Fusion of visual data through dynamic stereo-motion cooperation

Nassir Navab, and Zhengyou Zhang
INRIA Sophia Antipolis
2004 Route des Lucioles
06565 Valbonne Cedex
FRANCE

Abstract

Integrating information from sequences of stereo images can lead us to a robust visual data fusion. Instead of considering stereo and temporal matchings as two independent processes, we propose a unified scheme in which each borrows dynamically some information from the other. Using an iterative approach and statistical error analysis, different observations are appropriately combined to estimate the motion of the stereo rig and build a dynamic 3D model of the environment. We also show how motion estimation and temporal matching can be used to add new stereo matches. The algorithm is demonstrated with real images. Implemented on a mobile robot, it shows how fusion of visual data can be useful for an autonomous vehicle working in an unknown environment.

1 Introduction

In stereo and motion analysis, most of previous work has been conducted using either two or three static cameras [27] or a sequence of monocular images obtained by a moving camera [4]. Several researchers tried to combine these two process to find faster and more robust algorithms [7, 23, 18, 16, 21, 17, 22, 19, 2, 11].

We believe in the efficiency of stereo-motion cooperation. This paper is another attempt to improve this idea.

To extract 3D information from real images, "meaningful" extracted features, such as corner points, edges, regions, etc., are often used to reduce the computational cast and matching ambiguities. In this paper, we use the line segments obtained by an edge detector. Line segments are present in most of the real-world scenes such as : highways,car traffic tunnels, long indoor hallways or industrial assembly.

In [7], [19] and [9] , we tried to make cooperate two existing algorithms–a hypothesis-verification based stereo matching algorithm [3] and a monocular line tracking algorithm [5].Very soon we realized that each of these processes may work faster and better (in terms of robustness) if they could borrow dynamically some information from each other. And the motion estimation could play an important role of intermediary between these two processes. If we want a tighter cooperation between stereo and motion, we must not consider them as two different processes with some interactions from time to time.

We present a unified iterative algorithm for both temporal tracking of stereo pairs of segments and camera system ego-motion estimation; which consequently allows us to keep track of our 3D reconstructions. The algorithm is based on a dynamic interaction between different sources of information.

Figure3 shows the general scheme of the algorithm. This scheme is adapted from that of Droid [14, 10]. The basic difference is that we use straight lines features as token, where Droid make use of point feature, and once the cameras system ego-motion is estimated, we use that information for tacking 2D lines on each camera.

In section 4, we describe how to use straight line tokens to estimate the cameras system ego-motion and its associated covariance matrix. The algorithm is decomposed in three different steps, as shown in figure3. In section 6, we describe these three different steps. Finally, section 7 shows briefly the results of the different steps of our algorithm on real images obtained by INRIA mobile robot.

2 Preliminaries

Vectors are represented in bold face, i.e $\mathbf{x}$. Transposition of vectors and matrices is indicated by ^{T}, i.e $\mathbf{x}^T$. $\dot{\mathbf{x}}$ denotes the time derivative of $\mathbf{x}$, i.e $\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt}$. 3D points P are represented by vectors $\mathbf{P} = (X, Y, Z)^T$. 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 $\mathbf{y}$. $\mathbf{I}_n$ represents the $n \times n$ identity matrix.

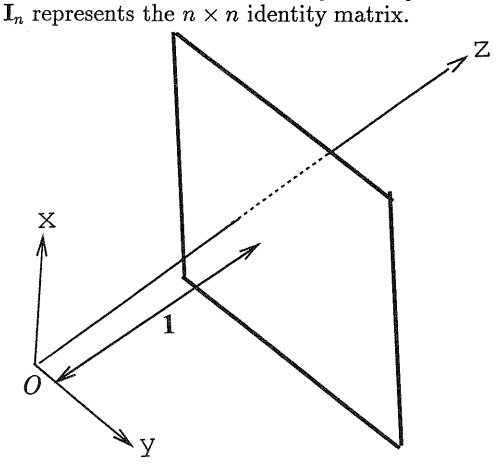

Fig. 1. Pinhole model of a camera

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 [24, 6] 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. When using a pair of calibrated stereo cameras, everything is written in one of the cameras coordinate systems.

3 The *Pluckerian* line representation

Different line representations in $\Re^2$ and $\Re^3$, have been used -1z in computer vision works. Though the theoretical results may be equivalent, one is more or less suitable for a possible implementation. Here, we use the *Pluckerian*

line representation. The *Pluckerian* representation is the canonical line representation in projective geometry.

The *Pluckerian* coordinates are defined as follow: Let $\mathbf{P}$ be the cartesian coordinates of an arbitrary point on a line D in a 3D space and $\mathbf{l}$ be the unit direction vector of the line D. We introduce and often use the vector $\mathbf{H}$ which is the orthogonal from the origin $\mathbf{O}$ to the line D. It is easily seen that

$$\mathbf{P} \wedge \mathbf{l} = \mathbf{H} \wedge \mathbf{l}$$

or

$$\mathbf{P} \wedge \mathbf{l} = \mathbf{N} = h\mathbf{n}$$

where $\mathbf{N} = \mathbf{H} \wedge \mathbf{l}$, and $\mathbf{n} = \frac{\mathbf{N}}{\|\mathbf{N}\|}$ is the normal to the plane defined by the 3D line D and the origin O, and finally $h = \| \mathbf{H} \| = \| \mathbf{N} \|$ represents the distance of the line to the origin (see fig 2). Therefore, the line equation will be

$$\mathbf{P} \wedge \mathbf{l} = \mathbf{N}$$

The two vectors $(\mathbf{l}, \mathbf{N})$ define the *Pluckerian* coordinates of the line D. Note that $\mathbf{H}$, $\mathbf{l}$, and $\mathbf{N}$ form a right handed coordinate system.

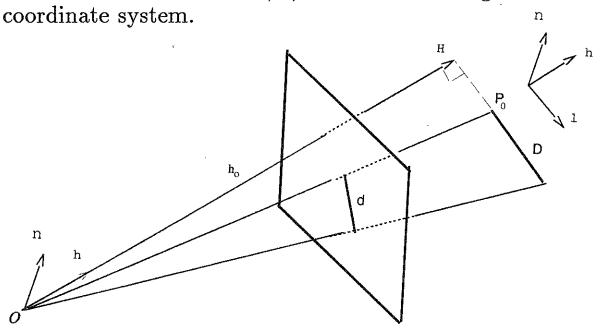

Fig. 2. The vectors $\mathbf{n}$, $\mathbf{l}$, and $\mathbf{h}$.

