

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

Internal Report: 1999-V02
**Camera Motion from Brightness on Lines. Combination of
Features and Normal Flow¹**

Guerrero J.J., Sagüés C.

If you want to cite this report, please use the following reference instead:
Camera Motion from Brightness on Lines. Combination of Features and Normal Flow,
Guerrero J.J., Sagüés C., *Pattern Recognition*, Vol. 32(2), pages 203-216, 1999.

¹This work was partially supported by projects TAP94-0390 and TAP97-0992-C02-01 of the Comisión Interministerial de Ciencia y Tecnología (CICYT).

Camera Motion from Brightness on Lines. Combination of Features and Normal Flow

J.J. Guerrero & C. Sagüés

Dpto. 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

In this paper feature extraction (straight lines) and optical flow techniques are combined to obtain camera motion. The expressions relating the brightness constraint to the 3D localization and motion of a line and its projected motion are established. These expressions allow to obtain the 3D motion of a camera using brightness information in image regions supporting straight lines of known 3D localization. The algorithm needs neither correspondence computation nor full optical flow computation and uses a direct formulation. It has been compared experimentally with a geometrically equivalent approach based on corresponding features, showing better results for close images. The proposed method can be combined with a feature-based approach, improving classical methods when obtaining 3D information using triangulation is ill-conditioned.

Keywords: Dynamic vision, motion from structure, straight edges, brightness constraint, direct motion computation.

1 Introduction

Methods for extracting shape and motion information from vision can be classified as correspondence-based approaches and optical flow methods [1]. Usually these methods have been considered in a separate way, without establishing connections between them.

The use of geometric features provides a good way to efficiently select, concentrate and manipulate vision information. In particular, features like straight edges are easy to extract and match [2], being robust to partial occlusions. Besides, they involve an implicit perceptual grouping that includes robust topological information, specially in man-made environments. However, the geometric representation of vision information turns out many times to be too simplified (Fig. 1). Even for a human, it is necessary to view the gray level image to recognize a scene, because image edges are not enough [3]. In methods based on features the image intensity information is not used after the features have been extracted and matched, and therefore useful information is discarded.

Methods based on corresponding features allow higher disparities between images than optical flow methods, and therefore motion and structure problems turn out to be better conditioned. It has been said [4] that image variations are better obtained using corresponding features than using the image brightness constraint. This is correct when image disparity is high and the brightness constraint and its gradient-based approach are not applicable. However, with close images, there are serious triangulation problems when obtaining 3D information, which can be partly circumvented using all available information.

On the other hand, methods based on optical flow assume that image brightness does not vary in time [5, 6]. These approaches work well with small disparities having a small computational cost per projected motion measurement [7]. They can be applied uniformly across the image, but when spatial or temporal gradients are small, the results are very sensitive to noise [8].

As it is proposed in this paper, we consider it is useful to combine the geometric description of the scene with information about image brightness. We use straight edges extracted with our version of

the method proposed by Burns [9]. This extractor provides not only the geometric representation of the image edge, but also a segmentation of the image into line support regions. These line support regions allow to combine methods based on geometric features with methods based on brightness information (Fig. 1). To do that, the kinematics of a straight edge in relation to its location in the image and in the scene, are obtained and related to the brightness constraint. Part of the scheme is based on the motion field of lines obtained by Faugeras and his colleagues [10, 11]. However, while they consider an indirect approach of the flow based on corresponding features, we have experimentally shown (§5.1) that, with close images and stable illumination conditions, the proposed method works better than an equivalent solution based on corresponding features.

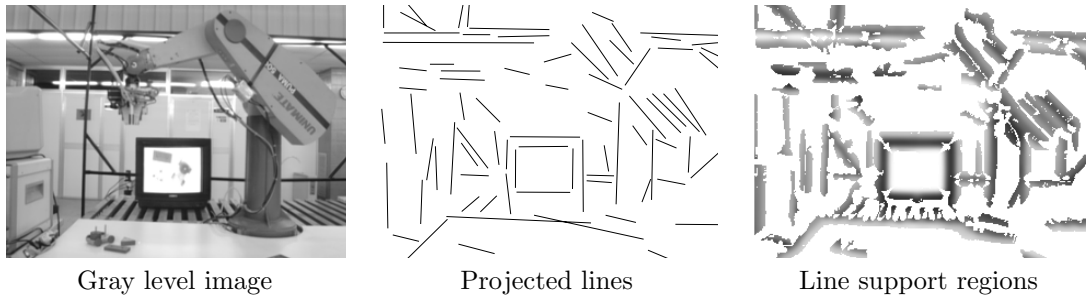


Figure 1: Classical perception methods can be improved using both the geometric representation of the straight edges and the intensity in their support regions. In this way, the potentiality of geometric features is maintained and all the information of the image edges is considered.

The proposed combination of features and flow techniques is aimed at obtaining the camera motion [12]. It is known that motion and structure cannot be obtained from lines in two images, even if many features are available [13]. When the rotation is known, the direction of translation can be obtained from lines [14, 15, 16]. But in a general case, more views or additional depth information are needed to obtain camera motion. In our work we do not simultaneously compute structure and motion. We consider the brightness constraint along the line support region, using its rectilinear topology and depth information to compute the camera motion.

A summary of the paper follows. After this introduction we present in §2 some preliminaries related to the models, representations and assumptions made in our work. The kinematics of a straight line in relation to its location in the scene and its projection in the image are obtained and related to the brightness constraint in §3. After that, we present in §4 the algorithm used to obtain the 3D camera motion from at least three lines whose 3D localization is known. At least two close images must be used, and the motion is computed directly from the spatial and temporal brightness gradient. Experimental results with real images, including a comparison with other techniques, are presented in §5. Finally, in §6 conclusions are presented.

2 Preliminaries

We first recall some details about the camera model, the extraction of straight lines and their support regions, the representations of both projected lines and 3D lines, and the brightness change constraint.

2.1 Camera and motion models

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

$$x = \frac{X}{Z} \quad , \quad y = \frac{Y}{Z}. \quad (1)$$

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

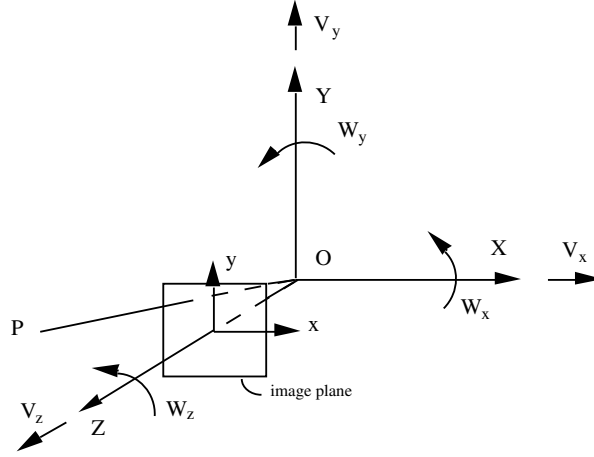


Figure 2: Pinhole camera model

2.2 Extraction of straight lines and their support regions

As mentioned above, our approach takes straight edges in the image as key features. They have been extracted with our version of the method proposed by Burns [9]. This extractor provides not only the geometric representation of the projected line, but also a segmentation of the image into line support regions.

