# A 3D world model builder with a mobile robot

Zhengyou Zhang, Olivier Faugeras

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° 1546

*Programme 4*
*Robotique, Image et Vision*

# A 3D WORLD MODEL BUILDER WITH A MOBILE ROBOT

**Zhengyou ZHANG**
**Olivier D. FAUGERAS**

**Novembre 1991**

*RR-1546*

# A 3D World Model Builder with a Mobile Robot*†

# Construction d'un modèle 3D du monde avec un robot mobile

**Zhengyou Zhang** **Olivier D. Faugeras**

INRIA Sophia-Antipolis
2004 route des Lucioles
06565 Valbonne Cedex, France
zzhang@sophia.inria.fr faugeras@sophia.inria.fr

*This work was supported in part by Esprit project P940.

†To appear in *The International Journal of Robotics Research*, 1992.

## Abstract

This article describes a system to incrementally build a world model with a mobile robot in an unknown environment. The model is, for the moment, segment-based. A trinocular stereo system is used to build a local map about the environment. A global map is obtained by integrating a sequence of stereo frames taken when the robot navigates in the environment. The emphasis of this article is on the representation of the uncertainty of 3D segments from stereo and on the integration of segments from multiple views. The proposed representation is simple and very convenient to characterize the uncertainty of segments. A Kalman filter is used to merge line segments matched. An important characteristic of our integration strategy is that a segment observed by the stereo system corresponds only to one part of the segment in space, so the union of the different observations gives a better estimate on the segment in space. We have succeeded in integrating 35 stereo frames taken in our robot room.



**Keywords**: Uncertainty Representation, Multiple View Integration, World Model Builder, 3D Vision, Mobile Robot.



## Résumé

Nous décrivons un système capable de construire, au fur et à mesure, un modèle du monde avec un robot mobile dans un environnement inconnu. Le modèle est pour le moment basé sur des segments de droite. Un système trinoculaire est utilisé pour construire une carte visuelle locale de l'environnement. Une carte globale est obtenue en intégrant une séquence de scènes stéréoscopiques acquises quand le robot se déplace dans l'environnement. Nous insistons dans cet article sur la représentation de l'incertitude des segments de droite 3D obtenus à partir de la stéréo et sur l'intégration de segments de droite de différentes vues. La représentation que nous proposons est simple et très pratique pour caractériser l'incertitude de segments. Un filtre de Kalman est utilisé pour fusionner des segments de droite appariés. Une caractéristique importante de notre stratégie d'intégration est la suivante: un segment observé par un système stéréoscopique correspond seulement à une partie du segment dans l'espace, donc l'union des observations différentes donne une meilleure estimation du segment dans l'espace. Nous avons réussi à intégrer 35 vues stéréoscopiques prises dans la salle du robot.



***Mots clés:*** Représentation d'incertitude, Fusion de vues différentes, Construction d'un modèle du monde, Vision 3D, Robot mobile.

# Contents

# List of Figures

# 1 Introduction

Mobile robotics is an active field of research. Using vision feedback to guide autonomously a mobile vehicle is an interesting and difficult subject of research. However, several research results show already some potential applications. A useful task of a mobile robot is to automatically build a model of an unknown environment. In this article, we present a system, called the *world-model builder*, which can accomplish this task. Since most parts of the system have been described elsewhere, we concentrate, after a brief description of the system, on how to fuse multiple 3D frames obtained by a stereo system.

The world model is for the moment based on 3D line segments, reconstructed by the trinocular stereovision system described in (Lustman, 1987; Ayache, 1991). Our world-model builder uses the take-and-look strategy: based on the information from stereo, it decides where is the interesting space to explore, plans the trajectories, navigates, and updates the global model of the environment with the currently observed stereo frame. This world model may be used later for navigation or recognition. Besides stereovision, this system involves several important issues in Robotics and Vision:

1. Matching consecutive 3D frames and estimating motion from 3D data,
2. Finding areas of interest for further exploring,
3. Planning trajectories and controlling the robot,
4. Fusing 3D data obtained at different instants,
5. Interpreting and possibly editing 3D data.