Using this line representation we need four parameters to represent a 3D line. Two parameters for the unit line direction $\mathbf{l}$, and two parameters to define the vector $\mathbf{N}$ or $\mathbf{H}$ which are orthogonal to the line direction $\mathbf{l}$.

Image lines : A line d, the projection of a 3D line D on the image plane is called a 2D line. In the camera coordinate system, this line may be considered as a 3D line which lies on the plane $z = 1$. Therefore its equation is simply:

$$\mathbf{m}^T\mathbf{n} = 0$$

where $\mathbf{m} = (x, y, 1)^T$ is an arbitrary point on d.

The vector $\mathbf{n}$ is the normal to the plane containing the 3D line, its image and the camera optical center. Therefore, this vector $\mathbf{n}$ is the same as the vector $\mathbf{n} = \frac{\mathbf{N}}{\|\mathbf{N}\|}$ introduced in the previous paragraph. We may even use the vector $\mathbf{N}$ to represent the image line when the 3D line is given in camera coordinate system. Usually we have only access to the image lines. Therefore, we prefer in general represent the image lines by a unit vector $\mathbf{n}$.

If we take $\mathbf{n} = (\alpha, \beta, \gamma)^T$ the line equation is written as:

$$\alpha x + \beta y + \gamma = 0$$

The vector $\mathbf{n}$ is a unit vector and therefore, our 2D line representation depend only on two parameters.

4 Ego-motion estimation

Much work has been done on the motion estimation from straight lines. In the case of discrete motion, we can particularly mention the works of Liu, Huang, Spetsakis, Aggarwal, Chellapa, and Vieville [13, 12, 20, 1, 4, 25] on the monocular sequences and that of Zhang [26] on the stereo sequences. And in the case of continuous line motion analysis approach we may refer to the works of Faugeras, Navab, and Henriksen [8, 9, 19, 11]. One may easily verify that for each of the fundamental formulae obtained in one case (discrete or continue) one can find a similar and related one in the other case [15]. In this paper, we use the continuous approach. After the first steps, when we can use more frames this approach shows its real advantages.

Let us take a 3D line represented by the vectors $(\mathbf{N}, \mathbf{l})$. We now describe its motion $(\dot{\mathbf{N}}, \dot{\mathbf{l}})$. In order to gain more insight into the problem, we 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 $\mathbf{O}$.We can also suppose that the object is static and the camera system has such a motion description.

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

$$\dot{\mathbf{P}} = \mathbf{V} + \mathbf{\Omega} \wedge \mathbf{P} \tag{1}$$

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

$$\dot{\mathbf{l}} = \mathbf{\Omega} \wedge \mathbf{l} \tag{2}$$

The vector $\mathbf{H}$ can be expressed as

$$\mathbf{H} = \mathbf{P} - (\mathbf{P}^T\mathbf{l})\mathbf{l} \tag{3}$$

therefore,

$$\dot{\mathbf{H}} = \dot{\mathbf{P}} - (\mathbf{P}^T\mathbf{l})\dot{\mathbf{l}} - (\dot{\mathbf{P}}^T\mathbf{l} + \mathbf{P}^T\dot{\mathbf{l}})\mathbf{l}$$

Replacing $\dot{\mathbf{P}}$ and $\dot{\mathbf{l}}$ by their values from equations 1 and 2,

$$\dot{\mathbf{H}} = \mathbf{V} + \mathbf{\Omega} \wedge [\mathbf{P} - (\mathbf{P}^T\mathbf{l})\mathbf{l}] - (\mathbf{V}^T\mathbf{l})\mathbf{l}$$

we obtain:

$$\dot{\mathbf{H}} = \mathbf{\Omega} \wedge \mathbf{H} + \mathbf{V} - (\mathbf{V}^T\mathbf{l})\mathbf{l} \tag{4}$$

Then it's easy to obtain $\dot{\mathbf{N}}$, the time derivative of $\mathbf{N} = \mathbf{H} \wedge \mathbf{l}$:

$$\begin{aligned} \dot{\mathbf{N}} &= \dot{\mathbf{H}} \wedge \mathbf{l} + \mathbf{H} \wedge \dot{\mathbf{l}} \\ &= (\mathbf{\Omega} \wedge \mathbf{H} + \mathbf{V} - (\mathbf{V}^T\mathbf{l})\mathbf{l}) \wedge \mathbf{l} + \mathbf{H} \wedge (\mathbf{\Omega} \wedge \mathbf{l}) \\ &= \mathbf{\Omega} \wedge (\mathbf{H} \wedge \mathbf{l}) + \mathbf{V} \wedge \mathbf{l} \end{aligned}$$

and we obtain:

$$\dot{\mathbf{N}} = \mathbf{\Omega} \wedge \mathbf{N} + \mathbf{V} \wedge \mathbf{l} \tag{5}$$

Therefore, the motion of a 3D line can be defined as follows:

$$\begin{bmatrix} \dot{\mathbf{l}} \\ \dot{\mathbf{N}} \end{bmatrix} = \mathbf{D} \begin{bmatrix} \mathbf{l} \\ \mathbf{N} \end{bmatrix} \tag{6}$$

where the matrix $\mathbf{D}$ is defined as follows:

$$\mathbf{D} = \begin{bmatrix} \tilde{\mathbf{\Omega}} & \mathbf{0} \\ \tilde{\mathbf{V}} & \tilde{\mathbf{\Omega}} \end{bmatrix}$$

Line Motion Field Equation: What we measure from the images are the 2D lines represented by the unit vectors $\mathbf{n}$ and their motion fields $\dot{\mathbf{n}}$. Therefore, we give here the line motion field equation. Line motion field equation was first given in [19]. We used two points on the line to obtain that result. Here we draw the same equation from the above equations. we have $\mathbf{n} = \frac{\mathbf{N}}{\|\mathbf{N}\|}$, therefore

$$\begin{aligned}\dot{\mathbf{n}} &= \frac{1}{\| \mathbf{N} \|}(\mathbf{I}_3 - \frac{\mathbf{N}\mathbf{N}^T}{\| \mathbf{N} \|^2})\dot{\mathbf{N}} \\ &= \Omega \wedge \mathbf{n} + \frac{1}{\| \mathbf{N} \|}(\mathbf{I}_3 - \mathbf{n}\mathbf{n}^T)(\mathbf{V} \wedge \mathbf{l})\end{aligned}$$

From the definition of $\mathbf{N}$, see section 3, we have $\| \mathbf{N} \| = h$ the distance of the 3D line from the origin $\mathbf{O}$. We use also the fact that for a unit vector $\mathbf{n}$ in $\Re^3$, $\mathbf{I}_3 - \mathbf{n}\mathbf{n}^T = \tilde{\mathbf{n}}^2$. After a little simplification we obtain the line motion field equation:

$$\dot{\mathbf{n}} = \Omega \wedge \mathbf{n} + \frac{\mathbf{V}^T\mathbf{n}}{h}(\mathbf{l} \wedge \mathbf{n}) \tag{7}$$

where n, L and h are defined as in section 2, and Ω and $\mathbf{V}$ represent the angular and translational velocities. And they are all written in the camera coordinate system.