The first step in the image segmentation algorithm is the extraction of spatial gradients. Afterwards, pixels having a gradient magnitude larger than a threshold are grouped into regions of similar direction of brightness gradient. Segmentation is globally achieved using fixed partitions of gradient orientation. Two overlapping sets of partitions are used in order to avoid problems related to the arbitrary boundary of fixed partitions (Fig. 3). From both partitions, 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 a straight edge) contains all available information in the image about the straight edges.

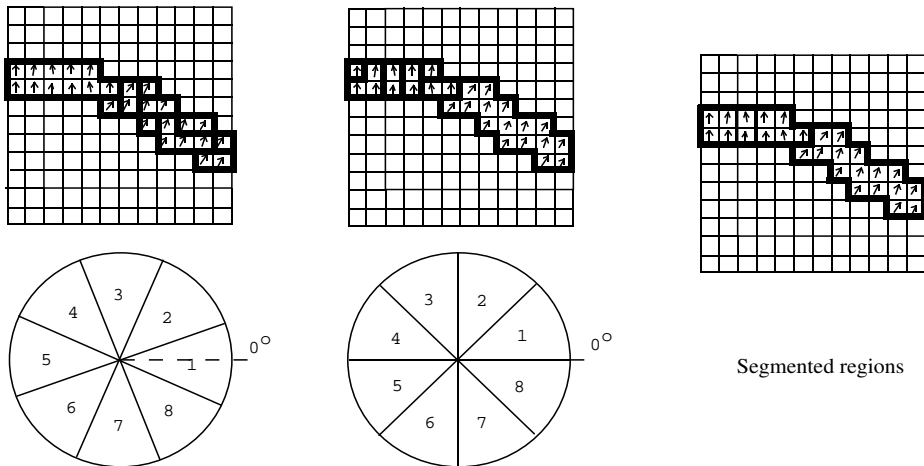


Figure 3: Image segmentation into LSR by using two fixed and overlapped partitions of the gradient orientation.

A straight line can be obtained from each LSR. We assume the planar model of the brightness on the projected edge. This model is consistent with the brightness constraint used in our method, where a linear variation of the image brightness is considered. To locate the projected line, a planar brightness surface is fitted to the LSR by a least-squares approach, predicting the brightness E as a function of image coordinates. In this fitting, a weighting norm proportional to the gradient magnitude

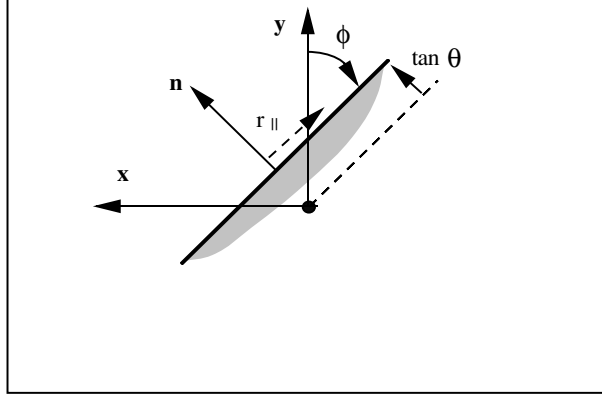


Figure 4: Projected line representation

is considered, so that larger changes in brightness have a greater influence on the result. The straight line is obtained as the intersection of this brightness plane and the horizontal plane of mean brightness E_m in the LSR.

2.3 Representation of projected lines and 3D lines

Two parameters are needed to locate the projection of a line in the image plane, and at least four parameters are needed to locate a 3D line. In this work, we use two parameters of its projection in the image and two additional depth parameters to represent both the projected line and the 3D line.

To define the representation of the projected line we attach a reference system to the projection plane of the line by making two rotations ($Rot(z, \phi)Rot(y, \theta)$) from the camera reference system. The angle ϕ describes the orientation of the line with respect to the y axis. As the focal length is the unit, the distance in the image from the origin to the line can be expressed as $\tan\theta$ (Fig. 4). The x axis of the new reference system will be perpendicular to the projection plane of the line. If we name \mathbf{n} the vector of this reference system in the x direction, the image line has the equation

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

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

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

To obtain a representation of the 3D line, the two parameters named above are combined with two additional parameters. Thus, we can define a third rotation $Rot(x, \psi)$ ($0 \leq \psi < \pi$) such that the new z axis (named \mathbf{a}) points in the direction of the line and the new y axis (named \mathbf{o}) is perpendicular to the 3D line (it points from the camera to the scene). The fourth parameter (d) is defined as the distance from the camera reference system to the 3D line (Fig. 5). This parameter is always greater than zero because the line is always in front of the camera.

Thus, the transformation that moves the camera reference system to a reference system attached to the 3D line (it is located in the point of the line closest to the focal center, having its z axis parallel to the 3D line and having its x axis normal to its plane of projection), is $\mathbf{T}_{cl} = Rot(z, \phi)Rot(y, \theta)Rot(x, \psi)Trasl(y, d)$. And expressed as a homogeneous matrix is

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

where c is \cos and s is \sin . This transformation also allows to transform points from the line reference system to the camera reference system.

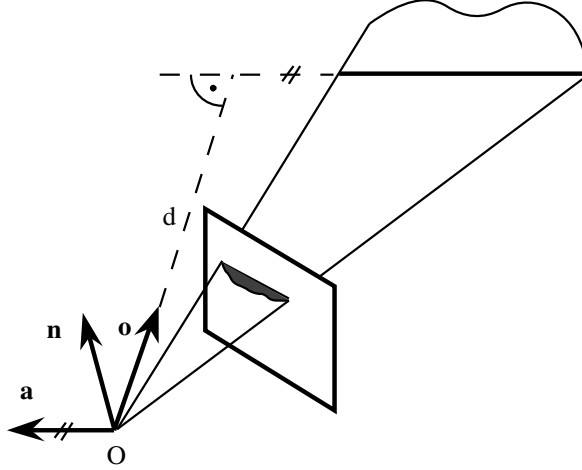


Figure 5: Representation of the 3D line using its plane of projection.

