

DIIS - I3A
Universidad de Zaragoza
C/ María de Luna num. 1
E-50018 Zaragoza
Spain

Internal Report: 2001-V03

Estimating the Motion Direction from Brightness Gradient on Lines¹

Guerrero J.J., Sagüés C.

If you want to cite this report, please use the following reference instead:

Estimating the Motion Direction from Brightness Gradient on Lines, Guerrero J.J., Sagüés C., *IEEE Trans. on SYSTEMS, MAN, AND CYBERNETICS. PART C: APPLICATIONS AND REVIEWS*, Vol. 31(3), pages 419-426, 2001.

¹This work was supported by projects TAP97-0992, DPI2000-1265, DPI2000-1272.

Estimating the Motion Direction from Brightness Gradient on Lines

J.J. Guerrero & C. Sagüés

Abstract

In previous works we combined feature-based techniques and optical flow methods to obtain depth or motion. The expressions relating the brightness constraint to the 3D localization and motion of a line and its projection were established. In this paper, those expressions have been used to obtain the motion direction of a camera when the rotation velocity is bounded, without assumptions about the depth of lines. Our approach exploits the visibility constraint and it allows to make use of a priori information about the scene or the motion. With the proposed technique, the topology (easily extracted in the image) that relates adjacent edge elements into line segments is exploited to better compute camera motion. Besides the motion direction, our method allows also to compute the rotation velocity when lines in prominent 3D directions are available.

Keywords

Motion and structure from vision, straight edges, optical flow, visibility constraint.

I. INTRODUCTION

Shape and motion information from vision has been usually extracted using corresponding features or optical flow measures [1]. Geometric features provide an efficient way to select, concentrate and manipulate vision information. In particular, features like straight edges are easy to extract and match [2], [3], being robust to partial occlusions. Besides that, 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. Even for a human, it is necessary to view the gray level image to recognize a scene, because image of edges are not enough [4]. In methods based on geometric features, the brightness information is not used after the features have been extracted and matched, and therefore useful information is discarded.

On the other hand, methods based on optical flow assume that image brightness can be related with image motion [5], usually considering that the brightness does not vary with time [6], [7]. These approaches work well with small disparities having a small computational cost per projected motion measurement [8]. They can be applied uniformly across the image, but when spatial or temporal gradients are small, the results are very sensitive to noise [9].

As we have previously proposed, it is useful to combine the geometric description of the scene with information about image brightness. So, we have computed the depth of the scene from motion [10] and the motion of the camera from scene depth combining the brightness constraint and geometric features [11]. We use straight edges extracted with the method proposed by Burns [12]. This extractor provides not only the geometric representation of the image edge, but also a segmentation of the image 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 edge. These line support regions allow to combine methods based on geometric features with methods based on brightness information. Thus, the kinematics of a straight edge, in relation to its location in the image and in the scene, was obtained and related to the brightness constraint [11].

In this paper the proposed combination of features and flow techniques is aimed to obtain the motion direction without a priori depth information. It is known that motion and structure cannot be obtained from line flow, even if many lines are available, because there exists so many unknowns of structure as equations, and there is not available equations to compute motion. Some experiments have been made using second derivatives of flow along rectilinear edges [13], however high order derivatives are very sensitive to noise. In a general case, more views or additional depth information are needed to obtain camera motion. However, in this reasoning, the visibility constraint was not considered. Using the visibility constraint (the scene observed is always in front of the camera), and assuming pure translational motion or general rigid motion with a bounded rotation, it is possible to obtain the motion direction from normal flow information [14], [15], [16]. From this approach, the motion direction is obtained as a region of possible values, whose size indicates its uncertainty.

The method proposed in this paper computes the motion direction of a camera from the brightness information on straight edges, making use of the visibility constraint. There are works [15], [16] that compute the same, but using local normal flow information or isolated edge elements. There are also works that compute the translation direction in a direct way using confidence measures [17]. Two aspects must be taken into account in these approaches: the process of

This work was partially supported by projects TAP97-0992, DPI2000-1265, DPI2000-1272.

The authors are with the Dpto. Informática e Ingeniería de Sistemas, University of Zaragoza, María de Luna 3, E-50015 Zaragoza, Spain, email: {jguerrer, csagues}@posta.unizar.es.

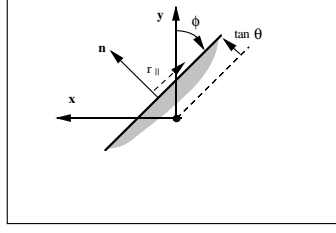


Fig. 1. Projected line representation

computing the motion direction is very conditioned by the orientation of the measures of normal flow; and the visibility constraint on an edge element is equivalent to the visibility constraint of other collinear elements.

Using local measures of normal flow, the direction of the brightness gradient is locally computed and it can result inaccurate. Thus, an heuristic filtering has been proposed [16] from experimental knowledge, which supposes the selection of edge elements collinear to each other. Therefore, it is interesting to consider the topology (easily extracted in the image) that relates adjacent edge elements into line segments, as it is proposed in our paper. Besides that, our proposal allows to obtain robust estimates of flow because not only the gradient direction but also the normal flow magnitude is globally obtained in the LSR.