If we define a vector $\mathbf{H}$, passing through optical center and orthogonal to 3D line D. One may recover Ω only in the direction of $\mathbf{H}$, which is parallel to $\mathbf{n} \wedge \mathbf{L}$:

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

And the rotation around an axis parallel to the 3D line is coupled with translational velocity:

$$\dot{\mathbf{n}}^T(\mathbf{n} \wedge \mathbf{L}) = \Omega^T\ \mathbf{L} + \frac{\mathbf{V}^T\mathbf{n}}{h} \tag{9}$$

That is all the information about the motion that we can draw from a 3D line, its image on a camera and the motion field of its image. In the case of two calibrated cameras, one can write the same equations for both cameras. The equations are then expressed in the same coordinate system using the calibration data ($\mathbf{R}$ and $\mathbf{T}$). In this case, for each 3D segment we may recover the angular velocity in the plane orthogonal to its direction define by $\mathbf{n} \wedge \mathbf{L}$, and $\mathbf{R}\mathbf{n}' \wedge \mathbf{L}$.

5 Motion Estimation and Covariance analysis

5.1 How to estimate the kinematic screw?

Using equations (8), and (9), for each segment we obtain the following matrix equation for the first camera:

$$\mathbf{Z} = \mathbf{F}\ \mathbf{X}$$

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

$$\mathbf{F} = \begin{bmatrix} \mathbf{n} \wedge \mathbf{L} & \mathbf{0} \\ h\mathbf{L} & \mathbf{n} \end{bmatrix} \tag{10}$$

is the transposed of the observation matrix and

$$\mathbf{Z} = \begin{bmatrix} \dot{\mathbf{n}}^T\mathbf{L} & h\dot{\mathbf{n}}^T(\mathbf{n} \wedge \mathbf{L}) \end{bmatrix}^T \tag{11}$$

defines our measurement vector. A similar equation can be obtained for the second camera. As can be observed, the input of our system consists of the 2D line parameters $\mathbf{n}$ (resp. $\mathbf{n}'$ in the other camera), their covariance matrices, and the calibration data $\mathbf{R}$ and $\mathbf{T}$, together with the stereo and temporal matchings of the 2D lines. The lines motion fields $\dot{\mathbf{n}}$ (resp. $\dot{\mathbf{n}}'$, 3D line directions $\mathbf{L}$ and their distances to the camera center h (resp. $\mathbf{h}'$) with their corresponding covariance matrices, are then estimated from the input. They complete the input to our final system of equations. Ω and $\mathbf{V}$ form the output of this system. Using a Kalman filter, we begin with an estimation of Ω and $\mathbf{V}$. In our case a reasonable assumption is that the interframe motion is small, thus we set the initial values of Ω and $\mathbf{V}$ to zero. However, to take into account our lack of information about the real motion, we set the diagonal elements of their covariance matrices to rather large values. The next section makes explicit the covariance analysis done in the kalman filtering process in our case.

5.2 Covariance analysis: Better Understanding of the Kalman filtering process

The Kalman Filtering process is very often used in computer vision. We use the estimations and covariance matrices, obtained by the kalman filtering process, to compute the Mahalanobis distances between the segments(see section 6.2). We show in this section that this is a good and justified choice for our purpose.

The behavior of a dynamic system can be described by the evolution of a set of variables, called *state variables.*

If we denote the state vector by $\mathbf{s}$ and denote the measurement vector by $\mathbf{m}'$, a dynamic system (in discrete-time form) can be described by

$$\mathbf{s}_{i+1} = \mathbf{g}_i(\mathbf{s}_i) + \mathbf{r}_i\ ,\quad i = 0, 1, \cdots\ , \tag{12}$$

$$\mathbf{f}_i(\mathbf{m}'_i, \mathbf{s}_i) = 0\ ,\quad i = 0, 1, \cdots\ , \tag{13}$$

where $\mathbf{r}_i$ is the vector of random disturbance of the dynamic system and is usually modeled as white noise: $E[\mathbf{r}_i] = \mathbf{0}$ and $E[\mathbf{r}_i\mathbf{r}_i^T] = Q_i$.

Assume that the measurement system is disturbed by additive white noise, i.e., the real observed measurement $\mathbf{m}_i$ is expressed as: $\mathbf{m}_i = \mathbf{m}'_i + \eta_i$. where

$$\begin{aligned} E[\eta_i] &= \mathbf{0}\ , \\ E[\eta_i\eta_j^T] &= \begin{cases} \mathbf{\Lambda}_{\eta_i} & \text{for } i = j\ , \\ \mathbf{0} & \text{for } i \neq j\ . \end{cases} \end{aligned}$$

When $\mathbf{g}_i(\mathbf{s}_i)$ is a linear function

$$\mathbf{s}_{i+1} = G_i\mathbf{s}_i + \mathbf{r}_i$$

and we are able to write down explicitly a linear relationship

$$\mathbf{m}_i = F_i\mathbf{s}_i + \eta_i$$

from

$$\mathbf{f}_i(\mathbf{m}'_i, \mathbf{s}_i) = 0\ ,$$

then the standard Kalman filter is directly applicable, as follows:

- Prediction of states:
  $\hat{\mathbf{s}}_{i|i-1} = G_{i-1}\hat{\mathbf{s}}_{i-1}$
- Prediction of the covariance matrix of states:
  $P_{i|i-1} = G_{i-1}P_{i-1}G_{i-1}^T + Q_{i-1}$
- Kalman gain matrix:
  $K_i = P_{i|i-1}F_i^T(F_iP_{i|i-1}F_i^T + \mathbf{\Lambda}_{\eta_i})^{-1}$
- Update of the state estimation:
  $\hat{\mathbf{s}}_i = \hat{\mathbf{s}}_{i|i-1} + K_i(\mathbf{m}_i - F_i\hat{\mathbf{s}}_{i|i-1})$
- Update of the covariance matrix of states:
  $P_i = (\mathbf{I} - K_iF_i)P_{i|i-1}$
- Initialization:
  $P_{0|0} = \mathbf{\Lambda}_{\mathbf{s}_0}$
  $\hat{\mathbf{s}}_{0|0} = E[\mathbf{s}_0]$

In our case, see section 4, $\mathbf{s}_i = \mathbf{X}_i$ and $G_{i-1} = \mathbf{I}_6$, and

$$\mathbf{F}_i = \begin{bmatrix} \mathbf{H}_i & \mathbf{0} \\ h_i\mathbf{L}_i & \mathbf{n}_i \end{bmatrix}$$

And for the initialization, we take $P_{0|0} = {\sigma_0}^2\mathbf{I}_6$ and $\hat{\mathbf{s}}_{0|0} = \mathbf{0}$ (see section 4).

Replacing these in the above Kalman filter equations, we calculate the Kalman gain matrix K_i, and the covariance matrix of states P_i:

$$K_i = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

$$P_i/{\sigma_0}^2 = \mathbf{I}_6 - \begin{bmatrix} \lambda_1\mathbf{H}_i\mathbf{H}_i^T + h_i^2\lambda_2\mathbf{L}_i\mathbf{L}_i^T & h_i\lambda_2\mathbf{L}_i\mathbf{n}_i^T \\ h_i\lambda_2\mathbf{n}_i\mathbf{L}_i^T & \lambda_2\mathbf{n}_i\mathbf{n}_i^T \end{bmatrix}$$

with $\lambda_1 = \frac{1}{1+\alpha_i^2}$ and $\lambda_2 = \frac{1}{1+h_i^2+\beta_i^2}$. Here, we suppose that $\mathbf{\Lambda}_{\eta_i}$ is a 2×2 diagonal matrix, with ${\alpha_i}^2$ and ${\beta_i}^2$ on the diagonal. This is only to simplify the symbolic calculations and does not effect the validity of our interpretations.

The matrix P_i is expressed in the camera coordinate system. Let us express this matrix in a coordinate system define by the orthonormal vectors $\mathbf{n}_i$, $\mathbf{L}_i$, and $\mathbf{H}_i$ at the same origin $\mathbf{O}$. In this coordinate system we have:

$$P_i = {\sigma_0}^2(\mathbf{I}_6 - \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & h^2\lambda_2 & 0 & h_i\lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_1 & 0 & 0 & 0 \\ 0 & h_i\lambda_2 & 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix})$$

We write also the state vector in the same coordinate system:

$$\mathbf{X}_i = [\Omega_{n_i}\ \Omega_{L_i}\ \Omega_{H_i}\ \ V_{n_i}\ V_{L_i}\ V_{H_i}]^T$$

From P_i the associated covariance matrix of $\mathbf{X}_i$, we see easily that no information is brought by segment $\mathbf{S}_i$ in the direction of $\mathbf{n}_i$ for $\mathbf{\Omega}$, and in the plane defined by $\mathbf{L}_i$ and $\mathbf{H}_i$ for the estimation of $\mathbf{V}$. Looking at $\mathbf{F}_i$ and $\mathbf{Z}_i$ (equation 11), we see that the information we have on Ω_{H_i} does not depends on depth h_i. And P_i tells us that the Kalman Filtering process is also taking it into consideration.

6 Overview of the algorithm

In this section, we outline our algorithm to solve the stereo-motion cooperation problem that arises in the context of a mobile vehicle navigating in an unknown static environment. A trinocular stereo vision system mounted on the mobile vehicle has been calibrated, and the algorithm described in this paper uses only two cameras. Our algorithm is applied to a sequence of stereo pairs obtained by the robot.Two adjacent pairs of stereo images are illustrated in figures6-7 and 8-9.

The algorithm has three different steps, see figure3. In the first step, we only track the image segments on the first and the second cameras which have been found to be the images of the same 3D lines. These segments are obtained through a stereo matching process described later in this paper. In this step we also obtain an estimation of the ego-motion of the cameras system. In the second step, we first update our 3D lines using the 3D-2D matches obtained in the first step. Then, we eliminate the stereo matches found in the first step and use epipolar constraint to find the stereo pairs of lines which come in the field of view. In this way, we not only update the available 3D line, but also we find the new stereo pairs which appear in the field of view of the both cameras. In the third step, we use the estimated ego-motion for the tracking of the rest of line segments on each camera. That gives us the temporal matchings on each camera. Finally, we use the results of the three steps of the algorithm to complete our three-dimensional reconstruction of the robots environment. Now, we explain in more detail the three different steps of the algorithm.

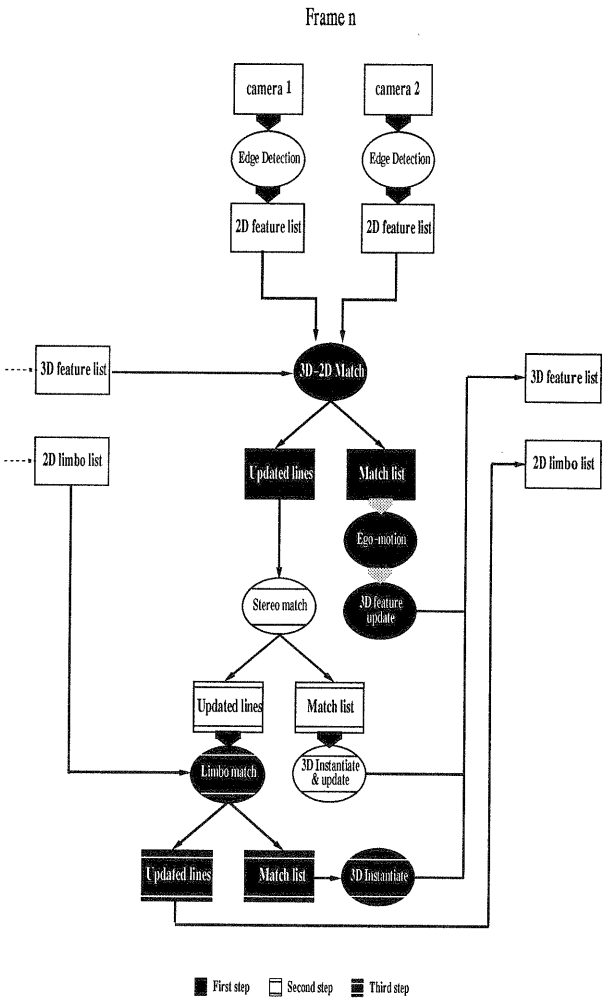

Fig. 3.