Some of the advantages of our representation are:

- Two out of the four parameters defining the 3D line are also used to define its projection in the image.
- We do not have singularities for lines which can appear in the image, because $|\theta|$ is always less than $\frac{\pi}{2}$, that is the sole singularity of the angles used to express the rotations (this singularity would correspond with projection planes parallel to the image plane, which cannot be observed).
- The cyclic symmetries of line and plane representations are avoided using the contrast sign of the edge and assuming that the line is always in front of the camera.

2.4 Brightness change constraint

The brightness continuity assumption ($\frac{dE}{dt} = 0$) has been considered as a good constraint to obtain motion [5]. It assumes that the irradiance in the image, received from a point in the scene, does not vary with time. This constraint has been questioned and avoided by some researches [4], but many others [5, 1, 17] base their work on it. Some authors [18] consider relaxing the continuity assumption to deal with brightness variations due to non motion events. 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 [19]. Many authors conclude that the computation of 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.

The brightness constraint equation can be formulated as

$$\frac{\delta E}{\delta x} \frac{dx}{dt} + \frac{\delta E}{\delta y} \frac{dy}{dt} + \frac{\delta E}{\delta t} = E_x u + E_y v + E_t = 0 \quad (3)$$

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

Using this expression, the visual motion is not fully defined at each pixel since only the flow that is normal to the contours of iso-brightness can be recovered. In order to solve this problem (known as the aperture problem), researchers have tried to assume a motion field, close to the real one when possible, by imposing smoothness constraints [20]. Normally these smoothness constraints do not work well near edges, where the brightness change equation could be used in practice. Therefore, in order to have a robust method of motion computation, it is suitable to use only the normal flow, avoiding more general but ill-conditioned problems [17]. In our approach, this problem is circumvented because visual motion is combined with depth information. Thus, we adopt the brightness change constraint equation in regions supporting steep edges, using topological information related to their straightness and their depth.

3 Motion of straight edges

As mentioned above we have obtained the relations between the image brightness along a straight edge, the localization of the line in the space and the 3D camera motion. The goal is to have expressions which take the available topological information and directly combine the image brightness and the motion parameters to be estimated.

3.1 Normal flow of an edge and a projected line

It is known that the brightness constraint (3) can be written as $(E_x, E_y) \cdot (u, v)^T = -E_t$. Therefore, the optical flow in the direction of the brightness gradient $(E_x, E_y)^T$, or normal flow u_n , is

$$u_n = \frac{-E_t}{\sqrt{E_x^2 + E_y^2}} \quad (4)$$

This is the topological level often used in optical flow methods. However, in this way the gradient orientation is obtained locally and it is not robust enough. Experimentally we have observed deviations greater than 10° in the locally obtained gradient direction. Other authors [21] have shown deviations about $\pm 15^\circ$ in the local computation of the gradient orientation. Besides that, the topology that relates edge elements into lines is not taken into account at this level.

We propose to extend this constraint to image regions corresponding to lines, combining flow methods and geometric features. Association of edge elements into features like lines, turns out very useful and easy to obtain on images taken in man-made environments.

With the representation proposed before, the normal flow of a generic point (x, y) on the image line can be written as $u_n(x, y) = u(x, y) \cos \phi + v(x, y) \sin \phi$. The image line equation is $x \cos \phi + y \sin \phi - \tan \theta = 0$. Taking the derivative of this equation, we have

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

that is

$$u_n(x, y) = (1 + \tan^2 \theta) \dot{\theta} - (y \cos \phi - x \sin \phi) \dot{\phi} = \frac{1}{\cos^2 \theta} \dot{\theta} - r_{\parallel} \dot{\phi} \quad (5)$$

where r_{\parallel} is the distance from the generic point (x, y) in the line, to the point in the line closest to the image center (Fig. 4).

Combining these two expressions (4) (5), we obtain the relation between brightness information and the kinematics of a projected line.

$$\frac{-E_t}{\sqrt{E_x^2 + E_y^2}} = \frac{1}{\cos^2 \theta} \dot{\theta} - r_{\parallel} \dot{\phi} \quad (6)$$

In this way the rectilinear topology of the contour is taken into account, using the localization of the projected line, that has been obtained in a global way. It is interesting to see that using topology, the variation of line orientation $\dot{\phi}$ can be obtained from first order derivatives of brightness. This information can only be obtained locally from second order brightness variations, which are sensitive to noise [11].

3.2 Kinematics of the projected line and the 3D line

The relationship that exists between the 3D structure and the kinematics of a line in space, with the two dimensional structure and kinematics of its projection in the image is well known [10]. This relation can be summarized in the *line motion field equation*. When the camera moves rigidly in space (\mathbf{w}, \mathbf{t}) and considering our line representation, this equation can be expressed as follows:

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

where \times is the cross product between vectors and \cdot is the dot product.

Equation (7) gives the information that can be obtained from a 3D line, its projection and its motion field in the image. From the projected motion of a line it is only possible to recover the rotation about one direction and the translation along another direction. The rotation cannot be obtained about the direction of the 3D line (symmetry of the feature), nor about the perpendicular to its projection (symmetry of the line observation). The translation cannot be obtained along the direction of the 3D line (symmetry of the line), nor along the direction of its perpendicular which is contained in the plane of projection (it does not modify the projected line).

Making the dot product of vector $\dot{\mathbf{n}}$ with vector \mathbf{a} , 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 define two 3D motion parameters for each line. We name these parameters for the l -th line as w_{ol} and t_{nl} ,

$$w_{ol} = \dot{\mathbf{n}} \cdot \mathbf{a} = (-\mathbf{w} \times \mathbf{n}) \cdot \mathbf{a} + \frac{\mathbf{t} \cdot \mathbf{n}}{d} \mathbf{o} \cdot \mathbf{a} = \mathbf{w} \cdot \mathbf{o} \quad (8)$$

$$t_{nl} = \dot{\mathbf{n}} \cdot \mathbf{o} = (-\mathbf{w} \times \mathbf{n}) \cdot \mathbf{o} + \frac{\mathbf{t} \cdot \mathbf{n}}{d} \mathbf{o} \cdot \mathbf{o} = -\mathbf{w} \cdot \mathbf{a} + \frac{\mathbf{t} \cdot \mathbf{n}}{d}. \quad (9)$$

Information about the 3D orientation of the lines is frequently available, for example from vanishing point detection [22] or using the vertical cue [23]. Therefore, it is interesting to isolate the motion information that could be obtained knowing only the 3D orientation of the line. Using the proposed parameterization of the line kinematics, we can consider the 3D line orientation without knowledge of its 3D position. As can be seen below, these two motion parameters (w_{ol}, t_{nl}) can be obtained from the brightness information on a line, knowing its 3D orientation.

