On the information contained in the motion field pf lines and the cooperation between motion and stereo

Olivier Faugeras, Nassir Navab, Rachid Deriche

INRIA

UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 1453

Programme 4
Robotique, Image et Vision

ON THE INFORMATION CONTAINED IN THE MOTION FIELD OF LINES AND THE COOPERATION BETWEEN MOTION AND STEREO

Olivier D. FAUGERAS
Nassir NAVAB
Rachid DERICHE

Juin 1991

RR_1453

On the Information contained in the Motion Field of Lines and the cooperation between Motion and Stereo [1]

L'information contenue dans le champ de vitesses des droites et la coopération stéréo-mouvement [2]

Olivier Faugeras faugeras@mirsa.inria.fr
Nassir Navab navab@mirsa.inria.fr
Rachid Deriche der@mirsa.inria.fr

INRIA Sophia Antipolis
2004 Route des Lucioles
06565 Valbonne Cedex
FRANCE

[1] This work was supported in part by Esprit projects P2502 et BRA Insight
[2] Ce travail a été partiellement accompli dans le cadre des projets européens Esprit P2502 et BRA Insight

Abstract

We investigate the relationships that exist between the three-dimensional structure and kinematics of a line moving rigidly in space and the two-dimensional structure and kinematics (motion field) of its image in one or two cameras. We establish the fundamental equations that relate its three-dimensional motion to its observed image motion. We show how this motion field can be estimated from a line-based token tracker. We then assume that stereo matches have been established between image segments and show how the estimation of the motion field in the two images can be used to compute part of the kinematic screw of the corresponding 3D line. The equations are linear and if several lines belong to the same object provide a very simple way to estimate the full kinematic screw of that object. Finally we show how the motion field can constrain the stereo matches by establishing necessary conditions that must be satisfied by the motion field of segments which are images of lines belonging to the same object. The idea is based upon the notion of invariant. Part of this theory has been implemented yet. The part which has been uses Kalman filtering. Several experimental results using synthetic and real data are presented.

Key words: Motion field, Line tracking, Structure from Motion, Stereo matching, Egomotion estimation, Stereo-Motion Cooperation cooperation

Résumé

Nous avons étudié les relations qui existent entre d'une part la structure et la cinématique d'une droite en déplacement rigide dans l'espace, et d'autre part la structure et la cinématique (champ des vitesses) de son image dans une ou deux caméras. Les équations fondamentales qui relient son mouvement tridimensionnel au mouvement observé dans l'image ont été établies, ainsi que l'équation du champ des vitesses d'une droite. Nous avons montré comment ce champ des vitesses peut être estimé à l'aide d'une procédure de suivi de droites. Supposant ensuite que les mise en correspondances stéréo sont disponibles, nous montrons comment l'estimations de champ des vitesses peuvent être utilisées pour obtenir certains composants des parametres cinématiques de droite 3D. Les équations sont linéaires, et si plusieurs droites appartiennent au même objet, nous permettent d'obtenir tous les parametres cinématiques de l'objet. Nous proposons enfin une méthode pour guider le processus de mise en correspondance stéréo en utilisant le champ de vitesses. L'idée est fondée sur la notion d'invariant. Une partie de nos algorithmes ont été implémentés en utilisant le filtre de Kalman. Nous les avons testés sur des données synthétiques, puis réelles.

Mot clés: Champ des vitesse, Suivi des droites 2D, Structure à partir de mouvement, Appariement stéréo, Estimation de mouvement, Coopération stéréo-mouvement

Contents

1 Introduction

In stereo and motion analysis, many studies have been done using two or three static cameras or a sequence of monocular images obtained by a moving camera. Recently, researchers have begun to combine these two approaches, which have their own shortcomings and advantages, to find faster and more robust algorithms (see e.g. [KA87], [FDN89], and [YC90]). We believe in the potential efficiency of stereo-motion cooperation and this paper is another attempt to explore this idea.

We present an investigation of the relationships that exist between the three-dimensional structure and kinematics of a line moving rigidly in space and the two-dimensional structure and kinematics (motion field) of its image in one or two cameras.

To extract 3D information, from real images, "meaningful" features, such as corner points, edges, regions, etc., are often used to reduce the cost of computation and matching ambiguities. In this paper, we use the lines which support the line segments obtained by an edge detector.

Using lines as tokens has been found very useful in Vision. Edge segments can be reliably extracted from images through edge detection [Can86,Der87], and polygonal approximation [Ber86], and form a robust source of tokens. Using lines of support yields parameters less sensitive to noise and therefore leads to more reliable solutions of the structure and motion equations (see for example the uncertainty analysis on the line segment representation done in [DF90]). Line segments are present in many real-world environments such as : cities, factories, roads, inside of buildings, etc. Therefore the lines motion field has great potential applications to, for example, real time navigation, security patrolling, dynamic tracking, etc.

Of course, both points and lines could be used, when available, thereby increasing the overall robustness of the system.

Faugeras, Lustman and Toscani [FLT87], investigated the problem of camera displacement estimation and recovery of structure from two views when point matches were available and from three views when line matches were available.

Contrarily to that work which dealt with discrete questions, in [FDN89] and [NDF90], we considered the continuous case and investigated some theoretical points about the recovery of structure and motion of 3D lines. We push this approach further here.

The reader interested in related approaches in the discrete case is also referred to [Lon81,Lon84,LH86a,LH86b,SA90]. To our knowledge, the question of the motion field of lines has not been studied so far.

The implementation of the work described here is built on two previous pieces of work. The first one is a segment based trinocular stereo program that was developed

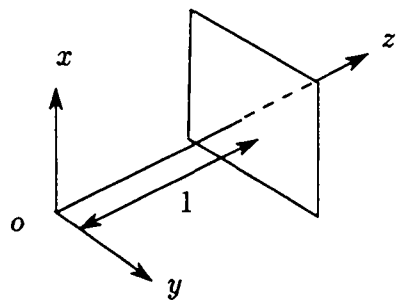

Figure 1: Pinhole model of a camera

by Ayache and Lustman [AL87]. The second one is a program for tracking line segments in sequences of images developed by Deriche et al. [TDF88,DF90].

For our present purposes, it is sufficient to say that all programs work on the same image description, i.e polygonal approximations of connected sets of edge pixels.

The stereo program matches them between three images acquired simultaneously to avoid problems related to motion and reconstructs 3D line segments.

The token tracker program matches them between images taken at consecutive instants and builds a model of their 2D motion.

2 Preliminaries

Vectors are represented in bold face, i.e $\mathbf{x}$, transposition of vectors and matrixes is indicated by ^{T}, i.e $\mathbf{x}^T$. To distinguish between affine and projective representations of points, we will note m a 2D point, $\mathbf{m} = (x, y, 1)^T$ its affine two-dimensional representation and $\mathring{\mathbf{m}} = (X, Y, T)^T$ its projective representation. 3D points M are represented by vectors $\mathbf{M} = (X, Y, Z)^T$. For a given three-dimensional vector $\mathbf{x}$ we also use $\tilde{\mathbf{x}}$ to represent the 3×3 antisymmetric matrix such that $\tilde{\mathbf{x}}\mathbf{y} = \mathbf{x} \wedge \mathbf{y}$ for all vectors $\mathbf{y}$. $\mathbf{I}_3$ represents the 3×3 identity matrix.

We model our camera with the standard pinhole model of figure 1 and assume that everything is referred to the camera standard coordinate frame $(oxyz)$. We know from work on calibration [Tsa86,FT86] that it is always possible, up to a very good approximation, to go from the real pixel values to the standardized values x and y.

2.1 3D line representation

We represent a straight line D in 3D space with a direction vector $\mathbf{l} = (\alpha, \beta, \gamma)^T$ and a point M of coordinates (x, y, z) which we leave unspecified at the moment. $\mathbf{l}$ is defined up to a scale factor.

2.2 2D line representation

A line d in the retinal plane is represented by a vector $\mathbf{n} = (u, v, w)^T$ giving its equation:

$$ux + vy + w = 0 \tag{1}$$

The interpretation of $\mathbf{n}$ is that it is the normal to the plane defined by the 2D line and the optical center of the camera (see figure 2); note that this plane contains also the correspondig 3D line, D. To simplify equations, we assume that $\mathbf{n}$ is of unit norm $\|\mathbf{n}\| = 1$. This leaves $\mathbf{n}$ unspecified up to its sign. This is resolved by the fact that image lines are the supports of measured line segments, corresponding to edges, which are oriented by the intensity contrast across the edge. It is a matter of convention to orient the segments in such a way that, for example, the intensity increases in the direction at +90 degrees.

2.3 Relations between D and d

We now establish the relationships between the representation of a 3D line and its image in the retinal plane. The first relation is that the direction $\mathbf{l}$ of the line is perpendicular to the normal $\mathbf{n}$ to the plane it defines with the optical center.

$$\mathbf{l}^T\mathbf{n} = 0 \tag{2}$$

We also need a point. We can choose any point M of D. Let us denote by X_m, Y_m and Z_m its coordinates. Let us recall that the transformation between a point M in 3D space and its retinal image m is linear in projective coordinates

$$\check{\mathbf{m}} = \mathcal{P}\check{\mathbf{M}} \tag{3}$$

in this equation $\check{\mathbf{m}}$ and $\check{\mathbf{M}}$ are projective representations of points m and M, i.e 3 × 1 and 4 × 1 vectors, respectively ; $\mathcal{P}$ is a 3 × 4 matrix which, with our choice of coordinates, has the very simple form :

$$\mathcal{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4}$$

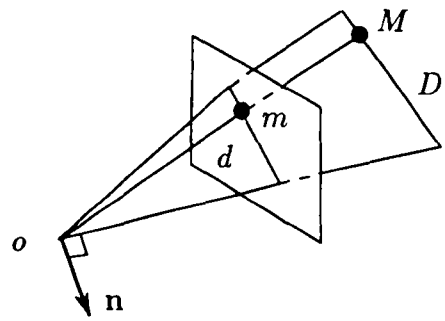

Figure 2: The vector $\mathbf{n}$

Through relation (3) the point M is mapped on the retina on to point m of projective coordinates $(X_m, Y_m, Z_m)^T = \breve{\mathbf{m}}$ or, equivalently, if $Z_m \neq 0$, $(\frac{X_m}{Z_m} = x_m, \frac{Y_m}{Z_m} = y_m, 1)^T = \mathbf{m}$.