6.1 First step: Temporal matching of stereo pairs

Suppose that at time t_1, we have a set of 3D line segments, which are reconstructed based on all stereo pairs available up to t_1. The reconstruction technique will be described later. In the initialization, the stereo algorithm described in [3] is used to obtain the first set of 3D line segments. We have also an initial estimate of motion between t_1 and t_2. Under the assumption of smooth motion, this motion estimation can be derived using information obtained by analysing previous stereo pairs. In the initialization phase, the motion estimate can be obtained from the odometric system of the mobile vehicle. In our implementation, since the interframe motion is assumed to be small, motion parameters are initialized to zero (i.e., no motion), but with a big covariance (i.e., very uncertain). That makes the initialization phase much more difficult.

First of all, the stereo reconstructions help us to choose the pairs of segments which correspond to the 3D lines closer to the camera. These pairs of stereo matches give us more robust depth information and once they have been tracked, they give us also more robust motion information. We then apply the estimated motion to such 3D lines and project them on the cameras. In figure 4, $D(t_1)$ is an example of such 3D lines and $D(t_2)$ is its transformed line. The dashed lines on the cameras are its projections.

We then find the nearest 2D segments to those projected segments in the next images. To compute the distance between reprojected segments and the next image

segments we use the Mahalanobis distance to take account of uncertainty measurements. We will give more details later in this article. To reduce the research area for neighboring segments we bucketize the pair of images at t_2.

The distance of a projected segment to its nearest neighbors is one of the criteria we use to choose the best matches. Another criterion is what we call the confidence factor, which is defined as a function of the relative distance between the two nearest neighbors and represents the rate of matching disambiguity.

Using these three criteria: distance to the cameras, distance of the reprojected segment to its nearest neighbor in the next image, and the confidence factor of the corresponding match, we find the best temporal matches. At this point we use again the fact that we are tracking pairs of stereo segments and verify if the candidates for temporal matching in two cameras *verify also the epipolar constraint.*

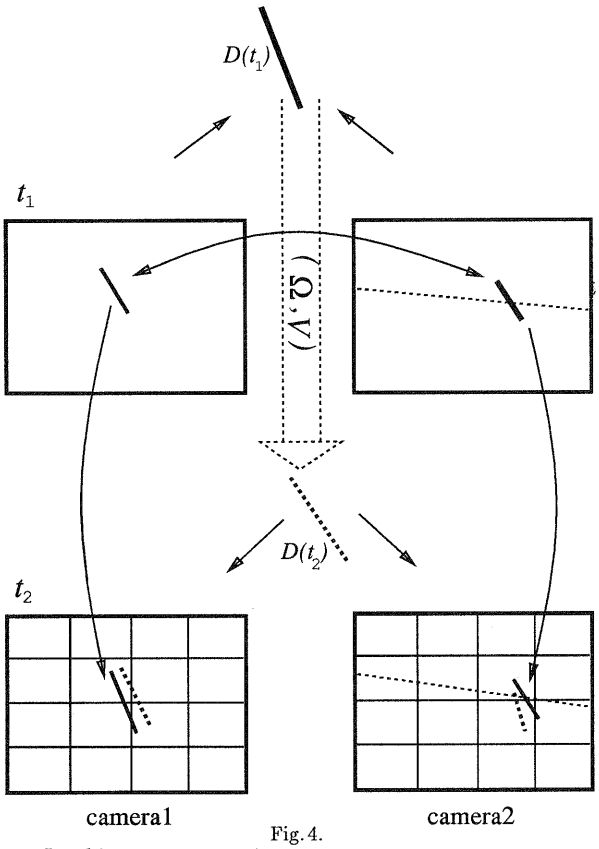

Fig. 4.

In this way, even when the estimated motion is uncertain, we usually find a few pairs of correct temporal matches, which are sufficient to compute a better estimation of motion parameters.

To update the motion estimation we use a modified version of the algorithm presented in [19] and [9]. The input of this algorithm, other than reconstructed 3D lines, consists of the 2D velocities of their images on the cameras. Here we estimate these velocities using two consecutive images (see section 4). Assuming that we have several (at least two) non parallel 3D lines undergoing same three-dimensional motion, this algorithm allows us to recover the full kinematic screw of the rigid object they are attached to. Since we keep track of uncertainty at all levels, a weighted least-squares minimization which can be implemented with the Kalman filter is used. Therefore even if there exist a few incorrect matches we can still obtain a reasonable motion estimation.

Now we begin the next iteration. At this time we apply the estimated motion to each reconstructed segment, reproject it on to the cameras and find their nearest neighbor distances and confidence factors. Based on these values we decide if we have two good temporal matches and if so, we then verify the epipolar constraint between the two candidates. Finally if the epipolar constraint is also verified we use this new matches to update the motion estimation.

Experimental results shows that even if we track a few pair of segments (3 or 4 pairs) in the first iteration, in the second and the third one we are able to track almost all stereo pairs of segments . If in the first set of tracked segments there are some incorrect matches they will be corrected in the following iterations.

6.2 Temporal matching of stereo pairs

To track a pair of stereo matching line segments in each camera:

1) We reconstruct 3D lines. We apply the estimated motion to them and we get 3D lines predicted for the next instant which are projected on the cameras.
2) We find the nearest segment to the projected segment in the next image and choose it as a candidate for our temporal matching process. To do that :
   - Images are bucketized and we search for the neighboring segments only in the buckets which intersect the projected line segment. In this way many useless computations are avoided and the algorithm is more efficient.
   - Mahalanobis distance is used to measure the distance of two line segments. It is very important, because the projected line segments are obtained from the stereo reconstruction process and the application of an estimated motion. Through the error analysis of these two process we are able to determine a covariance matrix associated to the projected line segments..
     For two vectors S_1 and S_2 with the respective covariance matrices Λ_1 and Λ_2, their Mahalanobis distance is defined as:
     $$D(S_1, S_2) = (S_1 - S_2)^T(\Lambda_1 + \Lambda_2)^{-1}(S_1 - S_2)$$
     We can simply interpret $D(S_1, S_2)$ as the square of the Euclidean distance between S_1 and S_2 weighted by the sum of their covariances.