Thus, from the flow of a projected line and its 3D orientation we can recover \mathbf{w} in the direction of the perpendicular from the origin to the line (\mathbf{o}). This can be recovered knowing neither line depth nor translational motion (equation (8)). Moreover, the translational motion in the direction of the normal to the plane of projection (\mathbf{n}) is coupled with the rotation in the camera reference system around an axis parallel to the 3D line (equation(9)).

On the other hand, we can compute the line motion field as Gai-Checa [11]:

$$\dot{\mathbf{n}} = \begin{bmatrix} -\cos \theta \sin \phi \dot{\phi} - \cos \phi \sin \theta \dot{\theta} \\ \cos \theta \cos \phi \dot{\phi} - \sin \phi \sin \theta \dot{\theta} \\ -\cos \theta \dot{\theta} \end{bmatrix} = \dot{\phi} \cos \theta \mathbf{l} - \dot{\theta} \mathbf{n} \times \mathbf{l} \quad (10)$$

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

Introducing the 3D orientation of the line (represented by ψ), we can deduce that $\mathbf{l} \cdot \mathbf{a} = -\sin \psi$ and $\mathbf{l} \cdot (\mathbf{a} \times \mathbf{n}) = \cos \psi$. Therefore, after some algebraic manipulations, we arrive at

$$\begin{aligned} \dot{\mathbf{n}} \cdot \mathbf{a} &= \cos \theta \dot{\phi} \mathbf{l} \cdot \mathbf{a} - \dot{\theta} (\mathbf{n} \times \mathbf{l}) \cdot \mathbf{a} = \\ &= -\cos \theta \dot{\phi} \sin \psi - \dot{\theta} \cos \psi \end{aligned} \quad (11)$$

$$\begin{aligned} \dot{\mathbf{n}} \cdot \mathbf{o} &= \cos \theta \dot{\phi} \mathbf{l} \cdot (\mathbf{a} \times \mathbf{n}) - \dot{\theta} (\mathbf{n} \times \mathbf{l}) \cdot (\mathbf{a} \times \mathbf{n}) = \\ &= \cos \theta \dot{\phi} \cos \psi - \dot{\theta} \sin \psi. \end{aligned} \quad (12)$$

From equations (11) and (12), we can compute both $\dot{\phi}$ and $\dot{\theta}$ as

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

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

Substituting these expressions in (5), we obtain

$$u_n \cos \theta + t_{nl} \left(\frac{\sin \psi}{\cos \theta} + \cos \psi r_{\parallel} \right) + w_{ol} \left(\frac{\cos \psi}{\cos \theta} - \sin \psi r_{\parallel} \right) = 0 \quad (13)$$

Taking the brightness constraint (4) and substituting in (13), we obtain an equation involving the 3D orientation of the line, with motion parameters associated to each line (t_{nl}, w_{ol}) and with some

measurable quantities related to its brightness gradient. Thus, we have the following equation for each pixel (x, y) of the line in the image:

$$\frac{E_t}{\sqrt{E_x^2 + E_y^2}} \cos \theta = t_{nl} \left(\frac{\sin \psi}{\cos \theta} + \cos \psi r_{\parallel} \right) + w_{ol} \left(\frac{\cos \psi}{\cos \theta} - \sin \psi r_{\parallel} \right) \quad (14)$$

This is our fundamental equation which will be used to compute the motion from brightness information on line support regions.

4 Camera motion from brightness on lines

With the expression deduced above, we can estimate the camera motion directly from brightness on straight edges whose 3D localization is known. To do that, two close images are needed. Line support regions (LSR) and the straight edges are extracted from the first image using the method presented above (§2.2). The temporal gradient, that is necessary to extract motion information, can be obtained using also the second image.

From the normal flow of a line, two equations are given. In a general case motion and structure cannot be simultaneously obtained even if many lines are available. If we know the 3D orientation of the lines, we can obtain the camera rotation. Besides that, when the line position is known, the camera translation can also be estimated. In both situations three straight edges, at least, are needed.

4.1 Separate computation of rotation and translation

From the support region of a line in the image (ϕ, θ) , whose orientation in space is known (ψ) , we have one expression (14) that is a function of the two motion parameters of the line (t_{nl}, w_{ol}) . Using a linear least-squares approach, the two motion parameters of each line can be obtained.

Using expression (14), the measurement obtained from the image corresponds with the normal flow of each point into the LSR, and therefore the error to minimize has a physical interpretation. It is known that computing brightness constraint at high-gradient points increases the likelihood of optical flow and image motion being equivalent. Correspondingly, we weigh the brightness gradient magnitude in such a way that better pixels and pixels centered in the LSR contribute more. The formulation used to obtain t_{nl} and w_{ol} can be seen in Appendix A.

Each line (l) and its motion parameters provides a linear equation in terms of camera rotation velocity, that is

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

From the motion parameters of three straight edges, the rotation velocity of the camera is computed by solving a set of linear equations. When the vectors \mathbf{o}_l are coplanar, the equation set is undetermined. In a similar way to the problem of obtaining camera orientation from 2D-3D line correspondences, rotation velocity \mathbf{w} can be obtained, except one of the following situations occurs [24]:

- The three lines are parallel in 3D.
- The three projected lines are collinear in the image.
- Two lines are parallel in 3D, being their projection collinear in the image.
- Two lines are parallel in 3D and perpendicular to the third one, being their planes of projection perpendicular to the plane of projection of the third line.
- Two lines are perpendicular in 3D to the third one, being their planes of projection both coincident and perpendicular to the plane of projection of the third line.

When the 3D position of the lines (d_l) is also known, the camera translation can also be obtained by using three lines. With the previous estimate, the contribution of the rotation to parameter t_{nl} can be extracted, and therefore each line provides an equation to obtain camera translation, that is

$$\mathbf{t} \cdot \mathbf{n}_l = (t_{nl} + \mathbf{w} \cdot \mathbf{a}_l) d_l.$$

This is made solving a set of linear equations. In addition to the requirements for obtaining rotation, none of the three lines can be collinear to each other. Neither can the three lines be parallel in the image, because all the normal vectors \mathbf{n}_l would be coplanar.

When more than three lines are available, a least-squares approach enables a more robust computation of the camera motion. Additionally, an estimation of the uncertainty of motion can be obtained. The formulation used can be seen in Appendix B.