In the basic algorithm the effect in the image of the camera translation must be bigger than the effect of the rotation. However, information about 3D orientation of the lines is frequently available, for example from vanishing point detection [18] or using the vertical cue [19]. Our proposal takes advantage of significant lines (whose 3D orientation is known) to better compute normal flow. Besides, from significant lines, camera rotation can also be computed and their effect compensated, to obtain afterwards the motion direction with the basic algorithm.

II. REPRESENTATION OF LINE AND CAMERA MOTION

We adopt the pinhole camera model with a planar screen. 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 $x = \frac{X}{Z}$, $y = \frac{Y}{Z}$. In our work, the camera is supposed to move with respect to the scene and its motion to be composed of translation \mathbf{t} and rotation $\mathbf{w} = [W_x, W_y, W_z]^T$ velocities, expressed in the camera reference system.

A. 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 our work, two parameters of its projection in the image and two additional depth parameters are used 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. 1). In this way, the vector \mathbf{n} in the x direction of the new reference system will be perpendicular to the projection plane of the line.

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 included in $(-\frac{\pi}{2}, +\frac{\pi}{2})$. 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, both parameters named above are combined with two additional parameters. Thus, we 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. The fourth parameter (d) is defined as the distance from the camera reference system to the 3D line (Fig. 2), always being greater than zero.

Using this representation the vectors that define the projected line and the direction of the 3D line are

$$\begin{aligned}\mathbf{n} &= (c\phi c\theta, s\phi c\theta, -s\theta)^T, \\ \mathbf{a} &= (c\phi s\theta c\psi + s\phi s\psi, s\phi s\theta c\psi - c\phi c\psi, c\theta c\psi)^T,\end{aligned}$$

where c is \cos and s is \sin . Details about the advantages of this representation are given in [11].

III. CAMERA MOTION AND LINE FLOW

As mentioned above, we obtained [11] the relations between the image brightness along a straight edge, the localization of the line in the space and the 3D camera motion, based on the line motion field [20]. The goal was to have expressions which take the available topological information combining directly the image brightness and the camera motion. Here,

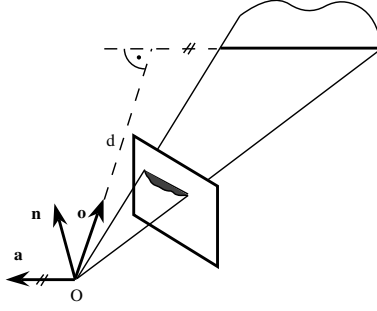


Fig. 2. Representation of the 3D line using its plane of projection.

we have retaken the expressions needed to define the computation of the line flow due to translation, which is the basic information used in our proposal.

It is known that the brightness constraint that relates the temporal and spatial brightness gradient (E_t, E_x, E_y) , with the projected motion (u, v) to obtain the normal flow is $(E_x, E_y) \cdot (u, v)^T = -E_t$. This is the topological level often used in optical flow methods. However, in this way the gradient orientation is obtained locally. Experimentally we 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 proposed to extend this constraint to image regions corresponding to lines, combining flow methods and geometric features [11]. Association of edge elements into lines, turns out very useful and can be carried out easily in images of man-made environments. With our line representation the relation between the brightness information and the kinematics of the projected line is,

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

where $r_{\parallel} = y \cos \phi - x \sin \phi$ is the distance in the line from the generic point (x, y) to the point closest to the image center (Fig. 1).

Information in the image about the 3D orientation of the lines is frequently available, for example from vanishing point [18] or using the vertical cue [19]. Therefore, it is interesting to isolate the motion information that could be obtained from the 3D orientation of the line.

Measuring the flow of a projected line, the camera rotation cannot be obtained around the 3D line (symmetry of the line), nor around \mathbf{n} (symmetry of the projected line). The translation cannot be obtained along the direction of the 3D line (symmetry of the line), nor along \mathbf{o} (symmetry of the projected line). As deduced [20], from the flow of a projected line and its 3D orientation we can recover \mathbf{w} in the direction \mathbf{o} knowing neither line depth nor translational motion. Moreover, the translational motion along direction \mathbf{n} is coupled with the rotation in the camera reference system around an axis (\mathbf{a}) parallel to the 3D line.

Using the proposed parameterization of the line kinematics, we can easily consider the 3D line orientation without knowledge of its 3D position. We obtained [11] an equation involving the 3D orientation of the line, with motion parameters associated to each line ($t_{nl} = -\mathbf{w} \cdot \mathbf{a} + \frac{\mathbf{t} \cdot \mathbf{n}}{d}$, $w_{ol} = \mathbf{w} \cdot \mathbf{o}$) and its brightness gradient. Thus, for each pixel of the projected line, we obtained the following equation:

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

Using our representation θ is always small with real cameras (they have a small field of view), and therefore the special case $\cos \theta = 0$ cannot happen in these expressions.

A. Computing line flow from brightness gradient

With the expressions deduced we can estimate the line flow directly from brightness on the LSR. To do that, two close images are needed. Line support regions (LSR) and the straight edges are extracted from the first image [12]. The temporal gradient, that is necessary to extract motion information, can be obtained using also the second image.