- We then define a confidence factor for each temporal match. It is defined as a function of the relative distance of the projected segment to its first and second nearest neighbors. Suppose that S_2 and S_3 are respectively the first and second nearest neighbors of S_1. The confidence factor c is defined as:
  $$c = \frac{D(S_1, S_3) - D(S_1, S_2)}{D(S_1, S_3)}$$
  If the confidence factor of a temporal matching is small (in our implementation less than 0.8), its means that there is an ambiguity and we reject the candidate even if its distance to its nearest neighbor is very small.
- Finally, for the pairs of the candidate segments obtained through temporal matching, we check if they verify the epipolar constraint. It means that we compute the epipolar line of the midpoint of one segment (for example in the first camera) and verify if it intersects the other segment (in the second camera). If the epipolar constraint is also verified, we confirm these temporal matches.

6.3 Second Step: Stereo matching

In the initialization phase of the algorithm we use a trinocular hypothesis-verification stereovision algorithm [3] to find the stereo matches.

As the number of segments in cameras field increases, these kinds of algorithms become very time consuming. Temporal matching is less expensive. Particularly, when the interframe motion is small and therefore, we can reduce the search area. As the cameras system moves, in the first step, we try to track the initial stereo matches in each camera . Obviously at each step some new segments enter the cameras visual fields and some others may have not been tracked. Therefore to add new coming segments to our set of stereo matchings, we run again a classical hypothesis-verification algorithm , see [3], only on those few segments. It is not very costly because of the reduced number of hypotheses generated.

6.4 Third Step: temporal matching

Once a part of the stereo pairs of the segments have been temporally tracked, in this step we try to track the other 2D line segments. We use the results of the first step in two different ways. First we mark the segments which take part of the stereo-temporal matching segments. Then we use the estimated motion obtained in first step to obtain the temporal matchings of the 2D limbo line segments, on each camera.

Here, we explain our temporal matching algorithm on one of the camera. For the extremities of each segment S_{t_i}, taken at time t_i, we draw the corresponding epipolar lines in the image taken by the same camera at time t_{i+1}. As we have an estimation of the motion of the camera between t_i and t_{i+1}, we can consider them as a pair of stereo camera. The length of the segment, obtained through edge detection and polygonal approximation processes, is not reliable. Therefore, we do not expect that the extremities of the segment $S_{t_{i+1}}$ temporal matching of S_{t_i}, belong to these epipolar lines. We define a function $F(S_{t_i}, S_{t_{i+1}})$ which measures the goodness of a temporal matching $(S_{t_i}, S_{t_{i+1}})$.

Suppose e_1 and e_2 be the corresponding epipolar lines of the two extremities of S_{t_i}. F is defined as follows:

$$F(S_{t_i}, S_{t_{i+1}}) = \alpha \frac{d_1 + d_2}{l_{S_{t_{i+1}}}} + \beta \frac{\angle(S_{t_i}, S_{t_{i+1}})}{\angle max}$$

where d_1 and d_2 are defined as in the figure6.4, and $l_{S_{t_{i+1}}}$ defines the length of the segment $S_{t_{i+1}}$, and in our experiments $\alpha = \beta = 1$ and $\angle max = \frac{\pi}{6}$.

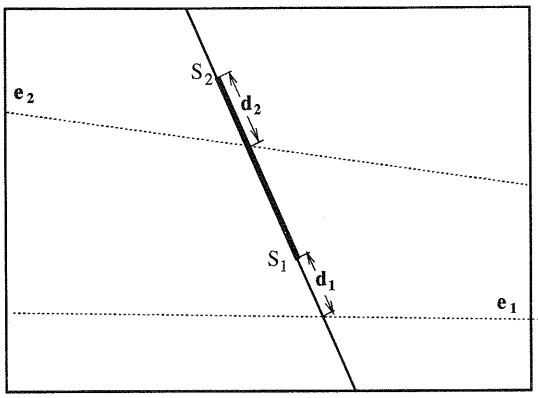

Fig. 5.

The second term of the function $F(S_{t_i}, S_{t_{i+1}})$ is to take into consideration that after a small motion there is only a small change in the direction the 2D line segments. More details on this subject can be found in [15].

7 Results

We have used several sequences of stereo images obtained by our mobile robot. The baseline is about 43mm, the focal length, 8mm and the pixel size, $8x14\mu m^2$. The distance between the objects and the cameras varies from 2m to 5m. In the experiments presented here, robot rotates 5.0 degrees around its vertical axis and moves forward 15cm, at each step. Experimental results are shown in figures6-13.

Figures 6-7 and 8-9 display images taken by the first and the second cameras, at t_1 and t_2 respectively. The results obtained at different steps of the algorithm are shown in figures 10-13. Figures Figures10-11 show the stereo-temporal matching segments after one iteration of the first step of the algorithm. We also apply the estimated motion to the 3D data obtained at t_1 and show their projections on the first camera (black segments) overlayed on the image taken at t_2 for comparison. Temporal matchings (white segments) are used to update the motion estimation. Figures 12-13 show the result of the third iterations. The motion estimation is improved and we track almost all the segments. After the second and the third steps of the algorithm almost all the 2D line segments are correctly matched. Due to the large number of tracked segments on each camera, it is not easy to visualize the results in black and white. Therefore, the results on the second and the third steps of the algorithm are presented, using the color slides, during the conference.

8 conclusion

We have presented a unified and iterative algorithm for the fusion of the visual data based on the dynamic cooperation between stereo matching and temporal matching processes. This cooperation is robust and less time consuming than doing the classical stereo reconstruction at each step of the motion of a mobile robot and the results are quite satisfactory. As we use all segments to estimate the kinematic screw the method only works if all segments considered actually belong to the same rigid object, otherwise it fails. A solution for the multiple objects motion analysis based on the stereo-motion cooperation is given in [16].

References

[1] J.K. Aggarwal and Y.F. Wang. Analysis of a sequence of images using point and line correspondences. In *Proc. Int'l Conf. Robotics Automation*, pages 1275–1280, Raleigh, NC, March 31–April 3 1987. IEEE.