4.2 Direct motion estimation

The previous solution separates motion computation obtained from line position and orientation in the scene, thus it is possible to obtain the camera rotation without knowing the depth of the lines. However, when the total localization of the lines is known, we can estimate the camera motion directly from the image brightness.

To do that we take the expression (14). Introducing also the 3D position of the line (d), we can substitute (8) and (9) in (14), and we have

$$\begin{aligned} \frac{E_t}{\sqrt{E_x^2 + E_y^2}} \cos\theta &= \frac{\mathbf{t} \cdot \mathbf{n}}{d} \left(\frac{\sin\psi}{\cos\theta} + \cos\psi r_{\parallel} \right) - \mathbf{w} \cdot \mathbf{a} \left(\frac{\sin\psi}{\cos\theta} + \cos\psi r_{\parallel} \right) + \\ &+ \mathbf{w} \cdot \mathbf{o} \left(\frac{\cos\psi}{\cos\theta} - \sin\psi r_{\parallel} \right). \end{aligned}$$

It can be seen that $\cos\psi \mathbf{o} - \sin\psi \mathbf{a} = \mathbf{l}$, and that $\sin\psi \mathbf{o} + \cos\psi \mathbf{a} = \mathbf{n} \times \mathbf{l}$. Therefore,

$$\frac{E_t}{\sqrt{E_x^2 + E_y^2}} \cos\theta = \left(\frac{\sin\psi}{\cos\theta} + \cos\psi r_{\parallel} \right) \frac{1}{d} \mathbf{n} \cdot \mathbf{t} + \left(\frac{1}{\cos\theta} \mathbf{l} - r_{\parallel} (\mathbf{n} \times \mathbf{l}) \right) \cdot \mathbf{w}$$

Thus, when the 3D localization of the line is known, we have a direct relationship between camera motion and brightness derivatives for every pixel along the line, that is

$$\frac{E_t}{\sqrt{E_x^2 + E_y^2}} \cos\theta = \begin{pmatrix} -\frac{\sin\phi}{\cos\theta} - r_{\parallel} \cos\phi \sin\theta \\ \frac{\cos\phi}{\cos\theta} - r_{\parallel} \sin\phi \sin\theta \\ -r_{\parallel} \cos\theta \\ \cos\phi \frac{\sin\psi}{d} + r_{\parallel} \cos\phi \cos\theta \frac{\cos\psi}{d} \\ \sin\phi \frac{\sin\psi}{d} + r_{\parallel} \sin\phi \cos\theta \frac{\cos\psi}{d} \\ -\tan\theta \frac{\sin\psi}{d} - r_{\parallel} \sin\theta \frac{\cos\psi}{d} \end{pmatrix}^T \begin{pmatrix} \mathbf{w} \\ \mathbf{t} \end{pmatrix} \quad (15)$$

Weighing with the gradient magnitude as proposed above, a linear least-squares approach allows to estimate the camera motion in a direct way. The formulation used can be seen in Appendix C. To do this, support regions of at least three lines must be used, and these lines must fulfill the requirements indicated above.

This direct computation seems more useful than the previous one when both the 3D orientation and the 3D position of the lines are available. However, with this direct solution we cannot obtain the rotation of the camera knowing only the 3D orientation of the lines, because the effects of the 3D line orientation and the 3D line location are mixed. Using the direct solution it can be observed (equation (15)) that the rotation could be obtained separately only when there is no translation at all, or the depth is very high (being these situations equivalent).

5 Experimental results

Experiments with real images acquired with a camera of 12 mm. focal length and an image size of 370x256 pixels have been carried out. Motion has been achieved by means of a PUMA robot arm with a camera coupled to the hand. Its controller provides a motion estimate, that serves to verify the algorithm.

We have used simple masks to extract the gradient, after gaussian filtering to smooth the images. We have observed that the method works well when the disparity in the image is small. Experimentally we have observed that disparities of about one pixel provide good results, and disparities higher than two pixels provide imprecise results. This restriction is often stated in optical flow methods. One

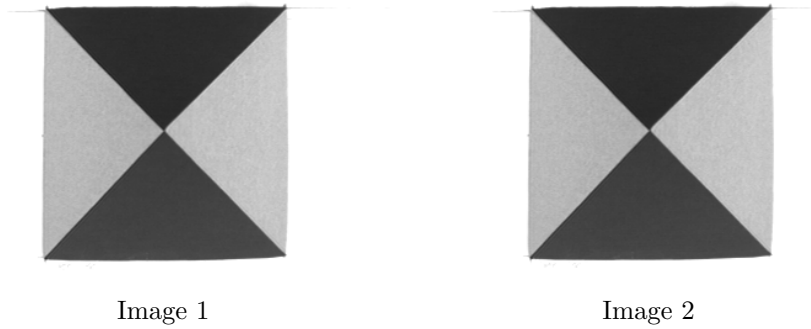


Figure 6: Two images of the pyramid in one experiment.

	<i>Test 1</i>			<i>Test 2</i>			<i>Test 3</i>		
	<i>C</i>	<i>M</i>	σ	<i>C</i>	<i>M</i>	σ	<i>C</i>	<i>M</i>	σ
V_x	0.5	5.61E-1	1.70E-2	0.0	-1.70E-2	1.38E-1	0.0	5.13E-2	1.78E-2
V_y	0.0	-9.73E-3	6.49E-3	0.0	1.85E-1	5.98E-2	0.0	-1.36E-2	4.93E-2
V_z	0.0	-1.90E-1	7.23E-2	2.0	2.46	3.20E-1	-1.0	-9.87E-1	3.78E-2
$ \mathbf{t} $	0.5	5.99E-1	3.71E-2	2.0	2.47	3.19E-1	1.0	9.90E-1	3.79E-2
α	0	18.89	6.23	0	5.38	1.61	0	4.19	1.03
W_x	0.0	-1.60E-2	7.85E-4	0.0	8.94E-3	1.24E-2	0.0	2.69E-2	4.86E-3
W_y	0.0	1.10E-2	1.16E-3	0.0	1.02E-2	2.53E-2	0.0	-7.39E-3	1.50E-3
W_z	0.0	3.88E-4	1.98E-4	0.0	-2.67E-2	1.43E-2	0.0	-3.41E-3	1.04E-2

Table 1: Motion commanded to the robot (C) and estimated motion (mean M , and standard deviation σ). They are expressed in mm. and degrees per frame. We can see both translation components (V_x, V_y, V_z), rotation components (W_x, W_y, W_z), translation magnitude ($|\mathbf{t}|$) and deviation angle (α) between estimated translation and commanded translation.

way to circumvent this restriction is to apply the differentiation in a coarse-to-fine manner, but such extensions are not examined in detail here. We are specially interested in showing the goodness of our method and in comparing it with an equivalent approach based on line correspondences. The known restrictions and accuracy of approaches based on the brightness constraint [7, 6], and approaches based on line features [13, 25] are out the scope of these experiments.