Two cases can be considered for its computation, in function of the available knowledge about the 3D orientation of the line.

- *Non significant lines.* When only the projected line is known, a least-squares with the expression (1) can be used to obtain $\dot{\theta}$ and $\dot{\phi}$. The measurement corresponds with the normal flow of each point into the LSR, and therefore

the error to minimize has a physical interpretation. 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. Finally, the expression to minimize is

$$J_{ns} = \sum_{x,y}^{LSR} \left[E_t \cos \theta + \frac{\sqrt{E_x^2 + E_y^2}}{\cos \theta} \dot{\theta} - r_{\parallel} \cos \theta \sqrt{E_x^2 + E_y^2} \dot{\phi} \right]^2. \quad (3)$$

• *Significant lines.* When the 3D orientation is also known the expression (2) can be used. A least-squares on the LSR allows to obtain the motion parameters t_{nl} and w_{ol} of the line. Thus, weighting with the brightness gradient magnitude, the expression to minimize is

$$J_s = \sum_{x,y}^{LSR} [-E_t \cos \theta + t_{nl} FT(x, y) + w_{ol} FW(x, y)]^2, \quad (4)$$

where

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

B. Line flow due to translation

To compute the motion direction when there is no knowledge about scene depth, the optical flow due to translation of the lines must be obtained. Assuming there is no camera rotation ($\mathbf{w} = 0$), we can define the line flow due to translation \mathbf{u}_t of a line as:

$$\mathbf{u}_t = \dot{\mathbf{n}}_{(\mathbf{w}=0)} = \frac{\mathbf{t} \cdot \mathbf{n}}{d} \mathbf{o}. \quad (5)$$

Two cases can also be considered for its computation:

- *Non significant lines.* When only the projected line (ϕ, θ) is known.
 - Estimate $\dot{\phi}$ and $\dot{\theta}$ directly from the brightness on the LSR minimizing expression (3).
 - The line flow due to translation can be obtained as

$$\mathbf{u}_t = \dot{\phi} \cos \theta \mathbf{l} - \dot{\theta} \mathbf{n} \times \mathbf{l},$$

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

- *Significant lines.* The projected line and its 3D orientation (ϕ, θ, ψ) are known. In this case, the sole parameter that includes all of the motion information of each line can be directly computed.
 - Estimate t_{nl} directly from the brightness, minimizing expression (4) with $w_{ol} = 0$. As only one parameter is estimated from brightness variations, the flow due to translation is computed more robustly. Thus, developing the least squares,

$$t_{nl} = \frac{\cos \theta \sum_{x,y}^{LSR} (E_t) \left[\sqrt{E_x^2 + E_y^2} \left(\frac{\sin \psi}{\cos \theta} + \cos \psi r_{\parallel} \right) \right]}{\sum_{x,y}^{LSR} (E_x^2 + E_y^2) \left[\frac{\sin \psi}{\cos \theta} + \cos \psi r_{\parallel} \right]^2}.$$

- As the 3D orientation of the line is known, we have

$$\mathbf{u}_t = t_{nl} \mathbf{o}.$$

In Fig. 3 the parameters needed to compute the line flow due to translation for non significant and for significant lines can be seen. The line flow due to translation can be seen as a rotation around the vanishing point of the line ($\mathbf{u}_t \times \mathbf{o} = 0$). Therefore, when rotation is zero, the 3D orientation of the non significant lines could be indirectly obtained from their projected flow, as $\mathbf{a} = \mathbf{n} \times \mathbf{o} = \mathbf{n} \times \frac{\mathbf{u}_t}{|\mathbf{u}_t|}$.

IV. RECOVERING THE MOTION DIRECTION FROM BRIGHTNESS ON LINES

As it is known, the line flow does not constraint the translation except when depth information is available, because the number of equations is equal to the number of depth unknowns. However, using the visibility constraint (the scene observed is always in front of the camera), and assuming pure translational motion or general rigid motion with a bounded rotation it is possible to obtain the motion direction from normal flow information [14], [15], [16]. In this section the line flow due to translation, computed previously, is used to obtain the motion direction applying the visibility constraint.

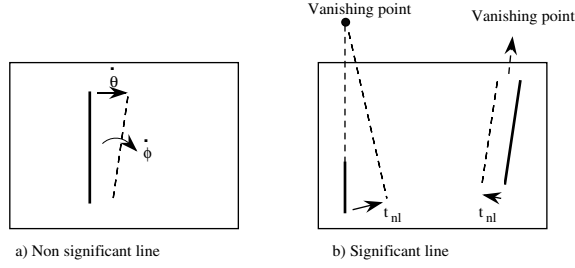


Fig. 3. a) Two parameters must be computed to estimate the flow due to translation of a projected line. b) If we know the 3D orientation of the line, its flow due to translation can be computed from only one parameter, because the flow due to translation can be seen as a rotation around the vanishing point of the line.