[2] J. Arspang. Direct scene determination: Local relative or absolute surface depth, geometry and velocity from monocular and multi ocular image sequences. Technical Report 88/3, Computer Science Department, University of Copenhagen, January 1988.

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

[4] T.J. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-D motion estimation from a monocular image sequence. *IEEE Trans. AES*, 26(4):639–656, July 1990.

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

[6] O.D. Faugeras and G. Toscani. The calibration problem for stereo. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pages 15–20, Miami, FL, June 1986. IEEE.

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

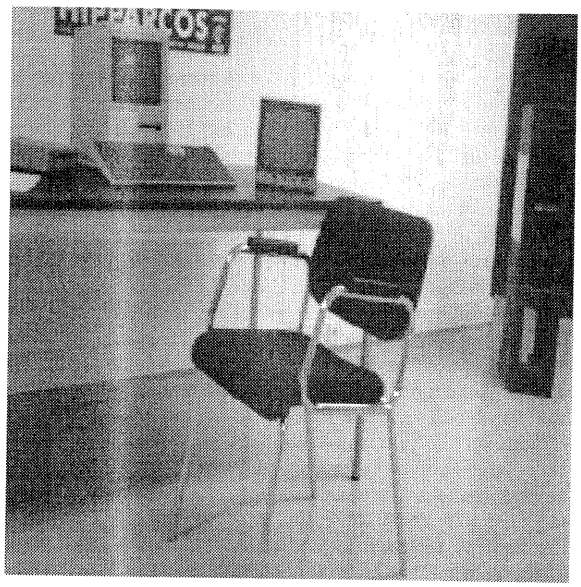

Figure 6:

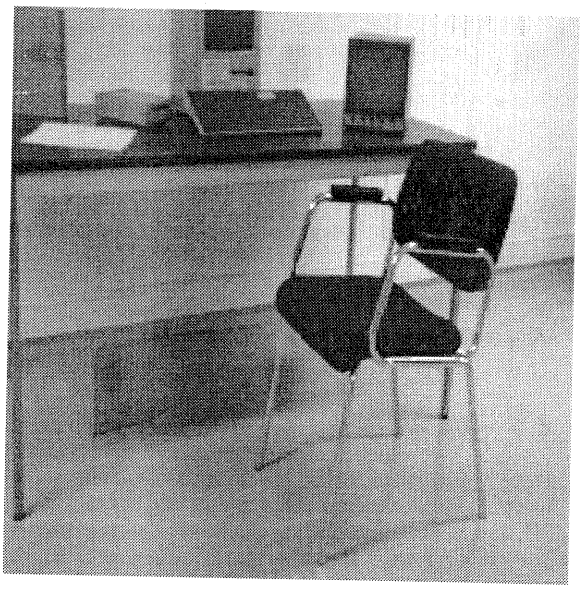

Figure 7:

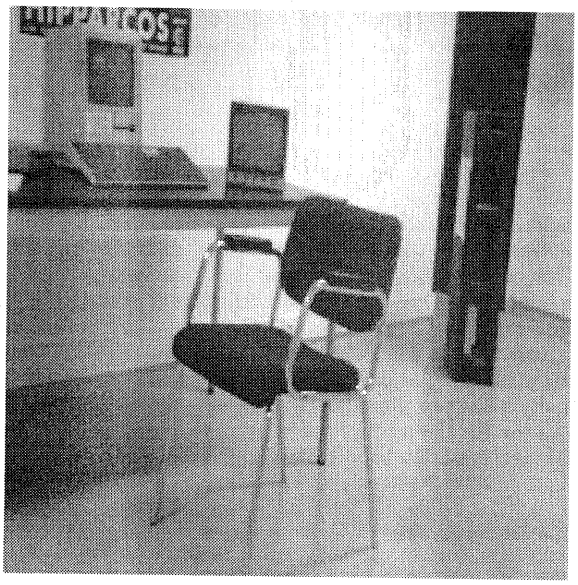

Figure 8:

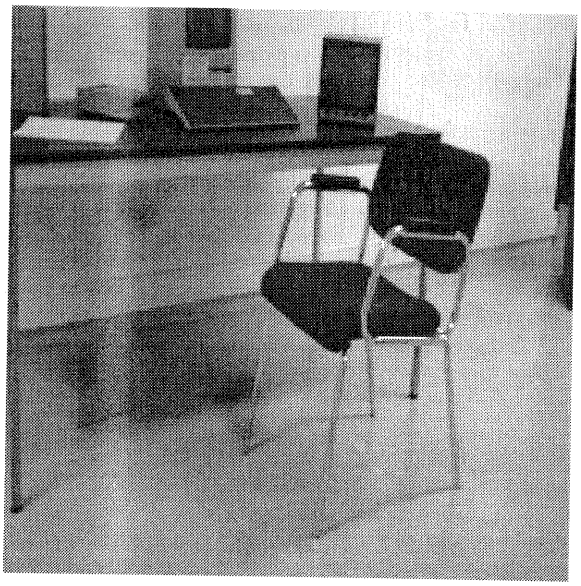

Figure 9:

[8] 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.

[9] Olivier D. Faugeras, Nassir Navab, and Rachid Deriche. On the information contained in the motion field of lines and the cooperation between motion and stereo. *International Journal on Imaging Systems and Technology*, 2:356–370, 1990.

[10] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conf.*, pages 189–192, 1988.

[11] K. Henriksen. *Projective Geometry and straight lines in Computational Vision.* PhD thesis, Computer Science Department, University of Copenhagen, March 1990.

[12] Thomas S. Huang and Arun N. Netravali. Motion and Structure from Feature Correspondences: A Review. *proc. IEEE*, 1992. to appear.

[13] Y. Liu and T.S. Huang. A linear algorithm for determining motion and structure from line correspondences. *Comput. Vision, Graphics Image Process.*, 44(1):35–57, 1988.

[14] D. Charnley E.P. Sparks M. J. Stephens, R.J. Blissett and J.M. Pike. Outdoor vehicle navigation using passive 3d vision. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pages 556–562, San Diego, CA, 1989. IEEE.

[15] N. Navab. *Motion of lines, and the cooperation between motion and stereo.* Dissertation, University of Paris XI, Orsay, Paris, France, 1992. in English, To appear.

[16] N. Navab and Z. Zhang. From multiple objects motion analysis to behavior-based object recognition. In *Proc. ECAI 92*, Vienna, Austria, August 1992. To appear.