There are many methods proposed in the literature to register two consecutive 3D frames, for example, Maximal tree-search matching (Chen and Huang, 1988; Chen and Huang, 1987), Hypothesize-and-verify (Faugeras et al., 1988; Zhang et al., 1988), and Relaxation (Kim and Aggarwal, 1987). There exist quite a number of methods (linear or nonlinear, analytical or numerical) to determine motion from feature correspondences (Ayache and Faugeras, 1987; Blostein and Huang, 1984; Faugeras and Hebert, 1986; Kim and Aggarwal, 1987). A comparative study of several methods for motion determination is found in (Zhang and Faugeras, 1991). The solution to the second problem is usually goal-dependent. How to displace the robot with minimum energy (or other criteria) while avoiding stationary or moving obstacles is a challenging subject of research. Some work related to this issue includes (Lozano-Pérez and Wesley, 1979; Tsuji and Zheng, 1987; Tournassoud, 1988). A simple example of trajectory planning for navigation can be found

in (Zhang and Faugeras, 1989). Data fusion, i.e., how to model sensor noise and how to integrate optimally data from multiple sensors or from the same sensor at different instants, has attracted many researchers (Faugeras et al., 1986; Ayache and Faugeras, 1987; Hager, 1988; Porrill, 1988; Durrant-Whyte, 1988a). Finally, 3D data should be organized and interpreted (possibly edited) by using some *a priori* geometric constraints in order to obtain a consistent and higher-level representation (Thonnat, 1988; Grossmann, 1989).

The problem of integrating a sequence of *monocular* views to build a global map was addressed by Jezouin and Ayache (Jezouin and Ayache, 1990), where the motion is assumed given with very good accuracy by inertia sensors. A sequence of images of an outdoor scene generated by a realistic image synthesis system was used, and they were able to reconstruct primitives (points and line segments) of buildings at a distance of several thousands of meters with an accuracy of a couple of meters. As pointed by the authors, the accuracy of 3D reconstruction depends heavily on camera motion.

The problem of integrating sonar and stereo range data was addressed by Matthies and Elfes (Matthies and Elfes, 1988). They used a cellular representation called the *Occupancy Grid* to describe the vicinity of a robot. Range information from sonar and one-dimensional stereo is combined into such a 2D map. Each cell in the map contains a probabilistic estimate of whether it is empty or occupied by an object in the environment. These estimates are obtained from sensor models that describe the uncertainty in the range data. A Bayesian estimate scheme is applied to update the current map using successive range readings from each sensor. The occupancy grid representation is suitable for robot navigation application. It is, however, very limited in its descriptive ability if we want to interpret and understand the 3D environment of a robot.

A similar system to ours, called the *3D Mosaic* scene understanding system, was designed by Herman at CMU (Herman, 1986). A comment to this system can be found in the conclusion section.

In the remainder of this article, we first present our system. After a brief explanation of each module of the system, our emphasis is on the representation of the uncertainty of 3D line segments from stereo and on the fusion of line segments measured at different instants. Two experimental results are provided. The first one is the fusion of two stereo views, which will give us a good feeling of the performance of the fusion process. The second shows the result of fusing 35 stereo views of a room.

# 2 System Description

An intelligent robot system (Brady, 1985) can be decomposed into three parts: Perception, Intelligent Decision and Control, and Action. Our system is being developed on the INRIA

mobile robot (see Fig. 1). It is dedicated to navigating in an indoor environment and to making visual maps of this environment. The *perception* tools include 3 cameras and 24 ultrasound cells. The three cameras form a trinocular stereovision system providing a local 3D map of the environment. Ultrasound is, for the moment, used only to avoid obstacles. The *action* is to displace the robot and is realized by wheels. The robot is equipped with two driving wheels and two passive rollers. Each of the two driving wheels is independently controlled by a motor, so that the robot can translate and rotate. The tasks of the *intelligent decision and control system* include finding points of interest and planning trajectories. Since the coordinate systems of the perception and action components are different, these coordinate systems should be first calibrated (Zhang and Faugeras, 1989).