A. Visibility constraint and camera motion

The depth (Z_i) of a point on the 3D line, that is projected in the image plane as $\mathbf{x}_i = (x_i, y_i, 1)^T$, can be related with the depth (d) of the line,

$$Z_i = \frac{d}{\mathbf{x}_i \cdot \mathbf{o}}. \quad (6)$$

Assuming there is no rotation ($\mathbf{w} = 0$), we can combine this expression with the line flow due to translation \mathbf{u}_t (5), of a line \mathbf{n} , to arrive at,

$$Z_i = \frac{\mathbf{t} \cdot \mathbf{n}}{\mathbf{x}_i \cdot \mathbf{u}_t}. \quad (7)$$

Therefore, the visibility constraint ($Z_i > 0$) implies that $\text{sign}(\mathbf{t} \cdot \mathbf{n}) = \text{sign}(\mathbf{x}_i \cdot \mathbf{u}_t)$, for all of the points of the line that appear in the image.

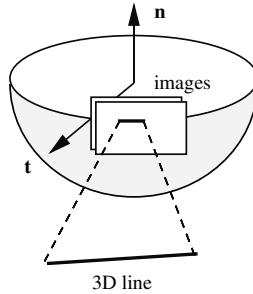


Fig. 4. The visibility constraint applied to a measure of normal flow limits the direction of motion to be in a hemisphere.

Each observation of normal flow allows to eliminate a half of the possible translation directions. Therefore, it constraints the set of possible motion directions to a hemisphere, the equator being the plane of projection of the line (Fig. 4). A voting process, followed by the searching of the most voted, allows to obtain a region compatible with most of the measures.

The size of the field of view and the distribution of the normals (\mathbf{n}) to the extracted lines have a considerable influence on the power of the set of constraints [16]. Due to the limited field of view, the normals of the lines projected on an image cannot be out of a band of the sphere of directions. The width of this band increases when the field of view increases [14], and the band is reduced to an smaller effective band when the normals of the observed lines are taken.

B. Computation of the rotation velocity and its compensation

The proposal to compute the motion direction from the positiveness of depth assumes zero rotation. Anyway, if there are enough significant lines, the rotation velocity can be estimated from their normal flow. Afterwards, the rotation can be compensated, and the algorithm that assumes pure translation can finally be applied.

To compute the rotation velocity of the camera, both motion parameters (w_{ol}, t_{nl}) of the significant lines must be extracted minimizing expression (4). As 3D orientation of significant lines is known, the motion parameters of each line (l) provide a linear equation in terms of the rotation velocity \mathbf{w} , which is $\mathbf{w} \cdot \mathbf{o}_l = w_{ol}$. Thus, from the motion parameters of at least three significant lines, the rotation velocity of the camera can be easily computed [11].

When the rotation velocity \mathbf{w} is obtained, their effect on the line flow can be compensated because it does not depend on depth. Thus, the line flow due to translation after compensation will be:

- *Non significant lines*

$$\mathbf{u}_t = \dot{\phi} \cos \theta \mathbf{l} - \dot{\theta} \mathbf{n} \times \mathbf{l} + \mathbf{w} \times \mathbf{n}.$$

- *Significant lines*

$$\mathbf{u}_t = (t_{nl} + \mathbf{w} \cdot \mathbf{a}) \mathbf{o}.$$

This compensation can also be carried out when the rotation velocity is obtained by another sensor.

C. Extended visibility constraint using segments

C.1 Bounded error of rotation

Until now, the rotation has been considered to be null or compensated. However, in practice, there will be always some uncertainties in its value that must be considered. The method has been extended to situations where there is a bound (Δ_w) on the uncertainty of the rotation velocity, $|\delta \mathbf{w}| < \Delta_w$.

It has been shown that the projected motion due to translation is coupled with the projected motion due to rotation along an axis (\mathbf{a}) parallel to the 3D line. Thus, due to the unknown rotation, the maximum error on the computed flow due to translation will be

$$|\delta \mathbf{u}_t|_{max} = \delta \mathbf{w} \cdot \mathbf{a} \leq |\delta \mathbf{w}| < \Delta_w.$$

The sign of the computed flow due to translation will be reliable when \mathbf{u}_t is greater than this maximum error. Therefore, to use the visibility constraint of a line, it must have a flow greater than the bound on the uncertainty of rotation velocity, $|\mathbf{u}_t| > \Delta_w$.

In some works [15], [16], the edges with a normal flow less than a limit are not used to compute the translation direction, and therefore the lines whose plane of projection is parallel to the camera translation are rejected. However, when there is qualitative information about the depth of the scene, the visibility constraint can be extended and small measures of projected motion can be considered.

C.2 Extended visibility constraint for the segment

In many practical situations, the depth can be supposed to be not only positive but also greater than a minimum value (D_{min}). This minimum depth makes the visibility constraint more powerful, because the hemisphere of compatible motion directions is converted to a smaller piece of a sphere. Due to the unknown scale factor between translation and size of the scene, which is inherent to the use of one mobile camera, the minimum depth must be expressed in units of the translation magnitude ($D_{min} = \frac{Z_{min}}{|\mathbf{t}|}$).