The relationship between $\mathbf{n}$, the normal to the plane defined by the optical center and the 3D line segment, and the 3D point M is given by the following equation :

$$\mathbf{M}^T\mathbf{n} = 0 \tag{5}$$

3 The motion equations for lines

We are dealing with the case where the token tracker (see [TDF88,DF90]) tracks in the image of a camera a number of line segments whose supporting lines are represented by the vectors $\mathbf{n}$. From this tracking, we can estimate the time derivative $\dot{\mathbf{n}}$, of those vectors (see section 5). The vector $\dot{\mathbf{n}}$ is the motion field of the corresponding line. If we have two or three cameras, this tracking process can be achieved independently for each image and the question we ask and answer in the next section is, how much information related to the 3D motion and structure of the rigid objects observed in the scene can be recovered from these quantities.

3.1 Recovering the direction of a 3D line from its motion field and the 3D angular velocity Ω.

In order to gain more insight into the problem, we now assume that the 3D line under consideration is attached to a rigid body whose motion is described by its

instantaneous angular velocity, $\mathbf{\Omega}$, and linear velocity $\mathbf{V}$, its kinematic screw at the origin o. Let $\mathbf{L}$ be a normalized line direction:

$$\mathbf{L} = \frac{\mathbf{l}}{\|\mathbf{l}\|}$$

Note that $\mathbf{L}$ satisfies also (2).

We know that the velocity $\dot{\mathbf{M}}$ of any point $\mathbf{M}$ attached to the rigid body is given by

$$\dot{\mathbf{M}} = \mathbf{V} + \mathbf{\Omega} \wedge \mathbf{M} \tag{6}$$

The normalized direction $\mathbf{L}$ satisfies a simpler differential equation:

$$\dot{\mathbf{L}} = \mathbf{\Omega} \wedge \mathbf{L} \tag{7}$$

We now express $\mathbf{l}$ as function only of the image measurements $\mathbf{n}$, $\dot{\mathbf{n}}$, and $\mathbf{\Omega}$. For this we can use equation (2) and take its time derivative

$$\dot{\mathbf{n}}^T\mathbf{L} + \mathbf{n}^T\dot{\mathbf{L}} = 0$$

Replacing $\dot{\mathbf{L}}$ in that equation by its value from equation (7), we obtain:

$$\dot{\mathbf{n}}^T\mathbf{L} + (\mathbf{n}, \mathbf{\Omega}, \mathbf{L}) = 0$$

where $(\mathbf{n}, \mathbf{\Omega}, \mathbf{L})$ is the determinant of the three vectors $\mathbf{n}$, $\mathbf{\Omega}$, and $\mathbf{L}$. Using standard properties of determinants, this equation can be rewritten as

$$(\dot{\mathbf{n}} + \mathbf{n} \wedge \mathbf{\Omega})^T\mathbf{L} = 0 \tag{8}$$

Combining it with equation (2) shows that $\mathbf{L}$ is perpendicular to both vectors $\mathbf{n}$ and $\dot{\mathbf{n}} + \mathbf{n} \wedge \mathbf{\Omega}$, therefore it is proportional to their cross-product; since $\mathbf{l}$ is a direction, i.e is defined up to a scale factor, we write

$$\mathbf{l} = \mathbf{n} \wedge (\mathbf{n} \wedge \mathbf{\Omega} + \dot{\mathbf{n}}) \tag{9}$$

Equation (9) is fundamental. It allows to recover the spatial orientation of a 3D line from the knowledge of $\mathbf{\Omega}$, the representation $\mathbf{n}$ of its 2D image and the motion field $\dot{\mathbf{n}}$ which can be estimated, as shown later, by the token tracker.

3.2 Recovering a point of the 3D line from its motion field $\dot{\mathbf{n}}$, and its kinematic screw $(\Omega, \mathbf{V})$

The velocity $\dot{\mathbf{M}}$ of any point $\mathbf{M}$ attached to the rigid 3D line is given by equation (6). On the other hand the relationship between $\mathbf{n}$, the representation of the 2D line and the 3D point M is given by the equation (5). Taking the time derivative of (5) yields

$$\dot{\mathbf{n}}^T\mathbf{M} + \mathbf{n}^T\dot{\mathbf{M}} = 0$$

Replacing $\dot{\mathbf{M}}$ in that equation by its value obained from equation (6), we write:

$$(\Omega \wedge \mathbf{M} + \mathbf{V})^T\mathbf{n} + \mathbf{M}^T\dot{\mathbf{n}} = 0$$

which can be rewritten as

$$\mathbf{M}^T(\mathbf{n} \wedge \Omega + \dot{\mathbf{n}}) = -\mathbf{V}^T\mathbf{n}$$

From equation (3), we have

$$\mathbf{M} = Z_m\mathbf{m} \tag{10}$$

This leads to the following relation that gives Z_m as a function of Ω , $\mathbf{V}$, $\mathbf{n}$, and the motion field $\dot{\mathbf{n}}$ obtained from the line segments tracker.

$$Z_m\mathbf{m}^T(\mathbf{n} \wedge \Omega + \dot{\mathbf{n}}) = -\mathbf{V}^T\mathbf{n} \tag{11}$$

A complete determination of the 3D position of the point $\mathbf{M}$ is then obtained through relation (10).

4 Recovering Ω and $\mathbf{V}$ from stereo and the motion field

Suppose now that we have a second camera of known position and orientation with respect to the first one, as shown in figure 3. Events or measurements in the first camera will be sometime indexed by 1 while events in the second camera will be indexed by 2 or followed by '.

If a line segment represented by $\mathbf{n}_1$ is matched by the stereo program to a line segment represented by $\mathbf{n}_2$ in the second image at time t, we can reconstruct a 3D line. We show that this yields four linear equations in Ω and $\mathbf{V}$.

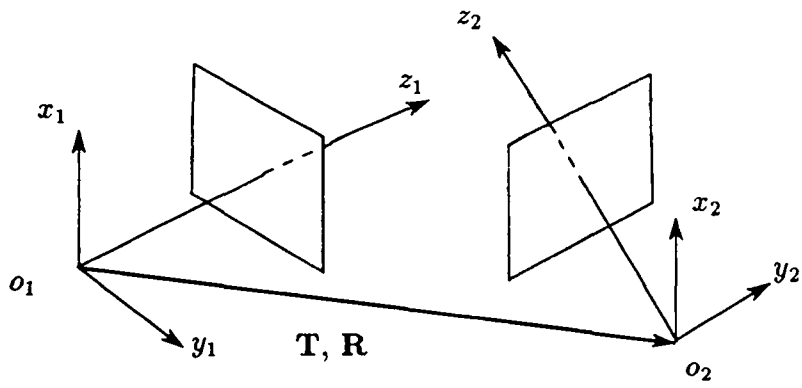

Figure 3: A stereo rig

Therefore, if we assume that we have identified two segments located on the same rigid object, we can in general compute its rigid motion from the eight equations. Also, if the motion of the segment observed at time t has constant velocity (constant $\mathbf{\Omega}$ and $\mathbf{V}$), then two observations of the same segment at two consecutive time instants will permit, in general, to compute $\mathbf{\Omega}$ and $\mathbf{V}$.

We will be using the relationship between the representation $\mathbf{M}$ of a point M in the coordinate system of the first camera and its representation $\mathbf{M}'$ in the coordinate system of the second camera

$$\mathbf{M} = \mathbf{R}\mathbf{M}' + \mathbf{T} \tag{12}$$

We will also be using the relationship between the kinematic screw written in the second camera coordinates ($\mathbf{\Omega}'$,$\mathbf{V}'$) and the first camera coordinates ($\mathbf{\Omega}$,$\mathbf{V}$). We have:
:

$$\mathbf{\Omega}' = \mathbf{R}^T\mathbf{\Omega} \tag{13}$$

$$\mathbf{V}' = \mathbf{R}^T(\mathbf{V} + \mathbf{\Omega} \wedge \mathbf{T}) \tag{14}$$

Proof : we write the motion equation (6) of the 3D point M in the first and second camera coordinate systems

$$\dot{\mathbf{M}} = \mathbf{V} + \mathbf{\Omega} \wedge \mathbf{M} \tag{15}$$

$$\dot{\mathbf{M}}' = \mathbf{V}' + \mathbf{\Omega}' \wedge \mathbf{M}' \tag{16}$$

then we replace $\mathbf{M}$ by its value in equation (12), we obtain:

$$\mathbf{R}\dot{\mathbf{M}}' = \mathbf{V} + \mathbf{\Omega} \wedge (\mathbf{R}\mathbf{M}' + \mathbf{T})$$

which leads us to:

$$\dot{\mathbf{M}}' = \mathbf{R}^T\mathbf{V} + \mathbf{R}^T\boldsymbol{\Omega} \wedge (\mathbf{M}' + \mathbf{R}^T\mathbf{T})$$

or

$$\dot{\mathbf{M}}' = \mathbf{R}^T(\mathbf{V} + \boldsymbol{\Omega} \wedge \mathbf{T}) + (\mathbf{R}^T\boldsymbol{\Omega}) \wedge \mathbf{M}'$$

Identifying the last equation with equation (16) one obtains (13) and (14), QED. □

4.1 Recovering $\boldsymbol{\Omega}_\perp$

Let us analyze equations (9) and (11). From equation (9), we can verify that for a given line direction, the rotation can only be recovered in the direction of the perpendicular from the origin to the line.

Indeed, let us add $\lambda\mathbf{l}$ to $\boldsymbol{\Omega}$:

$$\mathbf{n} \wedge (\mathbf{n} \wedge (\boldsymbol{\Omega} + \lambda\mathbf{l}) + \dot{\mathbf{n}}) = \mathbf{n} \wedge (\mathbf{n} \wedge \boldsymbol{\Omega} + \dot{\mathbf{n}}) + \lambda\mathbf{n} \wedge (\mathbf{n} \wedge \mathbf{l})$$

The last term is proportional to $\mathbf{l}$ since $\mathbf{n}$ is orthogonal to $\mathbf{l}$ therefore (9) is unchanged.

This proves that we cannot recover the component of $\boldsymbol{\Omega}$ along $\mathbf{l}$. If we add $\lambda\mathbf{n}$ to $\boldsymbol{\Omega}$ it is clear that (9) remains invariant, therefore we cannot either recover the component of $\boldsymbol{\Omega}$ along $\mathbf{n}$.

This is if we have only one camera. If we have two, the information that can be recovered from (9) is the projection of $\boldsymbol{\Omega}$ in a plane perpendicular to $\mathbf{l}$, we note it $\boldsymbol{\Omega}_\perp$.