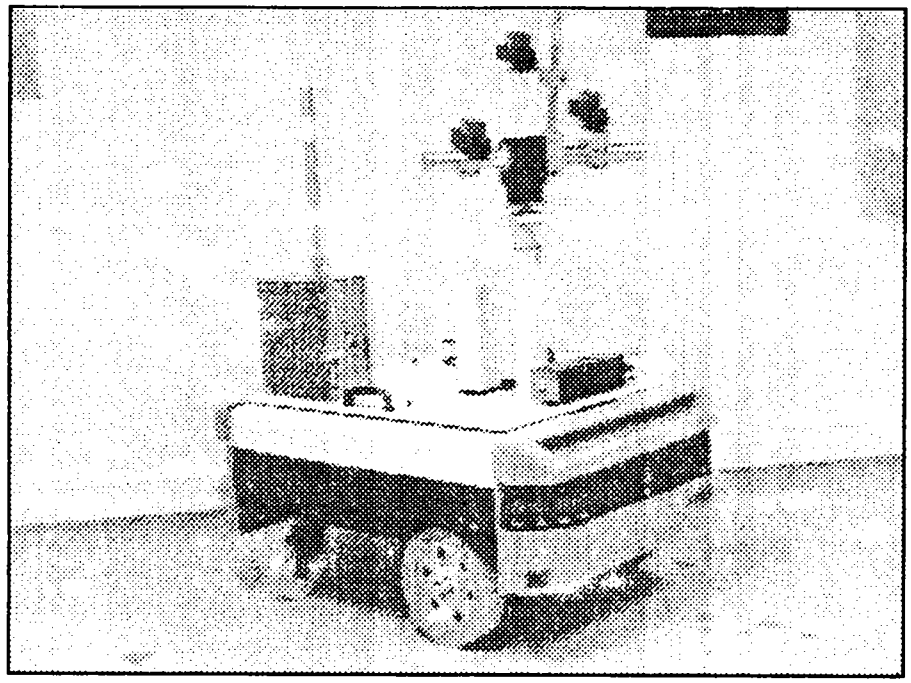


**Fig. 1:** The INRIA mobile robot

Figure 2 shows the architecture of our world model builder. As indicated by the name, the goal of our system is to incrementally build a model of its environment. We, for the moment, restrict the domain to the indoor environment. The system is simply composed of several individual modules. First, the *Stereovision Module* (**SM**) (Lustman, 1987; Ayache, 1991) builds a local visual map which is a set of 3D segments. The *Analysis and Decision Module* (**ADM**) then analyzes the local map. By combining the information from the *Data Fusion Module* (**DFM**) and *Interpretation Module* (**IM**), the **ADM** decides where the robot should go in order to explore further the environment and to obtain more precise data. This

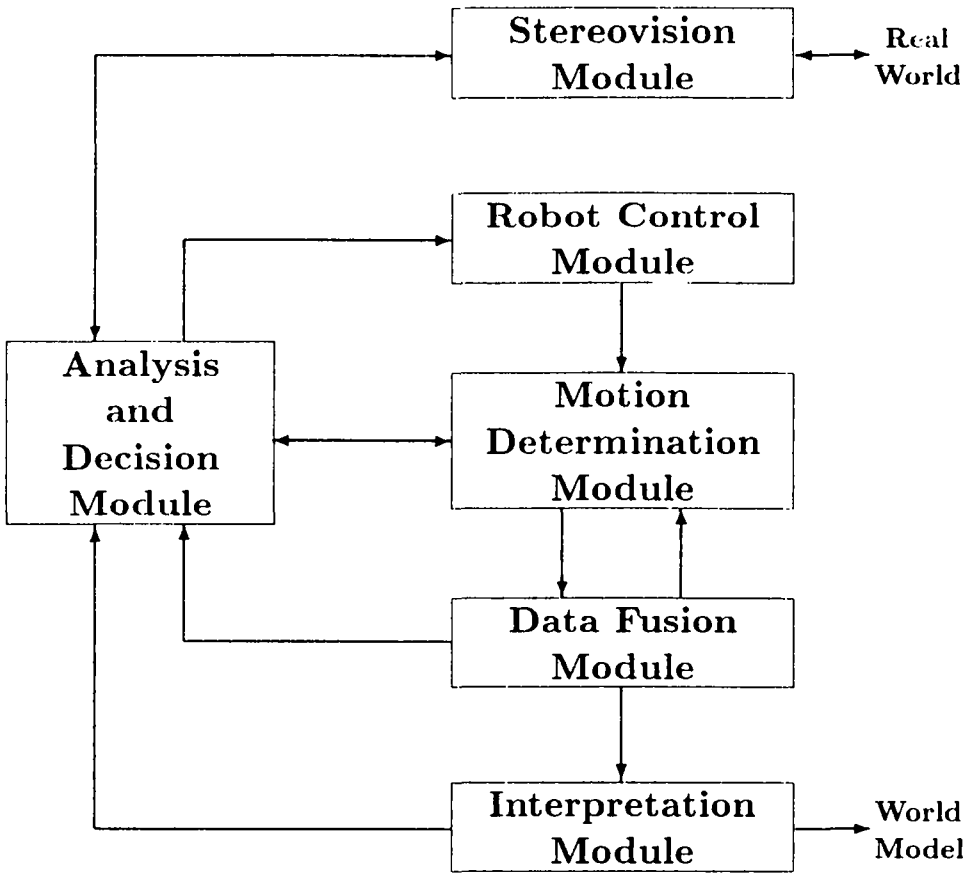



**Fig. 2:** System architecture

module is being under development (Buffa et al., 1990). For the moment, it is the user who accomplishes its task with the aid of a graphic interface. The graphic interface displays the projection of the 3D segments on the ground plane, the borders of objects recovered by making the Delaunay triangulation on the projected data, and also the position of the robot. The borders are updated when new observations become available. It is then sufficient for the user to click the mouse on the next preferred position. The following three strategies will be implemented in the system:

— **If** segments of an object are not accurate enough,
**and** the space between the object and the robot is free,
**then** move the robot ahead.
— **If** there is room for the robot to rotate $\theta$ degrees to the right (or left),
**then** rotate the robot to the right (or left) $\min(\theta, \Theta/4)$ degrees.
($\Theta$ is the field of view of the stereovision system)
— **If** the angle between the optical direction $OZ$ and the object to observe $MN$

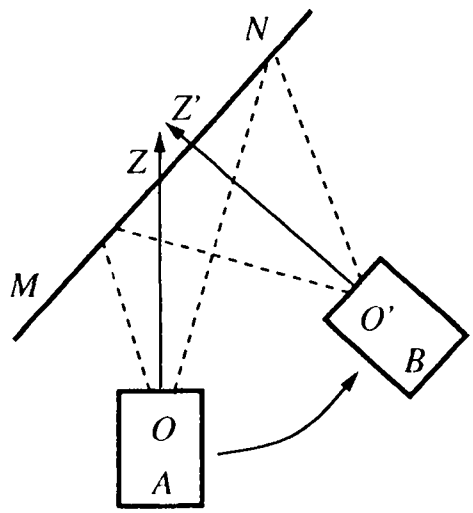



**Fig. 3:** Illustration of a planning strategy: the robot is represented by a rectangle

is small enough (less than 45 degrees, for example),
**then** move the robot to the front of the object. (see Fig. 3)

These strategies are motivated by the need to reduce the uncertainty due to the stereo triangulation. The first two are easy to understand. The situation corresponding to the third one is displayed in Fig. 3. The robot is found in the position $A$ and should move to the position $B$. There are two reasons for this. The first is due to the fact that the cameras are mainly calibrated for the central parts. The reconstructed 3D information is less reliable in the border than in the center. This effect is more severe in position $A$ than in position $B$. The second reason is that when the angle between $OZ$ and $MN$ is small, a part of the observed object is far from the camera and the reconstructed 3D information is not very useful.

The **ADM** module then gives a command of displacement to the *Robot Control Module* (**RCM**). Since the command of displacement is related to the stereo coordinate system and the **RCM** works in the odometric one, we should first calibrate the two coordinate systems (Zhang and Faugeras, 1989). When the **RCM** module receives a command of displacement, it plans the trajectories, as smooth as possible, to reach the goal. The ultrasound sensors are used to avoid local obstacles on the trajectory (Robles, 1988). The real trajectories are displayed by the graphic interface.

The task of the *Motion-Determination Module* (**MDM**) is to match successive 3D frames and to obtain a precise estimate about the robot displacement. The motion-determination algorithm is based on hypothesize-and-verify paradigm (Faugeras et al., 1988; Zhang et al., 1988). In the first stage, the rigidity constraint is heavily used to generate hy-

potheses of matches between two successive frames. Two pairings of segments constitute a plausible hypothesis if they satisfy the constraints on length, angle of two segments and distance of midpoints, and the constraint on the angle between one of the segments and the line joining the midpoints. All thresholds in the constraints are determined dynamically using the error estimates (covariance matrices) of the parameters of 3D segments. In the second stage, we propagate each hypothesis to the whole frame. We first obtain an initial estimate of displacement from each hypothesis using the iterative extended Kalman filter (Jazwinsky, 1970; Faugeras et al., 1986; Ayache and Faugeras, 1989). We apply this estimate to the first frame and compare the transformed frame with the second frame. If a transformed segment of the first frame is similar enough to a segment in the second frame, this pairing is considered as matched and the extended Kalman filter is again used to update the displacement estimation. A special treatment on the similarity of two segments is discussed in (Zhang et al., 1988). After all segments have been processed, we obtain, for each hypothesis, the optimal estimate of motion, the estimated error given by the filter and the number of matches. Once all hypotheses are evaluated, the hypothesis which gives the minimal estimated error and the largest number of matches is considered as the best one, and its corresponding optimal estimate is retained as the displacement between the two frames.