The first scene used corresponds with a pyramid on a white table observed from 360 mm. on the top (Fig. 6). We present tests with three different motions. In test 1 the commanded motion is a translation of 0.5 mm parallel to the image plane, that turns out in a disparity of about 1 pixel. In tests 2 and 3 the translation is made in the direction of the focal axis, and the maximum disparity is about 1 pixel and 0.5 pixels respectively. Each test has been repeated ten times in order to have mean and standard deviation of the estimated translation and rotation velocity. In this way we can evaluate the robustness of the method to the image formation and acquisition processes. Straight edges having a minimum gradient of 15 *glu/pix* and a minimum length of 50 *pix*, have been extracted (they correspond with the edges of the pyramid).

In table 1 the estimated motion parameters are shown. It can be observed that the results are better carrying out translation in Z direction than making translation parallel to image plane. As it is well known, translation in the direction of the focal axis provides flow lines converging into the field of view. However, translation parallel to the image plane provides flow lines converging outside the field of view, and therefore the estimation turns out to be less accurate. It is appropriate to emphasize that from two very close (nearly the same) images, the algorithm provides the direction of translation with an error smaller than 5° and the translation magnitude with an error smaller than 10%.

The previous scene turns out appropriate to easily check the results, to determine the 3D depth of the lines and to carry out systematic experiments. Nevertheless, the proposed method also works in complex and more realistic scenes. We now present an experiment made using an indoor scene, where there is not such a good knowledge about scene depth and reflection properties of objects. The commanded motion is a translation parallel to the focal axis without rotation, being the maximum disparity about 0.7 pixels. However, using the PUMA robot (which is an angular robot) the camera rotates with angles that are too small to be corrected by the robot controller. Two consecutive images

	C	RE	M	σ
W_x	0	-1.44E-1	-2.18E-1	8.59E-3
W_y	0	-4.39E-2	-8.02E-2	3.21E-3
W_z	0	-5.50E-2	-4.01E-2	1.72E-3

Table 2: Rotation commanded to the robot (C), rotation obtained after motion by the encoders of the robot (RE), and rotation estimated by vision (mean M , and standard deviation σ). They are expressed in degrees per frame.

of this scene can be seen in Fig. 7.



Figure 7: Two consecutive images of the indoor scene. The disparity is less than 0.7 pixels, that turns out to be inappreciable on sight.

In this case, we do not have knowledge of scene depth, and therefore we cannot compute the translation. However, it is possible to know the 3D orientation of some of the straight edges. We have considered the seven longest lines, assuming they are parallel to the image plane and vertical or horizontal. With this knowledge of depth we can obtain the camera rotation directly from the brightness information on their support regions. In table 2 the estimated rotation is shown. The experiment has been repeated ten times in order to also evaluate the stability of the solution. The proposed method obtains a small rotation which is stable and coherent with the rotation obtained from the encoders of the robot.

5.1 Comparison with a correspondence-based approach

It has been said [4] that image variations are obtained better using corresponding features than using the image brightness constraint. Obviously, this is correct when there is a large disparity and the brightness constraint and its gradient-based approach are not applicable. However, with close images, the triangulation to obtain 3D information is ill-conditioned. The support region concept combined with the brightness constraint can provide better estimations because all information available of the edge can be used.

We have compared the proposed method with an approach based on line correspondences. Both are equivalent except for the way used to obtain the motion of the projected line. The alternate correspondence-based approach can be outlined as follows:

1. The straight edges are extracted from the two images, and the correspondences between images are established.
2. Taking the representation of the line in the first and second images (\mathbf{n}_1 and \mathbf{n}_2), the flow of the line is obtained in a discrete way as, $\dot{\mathbf{n}} = \mathbf{n}_2 - \mathbf{n}_1$.
3. From the 3D orientation of the line, parameters w_{ol}, t_{nl} can be obtained as $w_{ol} = \dot{\mathbf{n}} \cdot \mathbf{a}$ and $t_{nl} = \dot{\mathbf{n}} \cdot \mathbf{o}$.
4. With these parameters, camera motion is obtained in the same way as proposed in §4.1.

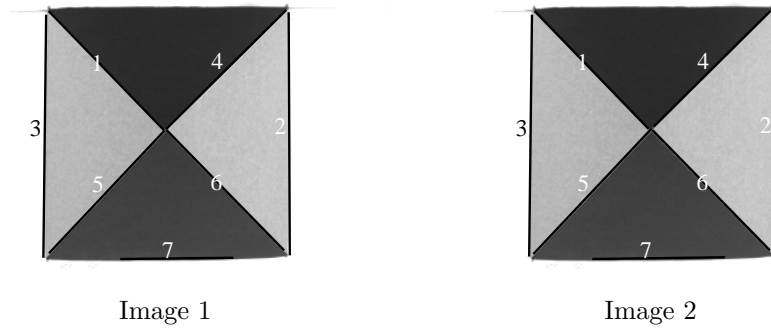


Figure 8: Two images of the pyramid scene, with the lines extracted and matched, used to compare the proposed method with an approach based on correspondences.

	V_x	V_y	V_z	W_x	W_y	W_z
Commanded	0.0	0.0	2.0	0.0	0.0	0.0
Brightness-based	-1.70E-2	1.85E-1	2.46	8.94E-3	1.02E-2	-2.67E-2
Correspondences	5.43E-2	9.75E-2	2.01	-1.12E-2	-5.70E-3	6.19E-3
Commanded	0.0	0.0	-1.0	0.0	0.0	0.0
Brightness-based	5.13E-2	-1.36E-2	-9.87E-1	2.69E-2	-7.39E-3	-3.41E-3
Correspondences	1.21E-1	4.24E-2	-3.72E-1	2.92E-2	-1.97E-2	1.17E-2

Table 3: Mean values of the estimated camera motion, comparing the brightness based approach and a correspondence-based approach in two experiments. Translation (V_x, V_y, V_z) is expressed in mm. and rotation (W_x, W_y, W_z) in degrees.

The pyramid scene (which is simple enough) has been used with a commanded translation in the direction of the focal axis. We present results with a translation of 2 and 1 mm. per frame. The maximum disparity is about 1 pixel and 0.5 pixels respectively. We have manually assured a correct matching in the correspondence-based approach, using the same straight edges in both methods. In Fig. 8 we can see an example of the images and the lines used.