Assuming that we know from stereo which segment in the second image matches the one in the first image, a similar equation can be written in the coordinate system of this camera :

$$\mathbf{l}' = \mathbf{n}' \wedge (\mathbf{n}' \wedge \boldsymbol{\Omega}' + \dot{\mathbf{n}}') \tag{17}$$

If $\mathbf{R}$ is the rotation matrix from the first camera to the second, equation (17) can be written in the first coordinate system by applying $\mathbf{R}$ to it :

$$\mathbf{R}\mathbf{l}' = \mathbf{R}[\mathbf{n}' \wedge (\mathbf{n}' \wedge \boldsymbol{\Omega}' + \dot{\mathbf{n}}')] = \mathbf{R}\mathbf{n}' \wedge (\mathbf{R}\mathbf{n}' \wedge \boldsymbol{\Omega} + \mathbf{R}\dot{\mathbf{n}}')$$

Letting $\mathbf{p} = \mathbf{R}\mathbf{n}'$, a direction of the line is given by $\mathbf{L} = \frac{\mathbf{n}\wedge\mathbf{p}}{\|\mathbf{n}\wedge\mathbf{p}\|}$ in the first coordinate system.

There are three "natural" coordinate systems in which we can write $\boldsymbol{\Omega}$. They are $(\mathbf{n}, \mathbf{n} \wedge \mathbf{L}, \mathbf{L})$, $(\mathbf{p}, \mathbf{p} \wedge \mathbf{L}, \mathbf{L})$, and $(\mathbf{n}, \mathbf{p}, \mathbf{L})$, the first two being orthonormal.

Let us use the first coordinate system

$$\boldsymbol{\Omega} = a\mathbf{n} + b\mathbf{L} + c\mathbf{n} \wedge \mathbf{L}$$

Replacing $\mathbf{\Omega}$ by this value in (9) yields

$$\mathbf{l} = -b\mathbf{L} - c\mathbf{n} \wedge \mathbf{L} + \mathbf{n} \wedge \dot{\mathbf{n}}$$

Taking the inner product of both sides with $\mathbf{n} \wedge \mathbf{L}$ yields c which is equal to $\mathbf{\Omega}^T(\mathbf{n} \wedge \mathbf{L})$:

$$\mathbf{\Omega}^T(\mathbf{n} \wedge \mathbf{L}) = (\mathbf{n} \wedge \dot{\mathbf{n}})^T(\mathbf{n} \wedge \mathbf{L}) = \dot{\mathbf{n}}^T\mathbf{L} \tag{18}$$

The second camera yields a similar equation

$$\mathbf{\Omega}^T(\mathbf{p} \wedge \mathbf{L}) = (\mathbf{p} \wedge \dot{\mathbf{p}})^T(\mathbf{p} \wedge \mathbf{L}) = \dot{\mathbf{p}}^T\mathbf{L} \tag{19}$$

Let us now use the third coordinate system.

$$\mathbf{\Omega} = \alpha\mathbf{n} + \beta\mathbf{p} + \gamma\mathbf{L} \tag{20}$$

Let us replace $\mathbf{\Omega}$ by this value in equation (9)

$$\mathbf{l} = \beta((\mathbf{n}^T\mathbf{p})\mathbf{n} - \mathbf{p}) - \gamma\mathbf{L} + \mathbf{n} \wedge \dot{\mathbf{n}}$$

Taking the inner product of both sides with $\mathbf{p}$ yields

$$\beta = \frac{-\dot{\mathbf{n}}^T\mathbf{L}}{\|\mathbf{n} \wedge \mathbf{p}\|}$$

In a similar way, we can use the equation :

$$\mathbf{R}\mathbf{l}' = \mathbf{p} \wedge (\mathbf{p} \wedge \mathbf{\Omega} + \dot{\mathbf{p}})$$

to derive:

$$\alpha = \frac{\dot{\mathbf{p}}^T\mathbf{L}}{\|\mathbf{n} \wedge \mathbf{p}\|}$$

Replacing α and β in (20) by their respective expressions, we obtain

$$\mathbf{n}^T\mathbf{\Omega} = (\dot{\mathbf{p}} - (\mathbf{n}^T\mathbf{p})\dot{\mathbf{n}})^T\frac{\mathbf{L}}{\|\mathbf{n} \wedge \mathbf{p}\|} \tag{21}$$

$$\mathbf{p}^T\mathbf{\Omega} = ((\mathbf{p}^T\mathbf{n})\dot{\mathbf{p}} - \dot{\mathbf{n}})^T\frac{\mathbf{L}}{\|\mathbf{n} \wedge \mathbf{p}\|} \tag{22}$$

Of course the two pairs of equations (18), (19), and (21) and (22) are equivalent in the sense that they both allow us to recover the projection of the angular velocity in the plane perpendicular to the direction of the line from the optical flows of its images in the two cameras.

4.2 Recovering $\mathbf{V}_\perp$

In this section, we suppose that we have calculated the depth of a point m of a segment (for example its midpoint) by triangulation. This is possible since we have a calibrated stereo rig and a match between two segments. Let M be the corresponding 3D point. We write (11) in the two coordinate systems of the stereo rig. In the first coordinate system we obtain

$$\mathbf{M}^T(\mathbf{n} \wedge \boldsymbol{\Omega} + \dot{\mathbf{n}}) = -\mathbf{V}^T\mathbf{n} \tag{23}$$

This equation is linear in $\boldsymbol{\Omega}$ and $\mathbf{V}$. Note that in the term $\mathbf{n} \wedge \boldsymbol{\Omega}$ appears the component of $\boldsymbol{\Omega}$ along $\mathbf{n} \wedge \mathbf{L}$ that we can estimate through equation (18) and the its component along $\mathbf{L}$ which, as shown in the previous section, cannot be recovered from a single pair of image segments.

We obtain a second equation by writing (23) in the coordinate system of the second camera

$$\mathbf{M}'^T(\mathbf{n}' \wedge \boldsymbol{\Omega}' + \dot{\mathbf{n}}') = -\mathbf{V}'^T\mathbf{n}'$$

Using (12), (13), and (14) we obtain a second linear equation in $\mathbf{V}$ and $\boldsymbol{\Omega}$

$$\mathbf{M}^T(\mathbf{p} \wedge \boldsymbol{\Omega} + \dot{\mathbf{p}}) = -\mathbf{V}^T\mathbf{p} + \mathbf{T}^T\dot{\mathbf{p}} \tag{24}$$

(23) and (24) are two linear constraints on $\boldsymbol{\Omega}$ and $\mathbf{V}$. Together with (18) and (19) or (21) and (22) they do not allow us to recover the full kinematic screw. It is only if two non parallel 3D segments are available that this is possible. This is the basis of the method described in the next section.

4.3 Recovering $\boldsymbol{\Omega}$ and $\mathbf{V}$

In sections 4.1 and 4.2 we have shown how to recover the projection in a plane perpendicular to a 3D line of its kinematic screw. In order to achieve this, we have derived four linear equations. Assuming that we have several (at least two) non parallel 3D lines with the same three-dimensional motion, we can use them to recover the full kinematic screw of the rigid object they are attached to. In order to do so we simply need to do a least-squares computation or, since we keep track of uncertainty at all levels, a weighted least-squares approximation which can be implemented with the Kalman filter. Note that in our implementation, the Kalman filter operates both spatially, in the images, and in time, assuming in the current implementation constant velocities, even though more complicated kinematic models are also possible.

Of course, the method only works if all segments considered actually belong to the same rigid object, otherwise it fails. The examples we present later have been obtained with a mobile robot moving in a static environment and therefore this condition holds true.

Here we present a summary of the different steps :

- Cameras calibration: this is an offline process which yields in particular $\mathbf{R}$ and $\mathbf{T}$ [FT86].
- Edge detection and polygonal approximation: after the acquisition of images by the cameras, using an edge detector [Can86,Der87], and a polygonal approximation algorithm [Ber86], we obtain the 2-D line representations $\mathbf{n}$. We use the contrast to define the orientation of these lines and to make sure that a line direction varies smoothly in consecutive frames.
- Stereo matching: the stereo algorithm [AL87] gives us a set of matched segments in two images. Using these two sources of information, we reconstruct a number of 3-D lines by computing their directions $\mathbf{L}$ and the position of an arbitrary point $\mathbf{M}$.
- Line tracking: when the cameras move we track the line segments in each camera [DF90], and we estimate the line motion field $\dot{\mathbf{n}}$ as explained in section 5.
- Ego-motion estimation: using a Kalman filter as we have already explained we estimate the kinematic screw($\mathbf{\Omega}$, $\mathbf{V}$). Using equations (18), (19), (23), and (24), For each segment we obtain the following matrix equation:

$$\mathbf{Z} = \mathbf{A}^T \mathbf{X}$$

where $\mathbf{X} = \begin{bmatrix} \mathbf{\Omega} \\ \mathbf{V} \end{bmatrix}$, in filtering terminology, is the state vector and

$$\mathbf{A} = \begin{bmatrix} \mathbf{n} \wedge \mathbf{L} & \mathbf{p} \wedge \mathbf{L} & \mathbf{M} \wedge \mathbf{n} & \mathbf{M} \wedge \mathbf{p} \\ \mathbf{0} & \mathbf{0} & \mathbf{n} & \mathbf{p} \end{bmatrix}$$

is the transposed of the observation matrix and

$$\mathbf{Z} = \begin{bmatrix} \dot{\mathbf{n}}^T \mathbf{L} & \dot{\mathbf{p}}^T \mathbf{L} & -\mathbf{M}^T \dot{\mathbf{n}} & -\mathbf{M}^T \dot{\mathbf{p}} + \mathbf{T}^T \dot{\mathbf{p}} \end{bmatrix}^T$$

defines our measurement vector. To make it more clear we define the input and output of our system:

a) The 2-D line equations $\mathbf{n}$ (resp. $\mathbf{n}'$ in the other camera), their covariance matrices, and the calibration data $\mathbf{R}$ and $\mathbf{T}$ form our input.

b) the lines motion fields $\dot{\mathbf{n}}$ (resp. $\dot{\mathbf{n}}'$), 3D line directions $\mathbf{L}$ and a point $\mathbf{M}$ on these 3D lines with their corresponding covariance matrices, are then estimated from the input, and all together form a new input to the final system of equations.

c) Finally $\mathbf{\Omega}$ and $\mathbf{V}$ form the output of our system. Using a Kalman filter, we begin with an estimation of $\mathbf{\Omega}$ and $\mathbf{V}$, in our case null vectors, with large covariance matrices. And at each iteration the information in the new segment being processed increases the precision of our output.

We do not describe in this paper the details of the Kalman filter implementation, they can be found, for example, in [AF89]. For more details on Kalman filtering techniques, the reader is referred to the beautiful book by Maybeck [May79].

