Small motions have the inconvenient that make ill-conditioned the 3-D triangulation, and therefore the estimated depth is very sensitive to camera motion errors. However, our method can also be applied with large motions and a fixation procedure. In this way, accurate depth can be obtained in the fixated region.

Currently, we have obtained errors that are similar to those of an ideal stereo system (geometrically equivalent) in which the lines were extracted in the images with errors of about 0.2 pixels. These results can be considered good, since most of the feature extractors do not have subpixel accuracy. Thus, our method can be used as an alternative to classical correspondence-based approaches when the 3-D triangulation is ill-conditioned and the accuracy of the feature extractor is not good enough to obtain depth information. Besides that, the computational cost of the proposed method is smaller (about a half in our system) than an equivalent system based on line correspondences because both the extraction of lines in the second image and the matching are avoided.

6 Conclusions

We have presented an algorithm to extract the 3-D location of straight edges by a direct method, starting from two images taken by a mobile camera. The proposed method uses a formulation that directly combines the brightness continuity assumption and the displacement of the projected line.

Compared with classical differential approaches, our method uses topological information about straightness and obtains the depth globally. This topology (which relates edge elements into lines) can be easily obtained, and captures the more relevant information, in images of man made environments. Comparing with an equivalent system based on corresponding lines, our method takes a shorter computational time. Besides that, spurious data due to the extraction and the matching steps are not present in our method, but small disparities and stable illumination are needed.

As small disparities are needed, small motions are normally required and therefore depth results are not seemingly good, but alternative methods do not provide better results in these conditions. Several experiments have been carried out to show that the method performs well, specially when large motions are used and a method to fixate a region in the image is available.

The analysis of the results showed that our method can be used as an alternative to classical correspondence-based approaches when the 3-D triangulation is ill-conditioned and the accuracy of the feature extractor is not good enough to obtain depth information.

Acknowledgments

This work was partially supported by projects TAP94-0390 and TAP97-0992-C02-01 of the Comisión Interministerial de Ciencia y Tecnología (CICYT). Thanks to the anonymous reviewers whose comments significantly improved the manuscript.

Appendix A

The integral factors obtained from the brightness in the images are

$$
\begin{aligned}
S_p &= \cos\theta_l \sum_{x,y}^{LSR} (-x\sin\phi_l + y\cos\phi_l) \\
S_e &= \cos\theta_l \sum_{x,y}^{LSR} \frac{\mathbf{s}_d^T \mathbf{R}^T \mathbf{p}}{\mathbf{s}_d^T \mathbf{R}^T \mathbf{t}} \\
S_{p2} &= \cos^2\theta_l \sum_{x,y}^{LSR} (-x\sin\phi_l + y\cos\phi_l)^2 \\
S_{pem} &= \cos^2\theta_l \sum_{x,y}^{LSR} \left(-x\sin\phi_l + y\cos\phi_l - \frac{S_p}{S\cos\theta_l}\right)\left(\frac{\mathbf{s}_d^T \mathbf{R}^T \mathbf{p}}{\mathbf{s}_d^T \mathbf{R}^T \mathbf{t}} - \frac{S_e}{S\cos\theta_l}\right)
\end{aligned}
$$

$$S_{p2m} = \cos^2\theta_l \sum_{x,y}^{LSR} \left(-x\sin\phi_l + y\cos\phi_l - \frac{S_p}{S\cos\theta_l}\right)^2$$

$$S_{e2m} = \cos^2\theta_l \sum_{x,y}^{LSR} \left(\frac{\mathbf{s}_d^T\,\mathbf{R}^T\,\mathbf{p}}{\mathbf{s}_d^T\,\mathbf{R}^T\,\mathbf{t}} - \frac{S_e}{S\cos\theta_l}\right)^2$$

where S is the number of pixels in the LSR.

Appendix B

By using the *Symmetries and Perturbation Model*, the location vector of the line in the plane reference system is

$$\mathbf{x}_{PL} = \hat{\mathbf{x}}_{PL} \oplus \mathbf{d}_L \simeq \hat{\mathbf{x}}_{PL} + \mathbf{J}_{2\oplus[\hat{\mathbf{x}}_{PL},0]}\,\mathbf{d}_L$$

being $\hat{\mathbf{x}}_{PL} = (0, 0, \frac{1}{A}, 0, \arctan(\frac{B}{A}), \frac{-\pi}{2})^T$ and $\mathbf{J}_{2\oplus[\hat{\mathbf{x}}_{PL},0]}$ the Jacobian of the composition of location vectors with respect to the second one [15].

Besides that, the line location, expressed in the reference of the plane can be written as

$$\mathbf{x}_{PL} = f(A,B) \simeq \hat{\mathbf{x}}_{PL} + \frac{\delta\mathbf{x}_{PL}}{\delta(A,B)} \begin{bmatrix} dA \\ dB \end{bmatrix}$$

Therefore, with this two expressions we arrive at

$$\mathbf{d}_L \simeq \mathbf{J}_{2\oplus[\hat{\mathbf{x}}_{PL},0]}{}^{-1}\,\frac{\delta\mathbf{x}_{PL}}{\delta(A,B)} \begin{bmatrix} dA \\ dB \end{bmatrix} = \mathbf{J} \begin{bmatrix} dA \\ dB \end{bmatrix}$$

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] N. Draper and H. Smith. *Applied Regression Analysis.* Wiley, New York, 1981.

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

[4] J.J. Guerrero. *Percepción de Movimiento y Estructura con Visión basada en Contornos Rectos.* Dpto. de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, Zaragoza, Mayo 1996.

[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] J.M. Martínez and L. Montano. The effect of the image imperfections of a segment on its orientation uncertainty. In *International Conference on Advanced Robotics*, Barcelona, September 1995.

[9] S. Negahdaripour and B.K.P. Horn. Direct passive navigation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(1):168–176, 1987.

[10] S. Negahdaripour and J. Lanjing. Direct recovery of motion and range from images of scenes with time-varying illumination. In *International Symposium on Computer Vision*, pages 467–472, Coral-Gables, Florida, Nov. 1995.

[11] R.P. Paul. *Robot Manipulators: Mathematics, Programming, and Control.* MIT Press, Cambridge, Mass., 1981.

[12] D. Raviv. A quantitative approach to camera fixation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 386–392, 1991.

[13] 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.

[14] M.A. Taalebinezhaad. Direct recovery of motion and shape in the general case by fixation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14:847–853, 1992.

[15] J.D. Tardós. Representing partial and uncertain sensorial information using the theory of symmetries. In *IEEE International Conference on Robotics and Automation*, pages 1799–1804, Nice, France, May 1992.

[16] 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.