In table 3, the mean values of the estimated motion with both methods are shown. In table 4, we can see the mean values and standard deviation of the magnitude and deviation of the estimated translation. In the first case, both methods provide similar results. The translation magnitude is better using correspondences, but the direction of translation and its variability are similar in both methods.

However, the same does not happen in the second case, when the disparity is smaller. In this case, it can be observed that the correspondence-based method does not work correctly. The magnitude and direction of the estimated translation are very bad, and their variability is high. The brightness-based method obtains better estimations with smaller variability. Our approach turns out to be most useful when disparity is small.

If we have an edge with a gradient magnitude of at least 15 glu/pix , and we assume typical errors of 3 glu/pix and 3 $glu/frame$ in the computation of spatial and temporal gradients respectively,

	$ \mathbf{t} $	$\sigma_{ \mathbf{t} }$	$\bar{\alpha}$	σ_{α}
Commanded	2.0	-	0.0	-
Brightness-based	2.47	3.19E-1	5.37	1.60
Correspondences	2.01	1.19E-1	5.25	2.15
Commanded	1.0	-	0.0	-
Brightness-based	9.90E-1	3.79E-2	4.19	1.03
Correspondences	3.99E-1	1.04E-1	21.84	4.77

Table 4: Mean values and standard deviation of the translation magnitude ($|\mathbf{t}|$) and the deviation angle (α) between estimated translation and commanded translation, comparing the brightness-based approach and an equivalent correspondence-based approach in two experiments. They are expressed in mm. and degrees.

then we can obtain an accuracy of about 0.25 pixels in the projected motion. 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 3D triangulation is ill-conditioned and the accuracy of the feature extractor is not good enough to obtain 3D 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

In this paper we have presented a new approach to compute 3D motion. In the proposed method, the image is segmented into regions supporting straight edges, applying the brightness constraint and using all geometric information available about the lines. This combination of feature-based and normal flow techniques allows to consider topological relations between pixels of lines whose localization is known, and it allows also to consider brightness information in motion computation.

Based on the line motion field (used in correspondence-based approaches), we have obtained expressions relating brightness information to the kinematics of the line and the camera motion. These expressions have been used to obtain the translation and rotation of the camera between two images, knowing the 3D localization of at least three straight edges. The experiments have shown that the method turns out to be simple and accurate in obtaining motion when disparity is small.

The proposed approach has been experimentally compared with a geometrically equivalent approach based on corresponding lines. The results show that, with disparity less than half a pixel, the proposed method provides better and more stable results than the correspondence based approach, being less the computational time of our method. Therefore, when there are triangulation problems to obtain 3D information, it turns out very useful to consider constraints about the image brightness.

Although the experiments have been done using only two images, a higher number of images can be considered. Besides that, the proposed approach can be combined with classical approaches based on line features [13]. In this case, images must be obtained in groups of at least two, close to each other. For example, the proposed motion algorithm could be combined with a mobile stereo system based on line features. The stereo provides the 3D localization of the lines, and the proposed method will give the instantaneous velocity of the cameras using these 3D lines.

Acknowledgments

This work was partially supported by projects TAP94-0390 and TAP97-0992-C02-01 of the Comisión Interministerial de Ciencia y Tecnología (CICYT).

A Estimation of the line motion parameters

The motion parameters of the line, when its 3D orientation is known, can be obtained by minimizing the following expression:

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

where

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

Taking the derivative of this expression with respect to t_{nl} , w_{ol} and equating to zero, we arrive at a system of linear equations (in function of some integral factors), that can be solved to obtain t_{nl} and w_{ol} as

$$\begin{bmatrix} t_{nl} \\ w_{ol} \end{bmatrix} = \begin{bmatrix} S_{t2} & S_{tw} \\ S_{tw} & S_{w2} \end{bmatrix}^{-1} \begin{bmatrix} S_{et} \\ S_{ew} \end{bmatrix}$$

where the integral factors are extracted along the support region in a sequential way as follows:

$$\begin{aligned}
S_{t2} &= \sum_{x,y}^{LSR} (E_x^2 + E_y^2) \left[\frac{\sin \psi}{\cos \theta} + \cos \psi r_{\parallel} \right]^2 \\
S_{w2} &= \sum_{x,y}^{LSR} (E_x^2 + E_y^2) \left[\frac{\cos \psi}{\cos \theta} - \sin \psi r_{\parallel} \right]^2 \\
S_{tw} &= \sum_{x,y}^{LSR} (E_x^2 + E_y^2) \left[\frac{\sin \psi}{\cos \theta} + \cos \psi r_{\parallel} \right] \left[\frac{\cos \psi}{\cos \theta} - \sin \psi r_{\parallel} \right] \\
S_{et} &= \cos \theta \sum_{x,y}^{LSR} \left[\sqrt{E_x^2 + E_y^2} \left(\frac{\sin \psi}{\cos \theta} + \cos \psi r_{\parallel} \right) \right] [E_t] \\
S_{ew} &= \cos \theta \sum_{x,y}^{LSR} \left[\sqrt{E_x^2 + E_y^2} \left(\frac{\cos \psi}{\cos \theta} - \sin \psi r_{\parallel} \right) \right] [E_t]
\end{aligned}$$

B Motion estimation from several lines

Several lines are frequently available and therefore, camera rotation and translation can be obtained in a more robust way by using a least-squares approach. The formulation used (where l represents each line) is the following:

$$\begin{aligned}
\mathbf{z}_l^T &= [w_{ol}, t_{nl}] ; \quad \mathbf{A}_l = \begin{bmatrix} \mathbf{o}_l^T & , & 0, & 0, & 0 \\ -\mathbf{a}_l^T & , & & \frac{\mathbf{n}_l^T}{d_l} & \end{bmatrix} \\
\mathbf{z}^T &= [\mathbf{z}_1^T, \mathbf{z}_2^T, \dots, \mathbf{z}_l^T] ; \quad \mathbf{A}^T = [\mathbf{A}_1^T, \mathbf{A}_2^T, \dots, \mathbf{A}_l^T]
\end{aligned}$$

In this way, the motion parameters $\mathbf{x}^T = [\mathbf{w}^T, \mathbf{t}^T]$ that minimize $(\mathbf{A}\mathbf{x} - \mathbf{z})^T(\mathbf{A}\mathbf{x} - \mathbf{z})$ are obtained, using the pseudo-inverse, as

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

Considering the estimation errors of line motion parameters independent and equally distributed, uncertainty can be obtained from the residue of the least-squares fitting [26]. The variance of the estimation error is

$$s^2 = \frac{1}{2l-6} (\mathbf{z}^T \mathbf{z} - \mathbf{x}^T \mathbf{A}^T \mathbf{z}).$$

The residue is divided by the difference between the number of equations and the number of parameters obtained, obtaining an unbiased estimate.