This algorithm does not use any information about motion such as that from the odometric system. Indeed, the odometric system can give us approximately the rotation angle and the relative position between two successive views. Since we have already calibrated the mobile robot, we can transform the motion estimate from odometric system to stereo frame. We then skip over the first stage and enter directly the hypothesis verification phase. This new technique can speed up considerably the motion-determination process. Typically (each frame contains about 150 line segments), if we use the general algorithm, about ten hypotheses are generated, and the user time spent in the whole process is about 70 seconds on a SUN 3/60 workstation. If we use the new technique, the user time is only about 9 seconds. This is why there is in Figure 2 an arrow from **RCM** to **MDM**. Later, the initial estimate of motion may come directly from the **ADM**.

Motion estimation and segment correspondences are then provided to the *Data Fusion Module*. Section 4 describes in detail this module. The fused data are fed back to the *Motion-Determination Module* to track the position of the robot at the next instant. The fused data are also supplied to the *Analysis and Decision Module* and the *Interpretation Module*.

Now we have only a set of 3D line segments. It is not useful enough for navigation or object recognition. 3D line segments should be organized and interpreted in order to obtain a higher level representation. Since the 3D line segments we have are still noisy, we can impose some *a priori* geometric constraints on them to reduce the uncertainty as in (Porrill, 1988).

All these are the tasks of the *Interpretation Module*. This module is being developed. We have implemented some primitive procedures: parallelism of segments, perpendicularity of segments, coplanarity of segments, etc. We also notice several researchers working on this subject (Thonnat, 1988; Grossmann, 1989).

# 3 Representation of 3D Line Segments

Before proceeding further, we briefly describe in this section how to represent 3D line segments. The 3D line segments we have (from stereo or other sensors) are inherently uncertain and usually do not have the same error distribution in different directions. It has been recognized in the Computer Vision and Robotics community (Durrant-Whyte, 1988b; Ayache and Faugeras, 1989) that uncertainty should be explicitly represented and manipulated. Two representations of uncertainty in 3D reconstruction from stereo are available. Blostein and Huang (Blostein and Huang, 1987) assumed that the pixel errors were uniformly distributed and the 3D points reconstructed by stereo triangulation were also assumed to be uniformly distributed in the corresponding volume. Instead, Matthies and Shafer (Matthies and Shafer, 1987) modeled the stereo triangulation errors as three-dimensional Gaussian distributions and claimed that it gave good results in stereo navigation when the distance to points is not extreme (see also (Ayache and Faugeras, 1989)). The later modelization makes possible efficient and tractable computation. For the same reason, we also model measurement errors as Gaussian. However, we should remember that this assumption can only be justified for small random errors, but not for gross or systematic errors.

## 3.1 Motivation

A line segment in 3D space is usually represented by its endpoints $M_1$ and $M_2$, which requires 6 parameters, and their covariance matrices $\Lambda_1$ and $\Lambda_2$. $\Lambda_1$ and $\Lambda_2$ are estimated by stereo triangulation from point correspondences (Lustman, 1987; Ayache, 1991). Equivalently, a line segment can be represented by its direction vector $\mathbf{v}$ and its midpoint $M$, and their covariance matrices $\Lambda_{\mathbf{v}}$, $\Lambda_M$, and cross-correlation matrix $\Lambda_{\mathbf{v}M}$. The relation between them is simple:

$$\begin{aligned} &\mathbf{v} = M_2 - M_1, \quad M = (M_1 + M_2)/2, \\ &\Lambda_{\mathbf{v}} = \Lambda_1 + \Lambda_2, \quad \Lambda_M = (\Lambda_1 + \Lambda_2)/4, \quad \Lambda_{\mathbf{v}M} = (\Lambda_2 - \Lambda_1)/2. \end{aligned} \tag{1}$$

But we cannot use directly these parameters in most cases. The endpoints or the midpoint

of a segment are not reliable. The reason for this is that the way the uncertainty of the endpoints of a three-dimensional segment is computed takes only into account the uncertainty in pixel location due to edge detection and the uncertainty in the calibration of the stereo rig (Ayache and Faugeras, 1989). But it does not take into account the uncertainty due to the variations in segmentation of the polygonal approximation process. There are two main sources for those variations. The first one is purely algorithmic: because of noise in the images and because we sometimes approximate curved contours with straight line segments, the polygonal approximation may vary from frame to frame inducing a variation in the segments endpoints. The second is physical: because of partial occlusion in the scene, a segment can be considerably shortened or lengthened and the occluded part may change over time.