- Integration of new data: after a few steps, we perform another stereo matching, but only on segments which are not being followed or which just entered our visual field. We add these new pairs of segments to those which are being tracked to complete our 3-D model of environment and to continue to obtain robust estimations of our cameras system ego-motion. So, the system relies on stereo to establish initial matches which are then updated in time by the token tracker program which tracks simultaneously the pairs of matched segments. The Kalman filter is used to update the estimation of the kinematic screw $(\mathbf{\Omega}, \mathbf{V})$.

5 The line motion field $\dot{\mathbf{n}}$

In this section, we establish the line motion field equation as we did in [NDF90]. Since $\|\mathbf{n}\| = 1$, and thus $\mathbf{n}^T\dot{\mathbf{n}} = 0$, there exists a vector ω such that

$$\dot{\mathbf{n}} = \omega \wedge \mathbf{n} \tag{25}$$

The following proposition determines the value of ω.

Proposition 1 *ω in (25) is given by*

$$\omega = \mathbf{\Omega} + (\frac{\mathbf{V}^T\mathbf{n}}{h})\,\mathbf{L} \tag{26}$$

where h is the distance of the 3D line D to the optical center.

Proof : Let us choose M_1 and M_2 two arbitrary points on the 3D line D and define $\mathbf{N} = \mathbf{M}_1 \wedge \mathbf{M}_2$. We then have $\mathbf{n} = \frac{\mathbf{N}}{\|\mathbf{N}\|}$ and also :

$$\dot{\mathbf{n}} = \frac{\partial \mathbf{n}}{\partial \mathbf{N}}\dot{\mathbf{N}} = \frac{\partial \mathbf{n}}{\partial \mathbf{N}}(\dot{\mathbf{M}}_1 \wedge \mathbf{M}_2 + \mathbf{M}_1 \wedge \dot{\mathbf{M}}_2)$$

We know the motion equation of $\mathbf{M}_1$ and $\mathbf{M}_2$:

$$\dot{\mathbf{M}}_i = \boldsymbol{\Omega} \wedge \mathbf{M}_i + \mathbf{V} \quad (i = 1, 2)$$

it is easy to verify that

$$\frac{\partial \mathbf{n}}{\partial \mathbf{N}} = \frac{\mathbf{I}_3}{\| \mathbf{N} \|} - \frac{\mathbf{N}\mathbf{N}^T}{(\mathbf{N}^T\mathbf{N})^{3/2}} = \frac{1}{\|\mathbf{N}\|}(\mathbf{I}_3 - \mathbf{n}\mathbf{n}^T)$$

so that we can write

$$\dot{\mathbf{n}} = \frac{1}{\|\mathbf{N}\|}(\mathbf{I}_3 - \mathbf{n}\mathbf{n}^T)[(\boldsymbol{\Omega} \wedge \mathbf{M}_1 + \mathbf{V}) \wedge \mathbf{M}_2 + \mathbf{M}_1 \wedge (\boldsymbol{\Omega} \wedge \mathbf{M}_2 + \mathbf{V})]$$

If we notice that $\mathbf{I}_3 - \mathbf{n}\mathbf{n}^T = \tilde{\mathbf{n}}^2$ we obtain:

$$\dot{\mathbf{n}} = \tilde{\mathbf{n}}^2(\boldsymbol{\Omega} \wedge \mathbf{n} + \frac{(\mathbf{M}_1 - \mathbf{M}_2) \wedge \mathbf{V}}{\|\mathbf{N}\|})$$

Noticing that $\frac{\mathbf{M}_1 - \mathbf{M}_2}{\|\mathbf{M}_1 - \mathbf{M}_2\|} = \mathbf{L}$ and that $\frac{\|\mathbf{N}\|}{\|\mathbf{M}_1 - \mathbf{M}_2\|} = h$, the distance from the origin to the 3D line, we rewrite this as

$$\dot{\mathbf{n}} = \tilde{\mathbf{n}}^2(\boldsymbol{\Omega} \wedge \mathbf{n} + \frac{\mathbf{L} \wedge \mathbf{V}}{h})$$

It is easy to verify that:

$$\tilde{\mathbf{n}}^2(\boldsymbol{\Omega} \wedge \mathbf{n}) = \boldsymbol{\Omega} \wedge \mathbf{n}$$

and that

$$\tilde{\mathbf{n}}^2(\mathbf{L} \wedge \mathbf{V}) = (\mathbf{V}^T\mathbf{n})\mathbf{L} \wedge \mathbf{n}$$

from which (26) follows. □

Replacing ω by its value, (25) gives us the line motion field equation:

$$\dot{\mathbf{n}} = (\frac{\mathbf{V}^T\mathbf{n}}{h})\ \mathbf{L} \wedge \mathbf{n} + \boldsymbol{\Omega} \wedge \mathbf{n} \tag{27}$$

It is interesting to compare it with the point motion field equation [May87]. Maybank noted that: " If the image is formed by polar projection on to a spherical projection surface then the velocities $\dot{\mathbf{q}}$ of images points $\mathbf{q}$ are given by:

$$\dot{\mathbf{q}} = [\mathbf{V} - (\mathbf{V}^T\mathbf{q})]K + \boldsymbol{\Omega} \wedge \mathbf{q}$$

Where $1/K$ is the distance in the direction $\mathbf{q}$ from the optical center of the camera to the moving surface." We find similar terms in equation (27), one depending only on image information and angular velocity and the other one on image information,

translational velocity and also on some 3D information (the depth in one case and the depth and the direction in the other one).

In addition, equation (27) tells us all the information one may extract from a 3D line, its image and its motion field. It shows that without the knowledge of depth and translational velocity one may only recover $\mathbf{\Omega}$ in the direction $\mathbf{n} \wedge \mathbf{L}$, which is the direction of the perpendicular from the origin to the line , and that rotation around an axis parallel to the 3D line is coupled with translational velocity. These observations helped us (see section 4) to decompose correctly the problem and to recover the kinematic screw.

When working in the first camera coordinate system, but using two calibrated cameras, we may find h, the perpendicular distance from the origin (assumed to be at the first camera optical center) to the 3D line as a function of images information and calibration data. The direction of this shortest distance is $\mathbf{n} \wedge \mathbf{L}$, the distance d from the origin to the plane containing the 3D line and the optical center of the second camera is $\mid \mathbf{p}^T\mathbf{T} \mid$ since the equation of that plane is $\mathbf{p}^T(\mathbf{M} - \mathbf{T}) = 0$ (The symbols $\mathbf{T}$ and $\mathbf{P} = \mathbf{R}\mathbf{n}'$, that we use here are defined in the previous sections). We must have (see figure 4):

$$\mid \mathbf{p}^T(h\mathbf{n} \wedge \mathbf{L}) \mid = d$$

since $\mathbf{p}^T(\mathbf{n} \wedge \mathbf{L}) = \mathbf{L}^T(\mathbf{p} \wedge \mathbf{n})$ and $\mathbf{L} = \frac{\mathbf{n} \wedge \mathbf{p}}{\|\mathbf{n} \wedge \mathbf{p}\|}$, we find:

$$h = \frac{\mid \mathbf{p}^T\mathbf{T} \mid}{\|\mathbf{p} \wedge \mathbf{n}\|} \tag{28}$$

When estimating the velocity of a vector, we take time samples seperated by time intervals Δt which are small enough so that we can suppose that there is no acceleration during this short time. Dealing with unit vectors moving on the unit sphere, no acceleration implies constant angular velocity. Therefore, to estimate $\dot{\mathbf{n}}$, we assume that ω is constant between t and $t + \Delta t$. We can then integrate the differential equation (25). The solution is:

$$\mathbf{n}_{t+\Delta t} = \mathbf{e}^{\widetilde{\omega}\Delta t}\mathbf{n}_t \tag{29}$$

Remember Rodrigues [Rod40] formula for a given three-dimensional vector $\mathbf{r}$

$$\mathbf{e}^{\widetilde{\mathbf{r}}} = \mathbf{I}_3 + \frac{\sin\theta}{\theta}\widetilde{\mathbf{r}} + \frac{1 - \cos\theta}{\theta^2}\widetilde{\mathbf{r}}^2$$

where $\theta = \|\mathbf{r}\|$. Using this formula, we obtain:

$$\mathbf{n}_{t+\Delta t} = \mathbf{n}_t + \frac{\sin(\theta\Delta t)}{\theta}\widetilde{\omega}\mathbf{n}_t + \frac{1 - \cos(\theta\Delta t)}{\theta^2}\widetilde{\omega}^2\mathbf{n}_t \tag{30}$$

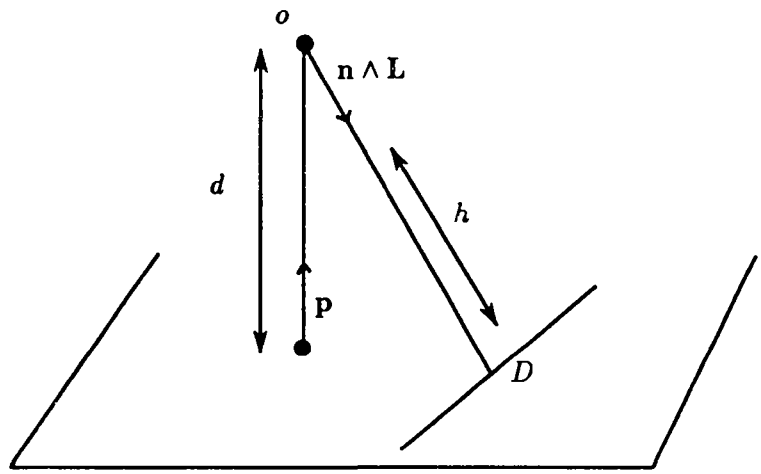

Figure 4: The shortest distance h of the origin o to the 3D line D

One can verify (see appendix) that :

$$\omega = \left(\frac{\Theta(\mathbf{n}_t, \mathbf{n}_{t+\Delta t})}{\Delta t}\right)\frac{\mathbf{n}_t \wedge \mathbf{n}_{t+\Delta t}}{\| \mathbf{n}_t \wedge \mathbf{n}_{t+\Delta t} \|} \tag{31}$$

is a solution of equation (30), where $\Theta(\mathbf{n}_t, \mathbf{n}_{t+\Delta t})$ is the angle between $\mathbf{n}_t$ and $\mathbf{n}_{t+\Delta t}$.

In our experiments we estimated $\dot{\mathbf{n}}$ from equations (25 and 31) with $\Delta t = 1$, using two consecutive images ($\mathbf{n}_{i-1}$ and $\mathbf{n}_i$).