Therefore, the uncertainty of camera motion, characterized by the covariance matrix of \mathbf{x} is

$$Cov(\mathbf{x}) = (\mathbf{A}^T \mathbf{A})^{-1} s^2.$$

On the other hand, when an uncertainty model of line motion parameters is available (obtained for example from the residue of their fitting), camera motion and its uncertainty can be obtained weighing each measurement with the inverse of its covariance. In this case, the camera motion and its uncertainty would be

$$\begin{aligned}
\mathbf{x} &= (\mathbf{A}^T \mathbf{Q}^{-1} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{Q}^{-1} \mathbf{z} \\
Cov(\mathbf{x}) &= (\mathbf{A}^T \mathbf{Q}^{-1} \mathbf{A})^{-1}
\end{aligned}$$

where $\mathbf{Q}_l = Cov(w_{ol}, t_{nl})$ and $\mathbf{Q} = diag(\mathbf{Q}_1, \mathbf{Q}_2, \dots, \mathbf{Q}_l)$.

C Direct motion computation

Considering (15) we have an equation for each point in the image belonging to a line support region of known 3D localization. A least-squares approach allows to estimate camera motion in a direct way.

Thus, measurement equation $z_p = \mathbf{B}_p \mathbf{x}$ for each pixel (p) in a support region is

$$E_t \cos \theta = \sqrt{E_x^2 + E_y^2} \begin{pmatrix} \frac{-\sin \phi}{\cos \theta} - r_{\parallel} \cos \phi \sin \theta \\ \frac{\cos \phi}{\cos \theta} - r_{\parallel} \sin \phi \sin \theta \\ -r_{\parallel} \cos \theta \\ \cos \phi \frac{\sin \psi}{d} + r_{\parallel} \cos \phi \cos \theta \frac{\cos \psi}{d} \\ \sin \phi \frac{\sin \psi}{d} + r_{\parallel} \sin \phi \cos \theta \frac{\cos \psi}{d} \\ -\tan \theta \frac{\sin \psi}{d} - r_{\parallel} \sin \theta \frac{\cos \psi}{d} \end{pmatrix}^T \begin{pmatrix} \mathbf{w} \\ \mathbf{t} \end{pmatrix}$$

The motion $\mathbf{x}^T = [\mathbf{w}^T, \mathbf{t}^T]$ can be estimated as

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

where

$$\mathbf{z}_d^T = [z_1, z_2, \dots, z_p] ; \quad \mathbf{B}^T = [\mathbf{B}_1^T, \mathbf{B}_2^T, \dots, \mathbf{B}_p^T] .$$

References

- [1] J.K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images - a review. *Proceedings of the IEEE*, 76(8):917–935, 1988.
- [2] Y. Liu and T.S. Huang. Determining straight line correspondences from intensity images. *Pattern Recognition*, 24(6):489–504, 1991.
- [3] H. Kollnig and H.H. Nagel. 3d pose estimation by fitting image gradients directly to polyhedral models. In *International Conference on Computer Vision*, pages 569–574, Cambridge, Massachusetts, June 1995.
- [4] O. Faugeras. *Three-Dimensional Computer Vision. A Geometric Viewpoint*. The MIT Press, Massachusetts, 1993.
- [5] B.K.P. Horn. *Robot Vision*. MIT Press, Cambridge, Mass., 1986.
- [6] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Systems and experiment performance of optical flow techniques. *Int. Journal of Computer Vision*, 12(1):43–77, 1994.
- [7] E. De Micheli, V. Torre, and S. Uras. The accuracy of the computation of optical flow and of the recovery of motion parameters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(5):434–447, 1993.
- [8] D.W. Murray and B.F. Buxton. *Experiments in the Machine Interpretation of Visual Motion*. The MIT Press, Massachusetts, 1990.
- [9] 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.
- [10] 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.
- [11] B. Gaii-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.
- [12] J.J. Guerrero, C. Sagüés, and F.J. Domínguez. Using direct methods to obtain motion from 3d lines. In *International Conference on Advanced Robotics*, pages 687–693, Barcelona, September 1995.

- [13] T.S. Huang and A. N. Netravali. Motion and structure from feature correspondences: A review. *Proceedings of the IEEE*, 82(2):252–268, 1994.
- [14] B.K.P. Horn and E.J. Weldon. Direct methods for recovering motion. *International Journal of Computer Vision*, (2):51–76, 1988.
- [15] D. Sinclair, A. Blake, and D. Murray. Robust estimation of egomotion from normal flow. *International Journal of Computer Vision*, 13(1):57–69, 1994.
- [16] J.J. Guerrero. *Percepción de Movimiento y Estructura con Visión basada en Contornos Rectos*. PhD thesis, Dpto. de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, Spain, Mayo 1996.
- [17] Y. Aloimonos and Z. Duric. Estimating the heading direction using normal flow. *Int. Journal of Computer Vision*, 13(1):33–56, 1994.
- [18] S. Negahdaripour and C.H. Yu. A generalized brightness change model for computing optical flow. In *Fourth International Conference on Computer Vision*, Berlin, May 1993.
- [19] Bernd Jähne. *Digital Image Processing*. Springer-Verlag, Berlin-Heidelberg, 1993.
- [20] Ajit Singh and Peter Allen. Image flow computation: An estimation-theoretic framework and a unified perspective. *CVGIP: Image Understanding*, 56(2):152–177, 1992.
- [21] C.A. Rothwell, J.L. Mundy, W. Hoffman, and V.D. Nguyen. Driving vision by topology. In *International Symposium on Computer Vision*, pages 395–400, Coral-Gables, Florida, Nov. 1995.
- [22] E. Lutton, H. Maitre, and J. Lopez-Krahe. Contribution to the determination of vanishing points using hough transform. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(4):430–438, 1994.
- [23] T. Viéville, E. Clergue, and P. Facao. Computation of ego-motion and structure from visual and inertial sensors using the vertical cue. In *Fourth International Conference on Computer Vision*, pages 591–598, Berlin, May. 1993.
- [24] N. Navab. *Motion of Lines, and Cooperation Between Motion and Stereo*. PhD thesis, University of Paris XI, Orsay, Paris, France, Jan. 1993.
- [25] J. Weng, T.S. Huang, and N. Ahuja. *Motion and Structure from Image Sequences*. Springer-Verlag, Berlin-Heidelberg, 1993.
- [26] N. Draper and H. Smith. *Applied Regression Analysis*. Wiley, New York, 1981.