Similarly, when a maximum limit of the scene depth ($D_{max} = \frac{Z_{max}}{|\mathbf{t}|}$) is available, some possible translation directions can also be eliminated. Both limits can be estimated qualitatively in most situations from a priori information. In this way, the visibility constraint can be expressed as

$$D_{max} > \frac{\mathbf{t} \cdot \mathbf{n}}{\mathbf{x}_i \cdot \mathbf{u}_t} > D_{min} > 0. \quad (8)$$

A null or very small normal flow due to translation on a straight edge indicates one of the following situations:

- The 3D line is far away and therefore the camera translation does not produce projected flow.
- The translation is parallel to the plane of projection of the line.

When a maximum limit of the scene depth is available, only the second situation is reliable. In this case (small or null normal flow), the sign of the flow is not reliable to constraint the translation direction. However, the translation can be constrained to the directions providing small normal flow on the corresponding line (translation nearly parallel to the plane of projection of the line).

The rotation uncertainty weakens the visibility constraint. To robustly assure this constraint when we have a bound on the uncertainty of the rotation velocity, we define the limit values $\mathbf{u}_t^m, \mathbf{u}_t^M$ of the real normal flow due to translation on each line, in function of the estimated \mathbf{u}_t as

$$\mathbf{u}_t^m = \frac{|\mathbf{u}_t| - \Delta_w}{|\mathbf{u}_t|} \mathbf{u}_t, \quad \mathbf{u}_t^M = \frac{|\mathbf{u}_t| + \Delta_w}{|\mathbf{u}_t|} \mathbf{u}_t.$$

This uncertainty in the real line flow due to translation turns out equivalent to enlarge the limits of maximum and minimum depth. To determine all the possible translation directions which are compatible with a real line flow included in the range $(\mathbf{u}_t^m, \mathbf{u}_t^M)$, we must take the union of all the intervals of translation directions compatible with the flow. So, these constraints may be reformulated in the *extended visibility constraint*:

- If $|\mathbf{u}_t| > \Delta_w$, the translation direction must fit $\frac{\mathbf{t} \cdot \mathbf{n}}{\mathbf{x}_i \cdot \mathbf{u}_t^m} > D_{min}$ and $\frac{\mathbf{t} \cdot \mathbf{n}}{\mathbf{x}_i \cdot \mathbf{u}_t^M} < D_{max}$.
- If $|\mathbf{u}_t| \leq \Delta_w$, the translation direction must fit $\frac{\mathbf{t} \cdot \mathbf{n}}{\mathbf{x}_i \cdot \mathbf{u}_t^m} < D_{max}$ and $\frac{\mathbf{t} \cdot \mathbf{n}}{\mathbf{x}_i \cdot \mathbf{u}_t^M} < D_{max}$.

The extended visibility constraint would have to be applied to every point observed \mathbf{x}_i on each line. However, the points of the 3D segment closest and farthest to the image plane always coincide with its end points. Thus, the maximum and minimum values of $\mathbf{x}_i \cdot \mathbf{u}_t^m$ and $\mathbf{x}_i \cdot \mathbf{u}_t^M$ are obtained from the end points \mathbf{x}_s and \mathbf{x}_e of the segment. Therefore, it suffices with the application of the constraint only to both points.

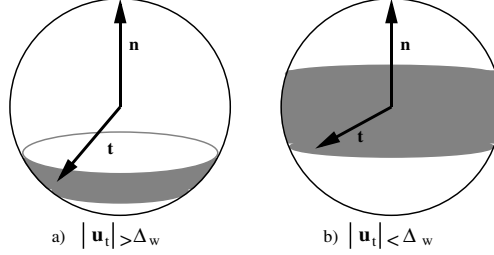


Fig. 5. The visibility constraint, considering the minimum and maximum depth, corresponds to a spherical band between two parallels of compatible translation directions, for each projected line.

So, the visibility constraint for a segment can be expressed as

$$Q_m < \mathbf{t} \cdot \mathbf{n} < Q_M, \quad (9)$$

where Q_m and Q_M are the minimum and maximum values obtained from the evaluation of the extended visibility constraint for both end points of the segment.

Therefore, the compatible translation directions must be on a band of the unit sphere (Fig. 5), being this band on a unique hemisphere if $|\mathbf{u}_t| > \Delta_w$ (Q_m and Q_M of equal sign) or in both hemispheres (less restrictive) if $|\mathbf{u}_t| < \Delta_w$ (Q_m and Q_M of different sign). In this way, only one visibility constraint (that includes those of every collinear edge element in the image) remains for each line. This result is valid even considering a minimum depth greater than zero, which expands some geometric interpretation of the visibility constraint [15].

D. Computing the motion direction

To obtain the motion direction, we carry out a voting process on an accumulation matrix which represents, in a discretized way, all the possible translation directions. Each segment votes to the directions which are compatible with the proposed visibility constraint. A subsequent process, that selects the most voted points, gives the direction or the set of directions compatible with the measurements. This process has been actually implemented with the adaptive Hough transform [22], which makes use of a reduced accumulating matrix in an iterative process.