Thus, instead of the line segment, the infinite line supporting the segment is usually used, as in (Kim and Aggarwal, 1987). In a previous version of our algorithm for displacement analysis from two stereo views (Ayache and Faugeras, 1987; Faugeras et al., 1988; Zhang et al., 1988), a line segment was treated in a mixed way. The infinite supporting line was used in estimating the displacement, and the line segment was used for matching. There has been a lot of representations proposed for a line in the literature (Ayache and Faugeras, 1989; Roberts, 1988). The main problem is that **the uncertainty on the line parameterization does not reflect that of the segment which the line supports** (see (Zhang, 1990) for more details). A segment with big uncertainty may result in small uncertainty in the line parametrization. In the next subsection, we describe a representation for 3D line segments taking into account the variations in segmentation.

## 3.2 Our Representation

Because of the deficiencies of the previous representations for a line or a line segment, we use a five parameter representation for a line segment: two for the orientation, and three for the position of a point on the segment. This is a trade-off between line and segment. If we add another parameter for the length, the line segment is fully specified. Special care is given to the uncertainty representation.

### 3.2.1 Representing the orientation by its Euler angles $\phi$ and $\theta$

Let us consider the spherical coordinates (see Fig. 4). Let $\mathbf{u} = [u_x,\ u_y,\ u_z]^T$ be a unit orientation vector, we have:

$$\begin{cases} u_x &= \cos\phi \sin\theta \\ u_y &= \sin\phi \sin\theta \\ u_z &= \cos\theta \end{cases} \tag{2}$$

with $0 \leq \phi < 2\pi,\ \ 0 \leq \theta \leq \pi$.

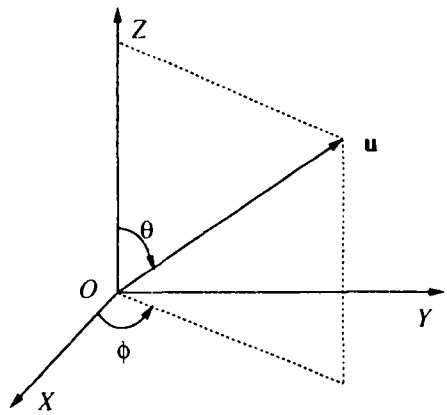



**Fig. 4:** Spherical coordinates

From $\mathbf{u}$, we can compute $\phi$, $\theta$:

$$\begin{aligned} \phi &= \begin{cases} \arccos \frac{u_x}{\sqrt{1-u_z^2}} & \text{if } u_y \geq 0 \\ 2\pi - \arccos \frac{u_x}{\sqrt{1-u_z^2}} & \text{otherwise,} \end{cases} \\ \theta &= \arccos u_z \ . \end{aligned} \tag{3}$$

If we denote $[\phi,\ \theta]^T$ by $\boldsymbol{\psi}$, then the mapping between $\boldsymbol{\psi}$ and $\mathbf{u}$ is 1-to-1 correspondence, except when $\theta = 0$. When $\theta = 0$, $\phi$ is not defined. A special treatment in covariance matrix of $\boldsymbol{\psi}$ is applied in this case, as described below. Another problem with this representation is the discontinuity in $\phi$ when a segment varies nearly parallel to the plane $y = 0$. This discontinuity must be dealt with in matching and data fusion.

In the following, we assume that the direction vector $\mathbf{v} = [x,\ y,\ z]^T$ and its covariance matrix $\Lambda_{\mathbf{v}}$ of a given segment are known, as computed by (1). We want to compute $\boldsymbol{\psi}$ and its covariance matrix $\Lambda_{\boldsymbol{\psi}}$ from $\mathbf{v}$ and $\Lambda_{\mathbf{v}}$. $\phi$ and $\theta$ are simply given by:

$$\begin{aligned} \phi &= \begin{cases} \arccos \frac{x}{\sqrt{x^2+y^2}} & \text{if } y \geq 0 \\ 2\pi - \arccos \frac{x}{\sqrt{x^2+y^2}} & \text{otherwise,} \end{cases} \\ \theta &= \arccos \frac{z}{\sqrt{x^2+y^2+z^2}} \ . \end{aligned} \tag{4}$$