Replacing ω by its value in (25) we obtain

$$\dot{\mathbf{n}}_t = \left(\frac{\Theta(\mathbf{n}_t, \mathbf{n}_{t+\Delta t})}{\Delta t \| \mathbf{n}_t \wedge \mathbf{n}_{t+\Delta t} \|}\right)(\mathbf{n}_{t+\Delta t} - (\mathbf{n}_t^T \mathbf{n}_{t+\Delta t})\mathbf{n}_t)$$

and we notice that the vector $\mathbf{n}_{t+\Delta t} - (\mathbf{n}_t^T \mathbf{n}_{t+\Delta t})\mathbf{n}_t$ is simply the projection of the difference $\mathbf{n}_{t+\Delta t} - \mathbf{n}_t$ in the plane perpendicular to $\mathbf{n}_t$. So, up to the factor $\frac{\Theta(\mathbf{n}_t, \mathbf{n}_{t+\Delta t})}{\sin\Theta(\mathbf{n}_t, \mathbf{n}_{t+\Delta t})}$ which is in practice very close to 1, the equation obtained from the integration of (25) and the usual approximation to the derivative are essentially the same.

6 Helping the disambiguation of stereo matches with the motion field

The previous analysis allows us to propose a way to guide the stereo correspondence process with the motion field. The idea is based upon the notion of *invariant*. Suppose

that we observe two segments d_1 and d_2 in the first camera and that we want to find their corresponding segments d'_1 and d'_2 in the second camera. For each segment d_i, $i = 1, 2$ we will find a number of possible matches d'_{ij} $j = 1, k_i$ in the second image. For each pair of hypothesis (d_1, d'_{1k}) and (d_2, d'_{2l}) we know that we can write eight linear equations in Ω and $\mathbf{V}$ if we assume that the two corresponding 3D segments D_1 and D_2 belong to the same object. Those equations are redundant since there are only six unknowns which means that we can write two equations, independent of Ω and $\mathbf{V}$, *invariant*, that must be satisfied if the two hypothesis are correct.

Denoting with an index 1 (resp. 2) quantities related to the first (resp. second) hypothesis, the four equations obtained from (18) and (19) can be written

$$\begin{aligned}
\dot{\mathbf{n}}_1^T \mathbf{L}_1 &= \Omega^T (\mathbf{n}_1 \wedge \mathbf{L}_1) \\
\dot{\mathbf{p}}_1^T \mathbf{L}_1 &= \Omega^T (\mathbf{p}_1 \wedge \mathbf{L}_1) \\
\dot{\mathbf{n}}_2^T \mathbf{L}_2 &= \Omega^T (\mathbf{n}_2 \wedge \mathbf{L}_2) \\
\dot{\mathbf{p}}_2^T \mathbf{L}_2 &= \Omega^T (\mathbf{p}_2 \wedge \mathbf{L}_2)
\end{aligned}$$

A necessary and sufficient condition for this system to yield a solution is that the 4×4 determinant $det(\mathbf{N}_1, \mathbf{P}_1, \mathbf{N}_2, \mathbf{P}_2)$ is equal to 0. The vectors $\mathbf{N}_i$ and $\mathbf{P}_i$ are four-dimensional and

$$\mathbf{N}_i = \begin{bmatrix} \mathbf{n}_i \wedge \mathbf{L}_i \\ \dot{\mathbf{n}}_i^T \mathbf{L}_i \end{bmatrix}$$

$$\mathbf{P}_i = \begin{bmatrix} \mathbf{p}_i \wedge \mathbf{L}_i \\ \dot{\mathbf{p}}_i^T \mathbf{L}_i \end{bmatrix}$$

If this condition is met, we can compute a value for Ω and replace it in the four equations (23) and (24)

$$\begin{aligned}
\mathbf{M}_1^T (\mathbf{n}_1 \wedge \Omega + \dot{\mathbf{n}}_1) &= -\mathbf{V}^T \mathbf{n}_1 \\
\mathbf{M}_1^T (\mathbf{p}_1 \wedge \Omega + \dot{\mathbf{p}}_1) &= -\mathbf{V}^T \mathbf{p}_1 + \mathbf{T}^T \dot{\mathbf{p}}_1 \\
\mathbf{M}_2^T (\mathbf{n}_2 \wedge \Omega + \dot{\mathbf{n}}_2) &= -\mathbf{V}^T \mathbf{n}_2 \\
\mathbf{M}_2^T (\mathbf{p}_2 \wedge \Omega + \dot{\mathbf{p}}_2) &= \mathbf{V}^T \mathbf{p}_2 + \mathbf{T}^T \dot{\mathbf{p}}_2
\end{aligned}$$

and we obtain a second 4×4 determinant obtained from those equations and which must also be 0. This second determinant is noted $det(\mathbf{Q}_1, \mathbf{R}_1, \mathbf{Q}_2, \mathbf{R}_2)$, the vectors $\mathbf{Q}_i$ and $\mathbf{R}_i$ are four-dimensional, and

$$\mathbf{Q}_i = \begin{bmatrix} \mathbf{n}_i \\ \mathbf{M}_i^T (\mathbf{n}_i \wedge \Omega + \dot{\mathbf{n}}_i) \end{bmatrix}$$

$$\mathbf{R}_i = \begin{bmatrix} \mathbf{p}_i \\ \mathbf{M}_i^T (\mathbf{p}_i \wedge \Omega + \dot{\mathbf{p}}_i) \end{bmatrix}$$

This provides a possible way to help the stereo correspondence. Note that the complexity is the product of the number of possible hypothesis $k_1 k_2$, typically a number between 10 and 100. We have not yet tested this idea on real data but we think it is worth pursuing.

7 Experimental Results

We have tested our method using both synthetic data and real image sequences.

7.1 Results on synthetic data

Our synthetic data include a sequence with a 3D object moving with constant angular and translational velocities. It is observed by a binocular stereo rig and noise is added to the images. The noise level is high, and is adapted to the fact that on real data each segment is a result of edge detection and polygonal approximation.

Noise free 3D data, in our case a box with 16 segments, is first constructed. We then apply to it a motion with $\mathbf{\Omega} = [0.0, 0.03, 0.0]$ and $\mathbf{V} = [9.55, 0.0, 30.15]$. We then project the object in the first and second cameras, using real calibration parameters obtained from our stereo system. The baseline is about 43mm, the focal lenth, 8mm and the pixel size, 8x14μm^2. We then add noise independently for each image to the endpoints of the projected segments. The noise for each segment has two independent components: one component is parallel to the segment and another is perpendicular to it. The parallel component is a random scalar with zero mean and five pixels of deviation; the normal component is a random scalar with zero mean and 0.5 pixels of deviation. This is a fair model of the precision of our edge detection and polygonal approximation processes.

We then estimate the lines motion field $\dot{n}$ using the method described in section 5. The results are presented in the following table when we use two, three or four consecutive frames. We show the computed values of $\mathbf{\Omega}$ and $\mathbf{V}$ and two measures of the error. ϵ_1 is the angle, in degrees, between the real and the computed velocities, ϵ_2 is the relative error for the magnitudes.

Frames	$\mathbf{\Omega}$	ϵ_1	ϵ_2	$\mathbf{V}$	ϵ_1	ϵ_2
2	[-0.006 0.028 -0.004]	14.4	0.03	[12.6 -19.1 30.3]	30.3	0.2
3	[-0.001 .027 0.0]	2.6	0.06	[14.3 -3.7 31.9]	8.02	0.11
4	[0.0 0.029 -0.001]	3.6	0.02	[10.3 -2.5 30.3]	4.6	0.01

7.2 Results on real data

For real data we have used two sequences of stereo images obtained by our mobile robot. The parameters of the stereo rig are the same as before. The distance between the objects and the cameras varies from 2m to 5m. In the first case the robot has a pure translational motion of 10cm at each step. Its motion is horizontal ($\mathbf{V} \simeq [100.0, 0.0, 0.0]$). Figure 5 shows two consecutive views as seen by the first camera. We obtained $\mathbf{V} = [99.781527, 3.370187, -4.967088]$ using four consecutive pair of images.

In the second case (figure 8), the robot rotates 3.30 degrees around its vertical axis and moves forward 10cm, at each step. But we have had problems with the control. It is naive to believe that the robot motion is going to be exactly the requested one. Therefore, in order to analyze the results, we estimate the kinematic screw of the 3D lines using the algorithms previously described. We then apply this estimated motion to the 3D line segments reconstructed by stereo at time t and compare the *predicted* 3D segments with those obtained from the stereo reconstruction at time $t + \Delta t$.

To display the results, we project these two sets of 3D segments on the same planes and verify if they superpose well.

Figures 6 and 7 show the perspective projection of these two sets of 3D line segments in the two image planes and also their orthographic projections on a fronto parallel and horizontal plane, for the first motion. The predicted lines are represented with solid lines and those reconstructed at time $t + \Delta t$ are represented with dashed lines.

The segments located in the very left or right of the image are not present in both images. The segments on the right hand side are further from the cameras than those on the left hand side, and that explains why in the front view our precision decreases from left to right. And that is not surprising because the precision of stereo decreases with depth. In both cases, the results are very satisfactory. Figures 9 and 10 show the results for the second motion.

Finally, another experimental result is presented using a more complicated scene, where depth variation is more important, and when matching the line segments, there are a lot of ambiguities. In this case (figure 11 and 12), the robot rotates 2.20 degrees around its vertical axis and moves forward 10cm, at each step. We represent the results as in the previous cases. Figures 13 and 14 show the perspective projection of these two sets of 3D line segments in the two image planes and also their orthographic projections on a fronto parallel and horizontal plane. Although the number of stereo pair of segments which have been tracked is less than in the previous experiments, but the egomotion estimation is yet robust.

The results on synthetic and real data show that the algorithm is reliable and robust, whenever our hypothesis are verified. That means when the angular and

translational velocities are both constant (see the results on synthetic data), or when the interframe motion is small (see the results on real data).

8 Discussion

The line motion field approach developed in this paper raises the question of the advantage of this approach over the pure use of stereo line correspondences as in, [ZF90] and [Zha90] for example. These are at least three interesting points in favor of the 2D approach:

1) As we saw in section 6, the motion field gives theoretical constraints to help disambiguating the stereo matches.

2) It allows to relieve the burden on stereo which can concentrate on new incoming segments and leave the two-dimensional token tracker do the matching works and still estimating 3D motion and structure.