The accumulation matrix is done parameterizing the translation direction with two angles in spherical coordinates $\mathbf{t} = (\sin \theta_t \cos \psi_t, -\sin \psi_t, \cos \theta_t \cos \psi_t)^T$. To speed up the voting process, it is interesting to have an explicit equation expressing a parameter as a function of the other. So, we can express the parameter θ_t as a function of ψ_t for the limit values Q_m or Q_M compatible with (9). For example,

$$\theta_t = \arccos \left(\frac{\frac{Q_m}{\cos \psi_t} + \tan \psi_t n_y}{\sqrt{n_x^2 + n_z^2}} \right) + \beta,$$

where $(n_x, n_y, n_z)^T = \mathbf{n}$ and $\beta = \text{atan2}(n_x, n_z)$.

V. 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 an industrial robot with a camera coupled to the hand. Its controller provides a motion estimate, that has been used to verify our algorithm.

As was reported in [11], we have used simple masks after Gaussian filtering to extract the brightness gradient. 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. To circumvent it, we can apply the differentiation in a coarse-to-fine

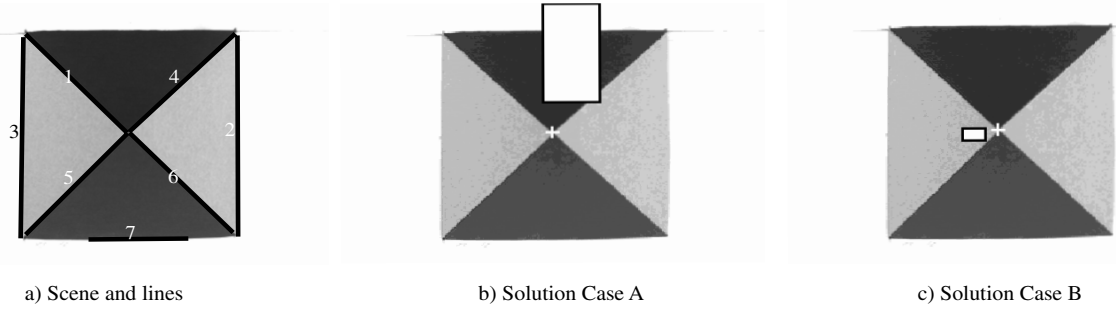


Fig. 6. a) Image of the pyramid with the lines used in the experiment. b), c) Approximate sets of solutions (white boxes) superimposed to the image. The cross (+) represent the direction of set-point motion. In the case A the solution is shifted towards the side of the image where no line has been used. In the case B the set of solutions is shorter because the rotation has been computed and compensated.

	ψ_t	θ_t	$\Delta\psi_t$	$\Delta\theta_t$	W_x	W_y	W_z
SetPoint	180.0	0.0	-	-	0.0	0.0	0.0
Case A	170.5	2.1	12.5	5.9	-	-	-
Case B	180.6	-2.7	1.7	2.5	2.4E-2	-5.9E-3	8.6E-3

TABLE I
MOTION DIRECTION USING THE PYRAMID SCENE.

manner, but such extensions are not examined here. The known restrictions and accuracy of approaches based on the brightness constraint [8], [7], and approaches based on line features [3] are out the scope of these experiments.

The first scene used corresponds with a pyramid, made on paper, placed on a white table. The camera stands approximately 360 mm. over the table. The set-point of the motion carried out with the robot, that moves the camera, has been of 1 mm. in the direction of the focal axis, being the disparity about 0.6 pixels. All straight edges of the pyramid have been used except the line on the top of the pyramid (Fig. 6). We have used $D_{min} = 100(|t|)$ and $D_{max} = 1000(|t|)$ as depth limits. With respect to the adaptive Hough transform we have implemented an accumulating matrix of size 24x24 obtaining the results in less than 7 iterations, starting from the complete range of translation directions ($\psi_t \in [-\frac{\pi}{2}, \frac{3\pi}{2}]$, $\theta_t \in [-\frac{\pi}{2}, \frac{\pi}{2}]$).

We have studied two different cases. In the case A, rotation has been considered to be zero, and none line being significant. In the case B the orientation of the 3D lines is given to the algorithm to compute the camera rotation. This rotation is compensated to estimate translation direction with the voting process.

Table I shows the motion direction obtained in each case. ψ_t, θ_t corresponds with the center of the rectangle given as solution (white boxes in Fig. 6), whose sides are $\Delta\psi_t$ and $\Delta\theta_t$. In the case B we also estimate the rotation velocity of the camera. All the values are expressed in degrees. Although the commanded rotation is zero, the camera rotates a little because it is moved by an angular robot. This rotation makes worse the solution in the case A. The estimation of rotation obtained from the significant lines in the case B gives a better estimation of the flow due to translation and therefore the motion direction is obtained in a shorter range. It turns out interesting to point out that the motion direction obtained in the case B is quite similar to that obtained knowing the depth of the lines [11].

The previous scene turns out appropriate to easily check goodness of the method, and to carry out systematic experiments. However the method works better with many lines of different orientations which are on the whole image. This effect is shown in Fig. 6 (case A), where the solution is over the edge of the pyramid which has not been used.

We now present the experiments with a scene of the laboratory, where the mean depth is about 7 meters (Fig. 7). The camera translates 20 mm in the direction of the focal axis towards the scene.