Since the relation between $\psi$ and $\mathbf{v}$ is not linear, we use the first order approximation to compute the covariance matrix $\Lambda_\psi$ from $\Lambda_\mathbf{v}$. That is

$$\Lambda_\psi = \frac{\partial \boldsymbol{\psi}}{\partial \mathbf{v}} \Lambda_\mathbf{v} \frac{\partial \boldsymbol{\psi}}{\partial \mathbf{v}}^T , \tag{5}$$

where the Jacobian matrix

$$\frac{\partial \boldsymbol{\phi}}{\partial \mathbf{v}} = \begin{bmatrix} \frac{\partial \phi}{\partial x} & \frac{\partial \phi}{\partial y} & \frac{\partial \phi}{\partial z} \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial y} & \frac{\partial \theta}{\partial z} \end{bmatrix} . \tag{6}$$

Note that

$$\frac{\partial \arccos x}{\partial x} = -\frac{1}{\sqrt{1-x^2}} .$$

After some simple computation, we have

$$\begin{aligned}
\frac{\partial \phi}{\partial x} &= -\frac{y}{x^2+y^2} , && \text{for all } y \\
\frac{\partial \phi}{\partial y} &= \frac{x}{x^2+y^2} , && \text{for all } y \\
\frac{\partial \phi}{\partial z} &= 0 , && \text{for all } y \\
\frac{\partial \theta}{\partial x} &= \frac{xz}{(x^2+y^2+z^2)\sqrt{x^2+y^2}} , \\
\frac{\partial \theta}{\partial y} &= \frac{yz}{(x^2+y^2+z^2)\sqrt{x^2+y^2}} , \\
\frac{\partial \theta}{\partial z} &= -\frac{\sqrt{x^2+y^2}}{x^2+y^2+z^2} .
\end{aligned}$$

Notice that when $x^2 + y^2 = 0$, i.e., $x = 0$ and $y = 0$,

$$\frac{\partial \theta}{\partial x} = \frac{\partial \theta}{\partial y} = \frac{1}{z} \text{ and } \frac{\partial \theta}{\partial z} = 0 ,$$

but $\frac{\partial \phi}{\partial x}$ and $\frac{\partial \phi}{\partial y}$ are not differentiable. It is reasonable since $\phi$ is not defined when the vector is parallel to the axis $z$, and a slight change in $x$ or $y$ may provoke a drastic change in $\phi$. To deal with this problem, we replace $\frac{\partial \phi}{\partial x}$ and $\frac{\partial \phi}{\partial y}$ by a very big number when $x^2 + y^2 = 0$, so that the components of $\phi$ in $\Lambda_\psi$ are very big. That is to say that the measurement of $\phi$ has no information content.

### 3.2.2 Modeling the midpoint of a 3D line segment

We choose the midpoint as the three parameters to localize the segment, but a special treatment on the covariance is introduced to characterize its uncertainty.

The midpoint $M$ and its covariance matrix $\Lambda_M$ can be computed from the endpoints of the segment by $M = (M_1 + M_2)/2$ and $\Lambda_M = (\Lambda_1 + \Lambda_2)/4$. However, the way the uncertainty of the endpoints of a three-dimensional segment is computed takes only into account the uncertainty of the pixel coordinates due to the edge detection process and the uncertainty of the calibration of the stereo rig. It does not take into account the uncertainty due to the variations in the different images of the stereo triplet of the polygonal approximations of corresponding contours. In an attempt to cope with all this, we model the midpoint $\mathbf{m}$ of a segment $M_1M_2$ as:

$$\mathbf{m} = (M_1 + M_2)/2 + n\mathbf{u} \ , \tag{7}$$

where $\mathbf{u}$ is the unit direction vector of the segment and $n$ is a random scalar. Equation 7 says in fact that the midpoint has some extra uncertainty attached to it. It may vary randomly along the line supporting it in successive views. Remark that this modelization is in accordance with the definition of a line. If a point $\mathbf{p}_0$ on a line and its orientation $\mathbf{u}$ are given, the line $L$ may be defined as a set of points in 3D space parametrized by a real variable $t$:

$$L = \{\mathbf{p} \mid \mathbf{p} = \mathbf{p}_0 + t\mathbf{u}, \ -\infty < t < \infty\} \ . \tag{8}$$

The random variable $n$ in (7) is modeled as Gaussian with mean zero and standard deviation $\sigma_n$, a positive scalar. If a segment is reliable, $\sigma_n$ may be chosen to be a small number; if not, it may be chosen to be a big one. In our implementation, $\sigma_n$ is related to the length $l$ of the segment, i.e., $\sigma_n = \kappa l$, where $\kappa$ is some constant. That is to say that the longer a segment is, the bigger the deviation $\sigma_n$ is. This is reasonable since a long segment is much likely to be broken into smaller segments in other views. In our experiments, we found that $\kappa = 0.2$ gives us very good results.