3) It increases the robustness of the motion from stereo system since if one of the cameras fails temporarily, the system can still operate through equation (18):

$$\mathbf{\Omega}^T(\mathbf{n} \wedge \mathbf{L}) = \dot{\mathbf{n}}^T\mathbf{L}$$

to recover the angular velocity and then equation (23):

$$\mathbf{M}^T(\mathbf{n} \wedge \mathbf{\Omega} + \dot{\mathbf{n}}) = -\mathbf{V}^T\mathbf{n}$$

to recover the translational velocity. We need of course, the images of at least three non coplanar and non collinear lines.

It means that we may use stereo to obtain only an estimation of the 3-D lines position at the begining, and then it is enough to follow this lines on one of the cameras. The key point is that we do not use stereo continuously, but only from time to time to help the system gather information about new visible segments. the rest of the time we are simply exploiting previous matches by tracking them simultaneously. Since image tracking is a priori a lot less costly than performing stereo [FDA*88], we obtain a significant decrease in cost.

9 Conclusion

We have investigated the relationships between the three-dimensional position and orientation of a line, its kinematic screw, and the motion field it generates. Since it

appears to be fairly easy to obtain a robust estimate of the line motion field through a token tracker, this approach is quite promising and we have used it to study some aspects of the cooperation between motion and stereo.

However, if we want a tighter cooperation between stereo and motion, there are other ways of doing it. For example, for tracking the image segments we may also use the reprojected stereo information; this is one of our current projects in which we have already obtained some interesting results.

We have implemented and tested the stereo-motion cooperation on synthetic and real data. The motion-stereo cooperation has only been investigated so far from the theoretical standpoint but will be implemented soon. We are also in the process of applying the results obtained to improve the prediction phase of the token tracker.

The contributions of this paper, as we see them, are first to provide a complete description of the relationship between the three-dimetional motion of lines and their two-dimensional motion field; a second contribution has been to show that cooperation between stereo and motion was not only possible but also beneficial: stereo can be used to establish initial matches which are then continued over time through image tracking and yield accurate estimations of the three-dimesional rigid motion. This reduces the burden placed on stereo and has been fully implemented; a third contribution has been to show that the computation of the two-dimensional motion field of line segments can be used to help disambiguate stereo matches. This part has not been implemented yet.

Acknowledgments We have enjoyed many fruitful discussions with Zhengyou Zhang, Thierry Vieville, and Regis Vaillant during the progress of this work.

10 Appendix

We verify that, if Θ is the angle between unit vectors $\mathbf{n}_t$ and $\mathbf{n}_{t+\Delta t}$:

$$\omega = (\frac{\Theta}{\Delta t})\frac{\mathbf{n}_t \wedge \mathbf{n}_{t+\Delta t}}{\| \mathbf{n}_t \wedge \mathbf{n}_{t+\Delta t} \|}$$

is a solution of the equation (30):

$$\mathbf{n}_{t+\Delta t} = \mathbf{n}_t + \frac{\sin(\theta \Delta t)}{\theta}\tilde{\omega}\mathbf{n}_t + \frac{1-\cos(\theta \Delta t)}{\theta^2}\tilde{\omega}^2\mathbf{n}_t$$

where $\theta = \| \omega \|$.
Let us note that:

$$\| \omega \| = |\frac{\Theta}{\Delta t}| \qquad \| \mathbf{n}_t \wedge \mathbf{n}_{t+\Delta t} \| = |sin(\Theta)| \qquad \mathbf{n}_t^T \mathbf{n}_{t+\Delta t} = cos(\Theta)$$

and let us define:

$$\mathbf{I}_1 = \frac{\sin(\theta\Delta t)}{\theta}\tilde{\omega}\mathbf{n}_t \qquad \mathbf{I}_2 = \frac{1-\cos(\theta\Delta t)}{\theta^2}\tilde{\omega}^2\mathbf{n}_t$$

We can then write:

$$\mathbf{I}_1 = (\mathbf{n}_t \wedge \mathbf{n}_{t+\Delta t}) \wedge \mathbf{n}_t = \mathbf{n}_{t+\Delta t} - cos(\Theta)\mathbf{n}_t$$

and we also have:

$$\mathbf{I}_2 = \frac{1-\cos(\Theta)}{\theta|sin(\Theta)|}\,(\mathbf{n}_t \wedge \mathbf{n}_{t+\Delta t}) \wedge \tilde{\omega}\mathbf{n}_t$$

replacing $\mathbf{I}_1$ by its value in $\mathbf{I}_2$:

$$\mathbf{I}_2 = \frac{1-cos(\Theta)}{sin(\Theta)^2}(\mathbf{n}_t \wedge \mathbf{n}_{t+\Delta t}) \wedge (\mathbf{n}_{t+\Delta t} - cos(\Theta)\mathbf{n}_t)$$

Using again the fact that all vectors $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c} \in \mathbf{R}^3$ verify:

$$(\mathbf{a} \wedge \mathbf{b}) \wedge \mathbf{c} = (\mathbf{a}^T\mathbf{c})\,\mathbf{b} - (\mathbf{b}^T\mathbf{c})\,\mathbf{a}$$

we obtain:

$$\mathbf{I}_2 = \frac{1-cos(\Theta)}{sin(\Theta)^2}(-1 + cos(\Theta)^2)\mathbf{n}_t = cos(\Theta)\mathbf{n}_t - \mathbf{n}_t$$

therefore, $\mathbf{I}_1 + \mathbf{I}_2 = \mathbf{n}_{t+\Delta t} - \mathbf{n}_t$ and equation (30) is verified.

References

[AF89] Nicholas Ayache and Olivier D. Faugeras. Maintaining Representations of the Environment of a Mobile Robot. *IEEE transactions on Robotics and Automation*, 5(6):804–819, December 1989. also INRIA report 789.

[AL87] N. Ayache and F. Lustman. Fast and reliable passive trinocular stereovision. In *Proceedings ICCV '87, London*, pages 422–427, IEEE, June 1987.

[Ber86] M. Berthod. Approximation polygonale de chaînes de contours. Programmes C, 1986. INRIA.

[Can86] J.F. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:769–798, November 1986.

[Der87] R. Deriche. Using Canny's Criteria to Derive an Optimal Edge Detector Recursively Implemented. *The International Journal of Computer Vision*, 2:167–187, April 1987.

[DF90] Rachid Deriche and Olivier D. Faugeras. Tracking Line Segments. In *Proceedings of the 1st ECCV*, pages 259–268, Springer Verlag, April 1990.

[FDA*88] Olivier D. Faugeras, Nourr-Eddine Deriche, Nicholas Ayache, Francis Lustman, and Ercole Giuliano. Depth and Motion Analysis: the machine being developed within esprit project 940. In *Proceedings of the IAPR Workshop on Computer Vision (Special Hardware and Industrial Applications), Tokyo, Japan*, pages 35–44, October 1988.

[FDN89] Olivier D. Faugeras, Nourr-Eddine Deriche, and Nassir Navab. From optical flow of lines to 3D motion and structure. In *Proceedings IEEE RSJ International Workshop on Intelligent Robots and Systems '89*, pages 646–649, 1989. Tsukuba, Japan.

[FLT87] Olivier D. Faugeras, Francis Lustman, and Giorgio Toscani. Motion and Structure from point and line matches. In *Proceedings of the First International Conference on Computer Vision, London*, pages 25–34, June 1987.

[FT86] Olivier D. Faugeras and Giorgio Toscani. The Calibration Problem for Stereo. In *Proceedings CVPR '86, Miami Beach, Florida*, pages 15–20, IEEE, June 1986.

[KA87] Yeon C. Kim and J. K. Aggarwal. Determining object motion in a sequence of stereo images. *IEEE journal of Robotics and Automation*, 3(6):599–613, December 1987.

[LH86a] Y. Liu and T.S. Huang. Estimation of Rigid Body Motion Using Straight Line Correspondences. In *Proceedings Workshop on Motion: Representation and Analysis, Charleston, South Carolina, USA*, pages 47–51, IEEE, May 1986.

[LH86b] Y. Liu and T.S. Huang. Estimation of Rigid Body Motion Using Straight Line Correspondences: Further Results. In *Proceedings 8th ICPR, Paris, France*, pages 306–307, IEEE, October 1986.

[Lon81] H.C. Longuet-Higgins. A Computer Algorithm for Reconstructing a Scene from Two Projections. *Nature*, 293:133–135, 1981.

[Lon84] H.C. Longuet-Higgins. The reconstruction of a scene from Two Projections - Configurations that defeat the 8-Point Algorithm. In *Proceedings First Conference on Artificial Intelligence Applications, Denver, Colorado*, pages 395–397, 1984.

[May79] P. S. Maybeck. *Stochastic Models, Estimation and Control.* Academic Press, 1979.

[May87] S.J. Maybank. *A theorical study of optical flow.* PhD thesis, Birkbeck College, University of London, 1987.

[NDF90] Nassir Navab, Rachid Deriche, and Olivier D. Faugeras. Recovering 3d motion and structure from stereo and 2d token tracking cooperation. In *Proc. the third International Conference on Computer Vision*, IEEE, Osaka, Japon, December 1990.

[Rod40] O. Rodrigues. Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire. *Journal de Mathématiques Pures et Appliquées*, 5, 1840. pp. 380–440.

[SA90] Minas E. Spetsakis and John Aloimonos. Structure from Motion Using Line Correspondences. *The International Journal of Computer Vision*, 4:171–183, 1990.

[TDF88] Giorgio Toscani, Nourr-Eddine Deriche, and Olivier D. Faugeras. 3D Motion Estimation Using a Token Tracker. In *Proceedings of the IAPR Workshop on Computer Vision (Special Hardware and Industrial Applications), Tokyo, Japan*, pages 257–261, October 1988.

[Tsa86] Roger Tsai. An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. In *Proceedings CVPR '86, Miami Beach, Florida*, pages 364–374, IEEE, June 1986.

[YC90] G. S. J. Young and R. Chellappa. 3-D Motion Estimation Using a Sequence of Noisy Stereo Images: Models,Estimation and Uniqeness, Results. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12, august 1990.

[ZF90] Zhengyou Zhang and Olivier D. Faugeras. Determining motion from 3D line segment matches: a comparative study. In *Proceedings of the British Machine Vision Conference*, pages 85–90, September 1990.

[Zha90] Zhengyou Zhang. *Motion Analysis from a Sequence of Stereo Frames and its Applications.* PhD thesis, University of Paris-Sud, Orsay, Paris, France, 1990. in English.