From the image, 60 lines have been automatically extracted (Fig. 7). As an example, seven of these lines have been manually selected, because they are significant, and therefore their 3D orientation is known. In every test we have considered the limits $D_{min} = 0$ and $D_{max} = 1000(|t|)$. As there are many lines, the set of solutions is small in all the cases, and they are not presented in the results. We only give the motion direction corresponding to the center of the set of solutions. To test the robustness of the method we have acquired ten images in each camera position to obtain mean values and its standard deviation.

We have considered four cases in function of the available information. Although the set-point rotation is zero, there exist a little rotation to be compensated, which can be measured from the robot controller or computed from the line flow. The cases considered are:

- *Case A.* Camera rotation is unknown and none of the lines is significant.
- *Case B.* Camera rotation is unknown but there are some significant lines, which are used to compute the rotation of the camera from the flow.

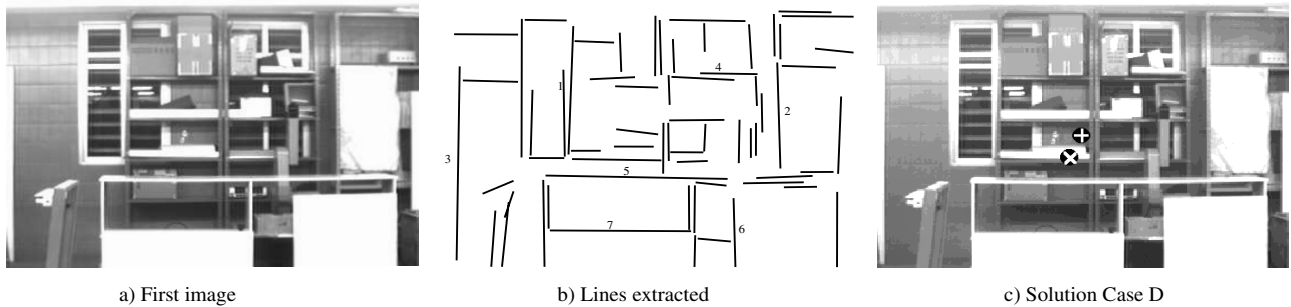


Fig. 7. a) First image of the laboratory, the second one has a disparity less than a pixel. b) Lines extracted in first image (gradient greater than 8 glu/pix. and length bigger than 25 pix). The lines with a number are the significant ones which are supposed to be parallel to the image plane. c) Example of the motion direction (\times) computed in the case D superimposed to the image. The cross (+) represent the commanded motion.

	W_x	W_y	W_z	σW_x	σW_y	σW_z
SetPoint	0.0	0.0	0.0	-	-	-
Case A	0.0	0.0	0.0	-	-	-
Case B	-2.18E-1	-8.02E-2	-4.01E-2	8.6E-3	3.2E-3	1.7E-3
Case C, D	-1.44E-1	-4.39E-2	-5.50E-2	-	-	-

TABLE II
ROTATION COMPENSATED FOR EACH CASE.

- *Case C.* Camera rotation is obtained from the robot controller. None line is significant.
- *Case D.* Camera rotation is obtained from the robot controller. The 3D orientation of the significant lines is also known and used to compute a sole parameter of line flow as explained in §III-B.

In Table II we show (in degrees) the rotation which is compensated in each case. In the case B, as the rotation is estimated from line flow, we present the mean values and their standard deviation. In Table III we present mean values ($\bar{\psi}_t$, $\bar{\theta}_t$) and standard deviation (σ_{ψ_t} , σ_{θ_t}) of the computed motion direction. We can observe that:

- The solution is better and more stable when the knowledge about camera rotation and 3D orientation of the lines increases.
- The use of the rotation obtained from the flow of several lines gives better results that considering the rotation commanded to the robot (case B better than case A).
- The rotation measured from the robot controller gives better results than the obtained from the optical flow using the significant lines (cases C,D better than case B).
- The flow is better computed when we use the 3D orientation of the lines. This can be seen when comparing the case C and the case D whose sole difference is the use (in the case D) of the 3D orientation of the significant lines to estimate a unique flow parameter (t_{nl}) for each line.

We have also evaluated the relevance of taking a value of uncertainty of the rotation Δ_w to apply the visibility constraint as proposed in §IV-C. As conclusion we state that it is better to consider a minimum value of Δ_w to forbid the use of the basic visibility constraint in the straight edges with small projected motion. The use of zero values of Δ_w gives bad results. This has been observed especially when there exist worse knowledge of the rotation (cases A and B). It is not so critical when there exist good knowledge of real rotation (cases C and D). This bound Δ_w can be tuned in each application in function of the rotation uncertainty. When the rotation is computed from the normal flow of the significant lines, the rotation uncertainty can be estimated from the residual of the least squares used to compute the rotation [11]. It has been probed in the case B that Δ_w obtained from the residual gives good results. Only manual tuning made ad-hoc has given better results.

	$\bar{\psi}_t$	$\bar{\theta}_t$	σ_{ψ_t}	σ_{θ_t}
SetPoint	0.0	0.0	-	-
Case A	-19.45	-13.81	9.85	2.41
Case B	10.94	8.38	6.03	1.38
Case C	-2.06	1.93	5.93	2.36
Case D	1.40	0.86	1.02	1.14

TABLE III
MOTION DIRECTION USING THE LABORATORY SCENE.