[17] N. Navab and Z. Zhang. A stereo and motion cooperation approach to multiple objects motion problems. In *Proc. 2nd Singapore International Conference on Image Processing ICIP*, Singapore, September 1992. To appear.

[18] N. Navab, Z. Zhang, and O.D. Faugeras. Tracking, motion and stereo: A robust and dynamic cooperation. In *Proc. Scandinavian Conf. Image Analysis*, pages 98–105, Aalborg University, Denmark, August 1991.

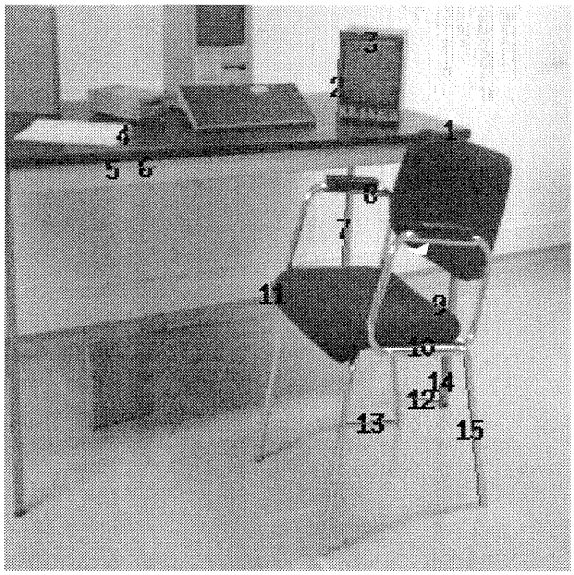

Figure 10:

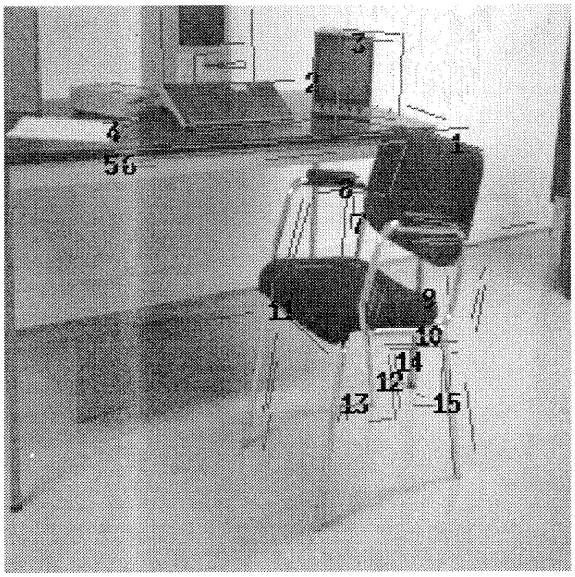

Figure 11:

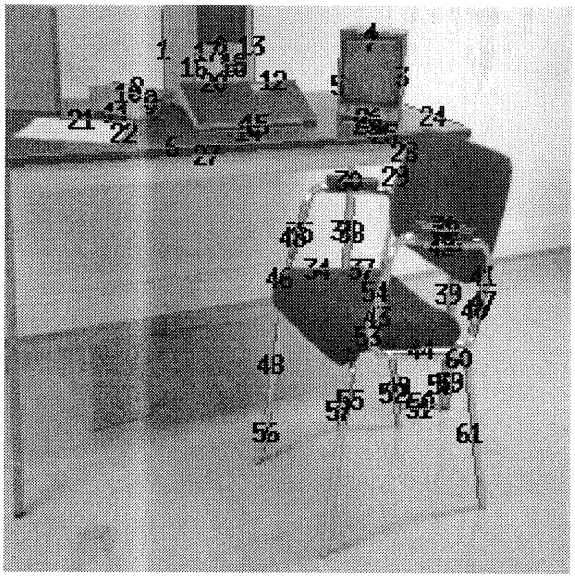

Figure 12:

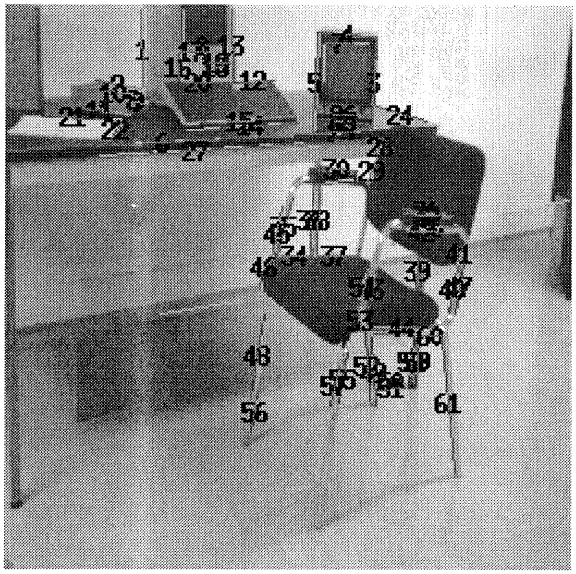

Figure 13:

[19] 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*, Osaka, Japon, December 1990. IEEE.

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

[21] A. Tamtaoui and C. Labit. Coherent disparity and motion compensationin 3dtv image sequence coding schemes. In *Proc. ICASSP'91*, Canada, May 1991.

[22] A. Tamtaoui and C. Labit. Tv3d: joined identification of global motion parameters for stereoscopic sequence coding. In *Proc. VCIP'91*, Boston, USA, 1991.

[23] N. A. Thacker, Y. Zheng, and R. Balckbourn. Using a combined stereo/temporal matcher to determine ego-motion. In *Proc. British Machine Vision Conf.*, pages 121–126, University of Oxford, London, UK, September 1990.

[24] R. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pages 364–374, Miami, FL, June 1986. IEEE.

[25] T. Viéville and O. Faugeras. Feed-forward recovery of motion and structure from a sequence of 2D-lines matches. In *Proc. Third Int'l Conf. Comput. Vision*, pages 517–520, Osaka, Japan, December 1990.

[26] Z. Zhang and O.D. Faugeras. Estimation of displacements from two 3D frames obtained from stereo. *IEEE Trans. PAMI*, September 1992. to appear.

[27] Zhengyou Zhang. *Motion Analysis from a Sequence of Stereo Frames and its Applications.* Dissertation, University of Paris XI, Orsay, Paris, France, 1990. in English.