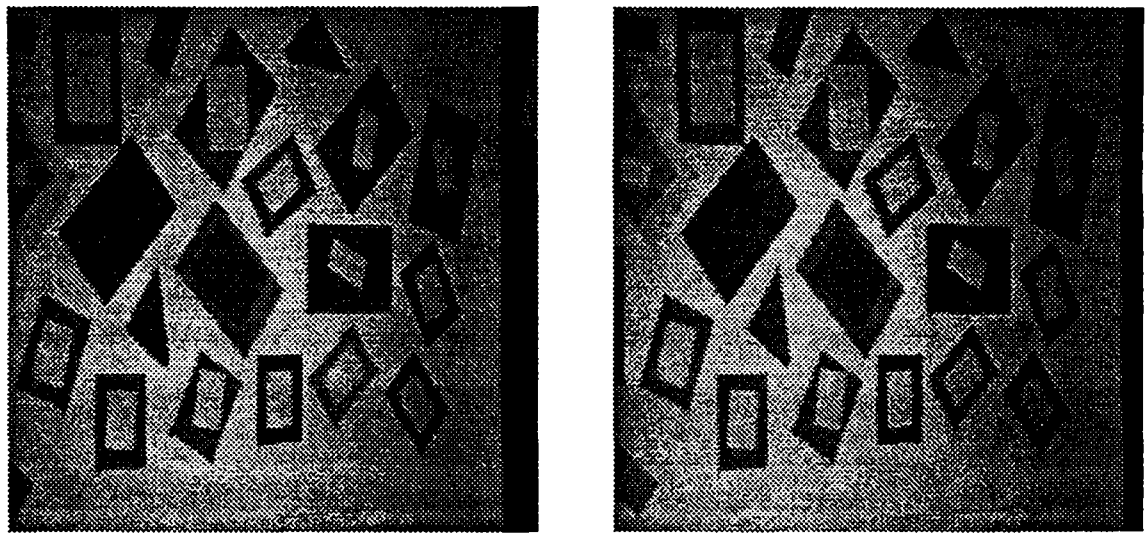

Figure 5: Pure translational motion; two consecutive images in the first camera

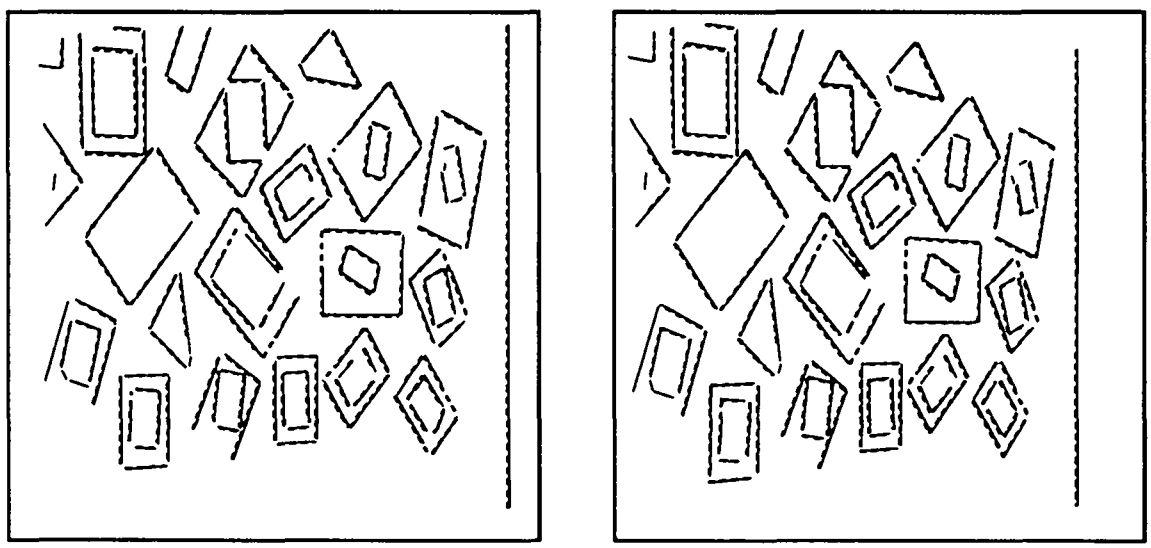

Figure 6: Real data comparison: Projection on the two cameras.

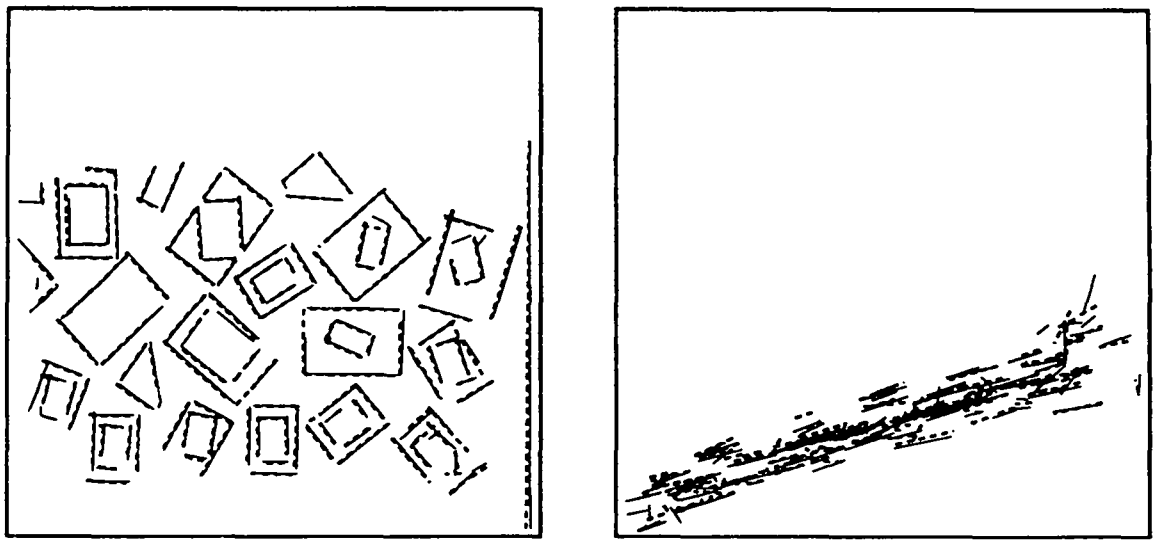

Figure 7: Real data comparison: Front view and top view

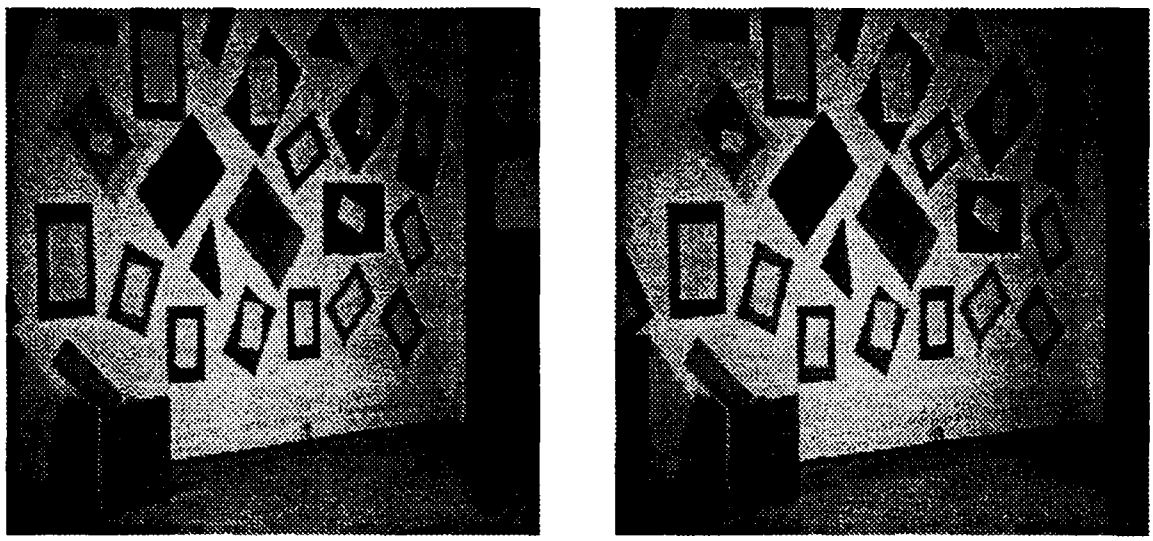

Figure 8: Rotational and translational motion; two consecutive images in the first camera

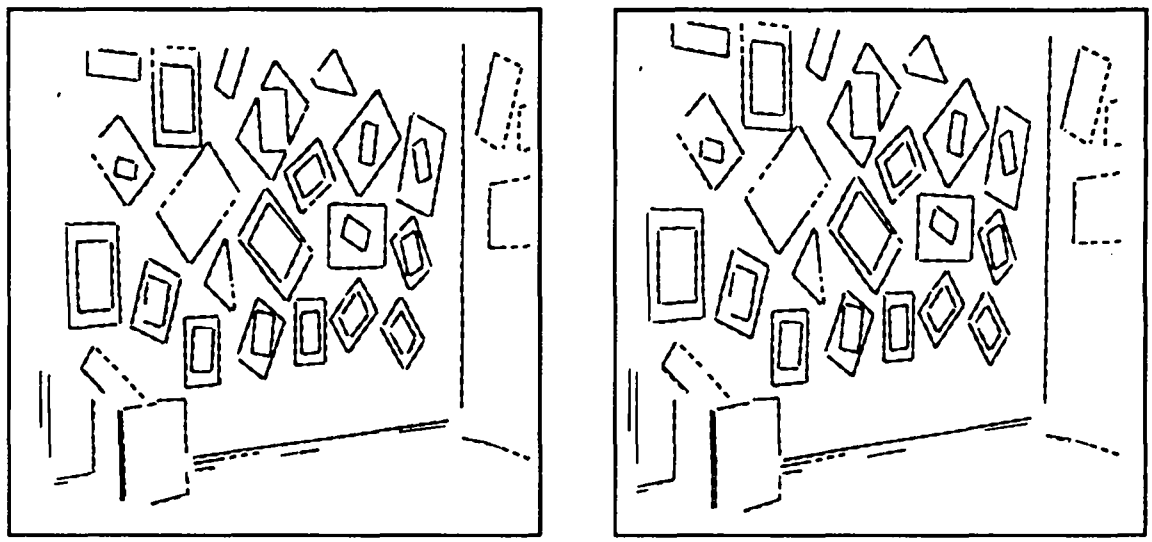

Figure 9: Real data comparison: Projection on the two cameras.