Further remarks

The visibility constraint is very conditioned by the direction of the normal flow, and therefore by the normals \mathbf{n} of the straight edges. Therefore, the accuracy of its computation turns out crucial for the method. In this work we have exploited topological properties of the edge elements in the line support region. Some advantages of our proposal in comparison with those using isolated edge elements or normal flow measures are:

- Orientation of the straight edge is globally and more accurately obtained.
- Normal flow of the line is also globally obtained.
- We avoid the points with little gradient where the brightness constraint is not valid, and we also avoid points with high curvature where some spurious may appear due to changes in the direction of the gradient.
- Lines with known 3D orientation can be explicitly considered to obtain better and more general results.
- The computational cost of the voting process is reduced using lines. Comparing with the voting at the level of edge elements, the reduction factor is approximately from 1 to $\frac{nm}{2}$, being (n, n) the size of the accumulation matrix and m the mean number of edge elements into a segment.

However, as the description of the scene is less dense because of the use of lines, it may turn out the estimation to be less accurate in scenes where lines are not plentiful. Besides that, our method has the drawbacks of optical flow based methods. Thus constancy of brightness is needed, and little motion between images must be involved.

VI. CONCLUSIONS

The method proposed in this paper uses the visibility constraint to estimate the motion direction when the translation effect on the image is higher than the rotation uncertainty. Nevertheless, the rotation velocity can also be estimated from the flow of at least three significant lines.

In our proposal, 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 and brightness information in motion computation. Differing to other works we have considered the flow globally in the regions supporting straight edges, establishing the visibility constraint for collinear edge elements. We have also introduced the 3D orientation of lines when available, taking into account the effect of the rotation uncertainty.

The experiments have shown good results when there is short rotation uncertainty and little motion between images, being better when some information about the line orientation is available.

REFERENCES

- [1] J.K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images - a review," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 917–935, 1988.
- [2] Y. Liu and T.S. Huang, "Determining straight line correspondences from intensity images," *Pattern Recognition*, vol. 24, no. 6, pp. 489–504, 1991.
- [3] T.S. Huang and A. N. Netravali, "Motion and structure from feature correspondences: A review," *Proceedings of the IEEE*, vol. 82, no. 2, pp. 252–268, 1994.
- [4] H. Kollnig and H.H. Nagel, "3d pose estimation by directly matching polyhedral models to gray value gradients," *International Journal of Computer Vision*, vol. 23, no. 3, pp. 283–302, 1997.
- [5] S. Negahdaripour, "Revised definition of optical flow: Integration of radiometric and geometric cues for dynamic scene analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 205, no. 9, pp. 961–979, 1998.
- [6] B.K.P. Horn, *Robot Vision*, MIT Press, Cambridge, Mass., 1986.
- [7] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Systems and experiment performance of optical flow techniques," *Int. Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.
- [8] 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*, vol. 15, no. 5, pp. 434–447, 1993.
- [9] D.W. Murray and B.F. Buxton, *Experiments in the Machine Interpretation of Visual Motion*, The MIT Press, Massachusetts, 1990.
- [10] J.J. Guerrero and C. Sagüés, "Direct method to obtain straight edge depth from motion," *Optical Engineering*, vol. 37, no. 7, pp. 2124–2132, 1998.
- [11] J.J. Guerrero and C. Sagüés, "Camera motion from brightness on lines. combination of features and normal flow," *Pattern Recognition*, vol. 32, no. 2, pp. 203–216, 1999.
- [12] J.B. Burns, A.R. Hanson, and E.M. Riseman, "Extracting straight lines," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, no. 4, pp. 425–455, 1986.
- [13] Bernd Jähne, *Digital Image Processing*, Springer-Verlag, Berlin-Heidelberg, 1993.
- [14] B.K.P. Horn and E.J. Weldon, "Direct methods for recovering motion," *International Journal of Computer Vision*, , no. 2, pp. 51–76, 1988.
- [15] Y. Aloimonos and Z. Duric, "Estimating the heading direction using normal flow," *Int. Journal of Computer Vision*, vol. 13, no. 1, pp. 33–56, 1994.
- [16] D. Sinclair, A. Blake, and D. Murray, "Robust estimation of egomotion from normal flow," *International Journal of Computer Vision*, vol. 13, no. 1, pp. 57–69, 1994.
- [17] S. Negahdaripour, "Direct computation of the flow with confidence measures," *Computer Vision and Image Understanding*, vol. 64, no. 3, pp. 323–350, 1996.
- [18] 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*, vol. 16, no. 4, pp. 430–438, 1994.
- [19] 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*, Berlin, May. 1993, pp. 591–598.

- [20] 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*, vol. 2, pp. 356–370, 1991.
- [21] C.A. Rothwell, J.L. Mundy, W. Hoffman, and V.D. Nguyen, "Driving vision by topology," in *International Symposium on Computer Vision*, Coral-Gables, Florida, Nov. 1995, pp. 395–400.
- [22] J. Illingworth and J. Kittler, "The adaptive hough transform," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 690–700, 1987.