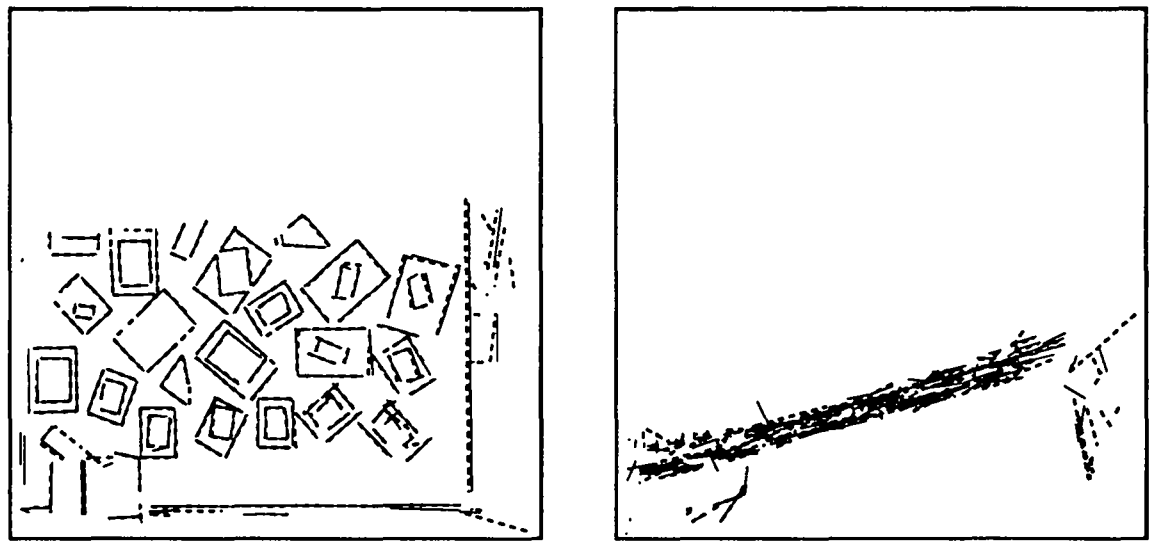

Figure 10: Real data comparison: Front view and top view

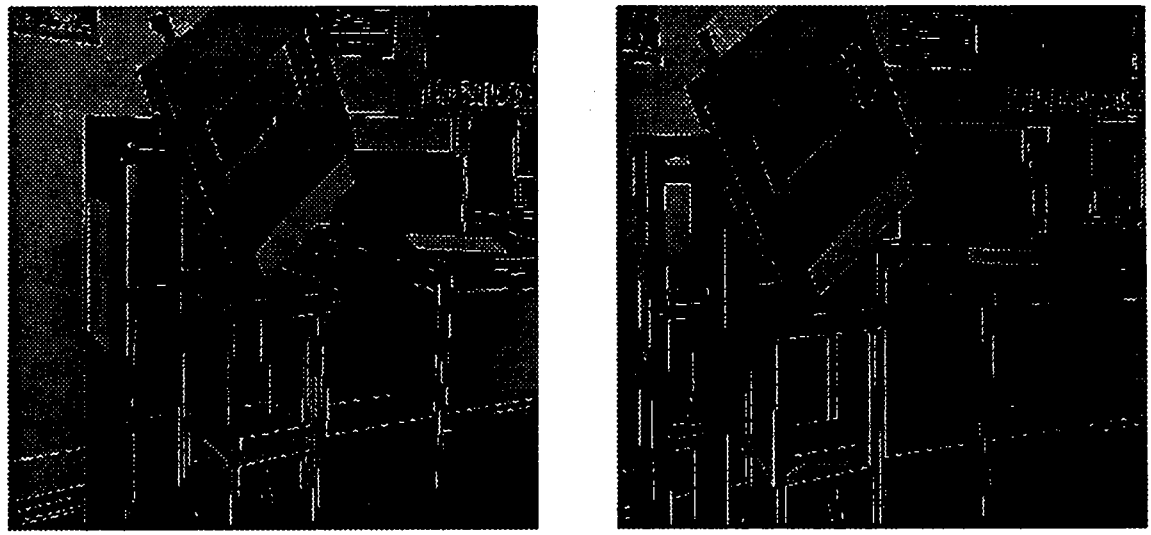

Figure 11: The detected edges on a pair of stereo images, before the robot motion.

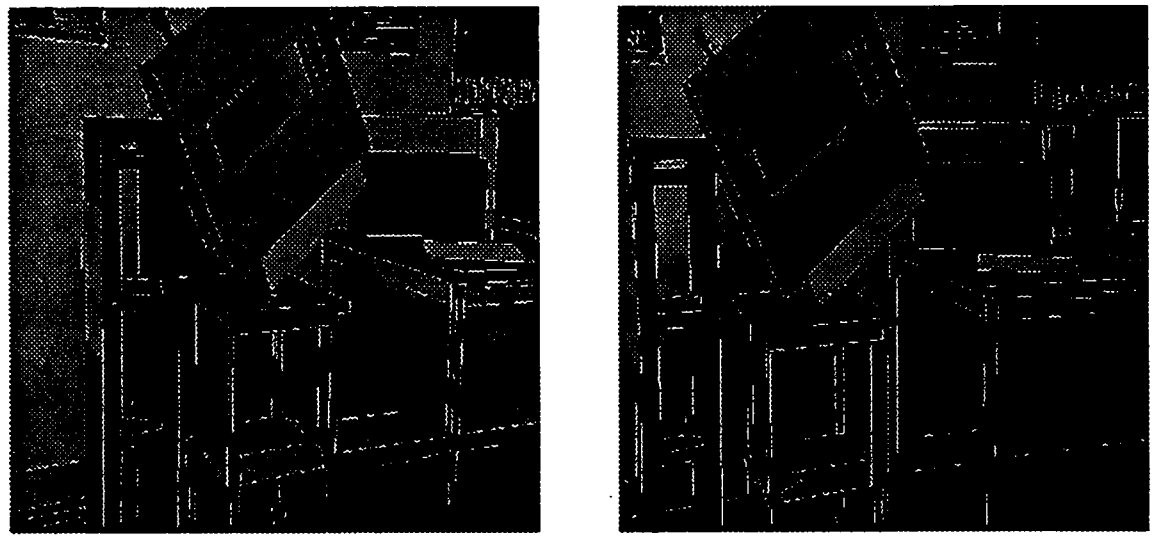

Figure 12: The detected edges on a pair of stereo images, after the robot motion.

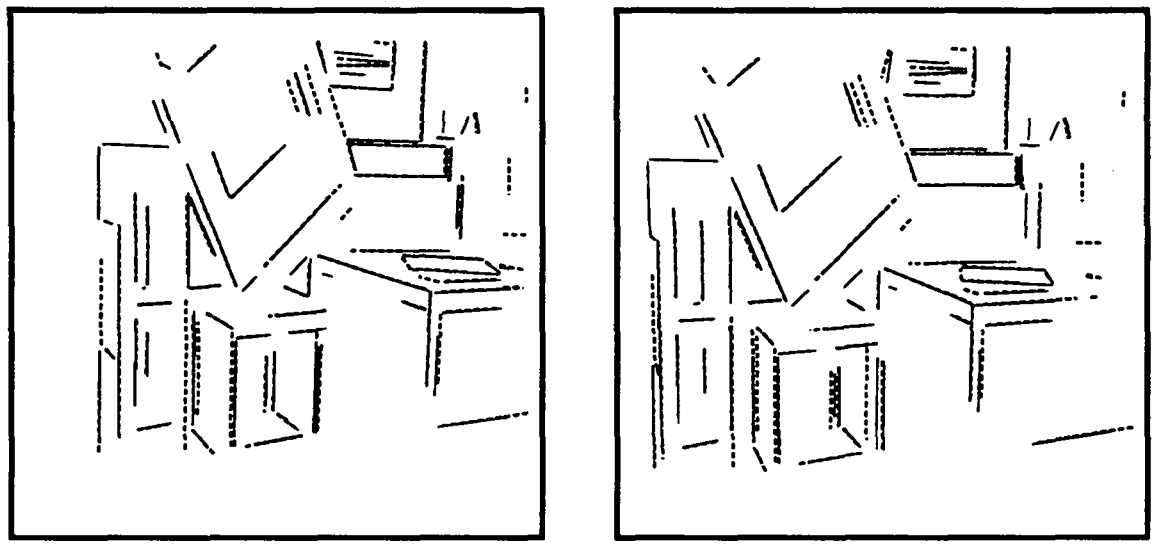

Figure 13: Real data comparison: Projection on the two cameras.

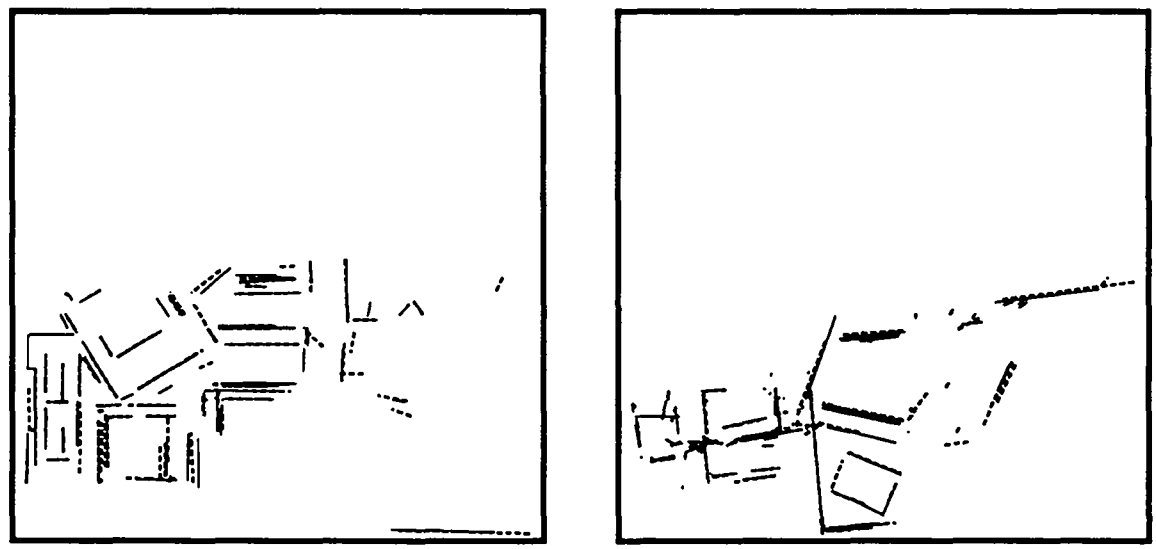

Figure 14: Real data comparison: Front view and top view

ISSN 0249-6399