In order to compute the covariance of $\mathbf{m}$, we should first compute the unit direction vector $\mathbf{u}$ and its covariance $\Lambda_\mathbf{u}$. They can be computed from the representation $\boldsymbol{\psi}$ and $\Lambda_\psi$. Indeed, we can compute $\mathbf{u}$ from $\boldsymbol{\psi}$ based on (2), and $\Lambda_\mathbf{u}$ is given by

$$\Lambda_\mathbf{u} = \frac{\partial \mathbf{u}}{\partial \boldsymbol{\psi}} \Lambda_\psi \frac{\partial \mathbf{u}}{\partial \boldsymbol{\psi}}^T \ , \tag{9}$$

where $\frac{\partial \mathbf{u}}{\partial \boldsymbol{\psi}}$ is a $3 \times 2$ matrix:

$$\frac{\partial \mathbf{u}}{\partial \boldsymbol{\psi}} = \begin{bmatrix} -\sin\phi\sin\theta & \cos\phi\cos\theta \\ \cos\phi\sin\theta & \sin\phi\cos\theta \\ 0 & -\sin\theta \end{bmatrix} . \tag{10}$$

Note that the covariance matrix $\Lambda_{\mathbf{u}}$ is singular (the determinant is zero). This is reasonable since the three components of $\mathbf{u}$ are not independent ($\|\mathbf{u}\| = 1$).

At this point, the covariance of $\mathbf{m}$ can be computed. We start with the covariance of $n\mathbf{u}$. Since $n$ and $\mathbf{u}$ are independent to each other, we have

$$E[n\mathbf{u}] = E[n]E[\mathbf{u}] = 0 \ , \tag{11}$$

$$\Lambda_{n\mathbf{u}} = E[(n\mathbf{u})(n\mathbf{u})^T] = E[n^2\mathbf{u}\mathbf{u}^T] = E[n^2]E[\mathbf{u}\mathbf{u}^T] = \sigma_n^2(\Lambda_{\mathbf{u}} + \bar{\mathbf{u}}\bar{\mathbf{u}}^T) \ , \tag{12}$$

where $\bar{\mathbf{u}} = E[\mathbf{u}]$. Now we have

$$E[\mathbf{m}] = E[M] \ , \tag{13}$$

and

$$\begin{aligned} \Lambda_{\mathbf{m}} &= E[(\mathbf{m} - E[\mathbf{m}])(\mathbf{m} - E[\mathbf{m}])^T] \\ &= E[(\mathbf{M} - E[\mathbf{M}])(\mathbf{M} - E[\mathbf{M}])^T] + E[(n\mathbf{u})(n\mathbf{u})^T] \\ &\quad + E[n(\mathbf{M} - E[\mathbf{M}])\mathbf{u}^T] \ . \end{aligned} \tag{14}$$

Since $n$ is independent of $\mathbf{M}$ and $\mathbf{u}$ and has zero-mean, the last term is equal to 0 and

$$\Lambda_{\mathbf{m}} = \Lambda_M + \Lambda_{n\mathbf{u}} \ .$$

If we add another parameter $l$ to denote the length of the segment, we can then represent exactly a line segment. The variance on the length does not need to be modeled, since this information is not required in the motion and fusion algorithms. This ends our modelization of a line segment.

More rigorously, there exists a correlation between $\boldsymbol{\psi}$ and $\mathbf{m}$, but this correlation is negligible. We have computed the correlation for many segments, and we found that the coefficient of correlation between $\boldsymbol{\psi}$ and $\mathbf{m}$ is less than 0.01.

# 4 Fusing Segments from Multiple Views

The problem addressed in this section is known as the multiple viewpoint problem. The objective is to build a consistent, accurate and complete description (model) of objects or environments by combining the observations taken by the stereo system from multiple stereo views. One stereo view can only provide partial imprecise information about the environment, which is not sufficient for interpretation, recognition or navigation.

Assume the robot navigates in a static environment. Suppose that at time $t_i$ we have built up a model $\mathcal{F}_{i-1}$ from previous views, we now want to integrate it with the new view $\mathcal{S}_i$. Figure 5 shows a more detailed description of the relation between the motion-determination and data-integration modules. The stereo frame at the current instant $\mathcal{S}_i$ and that of the

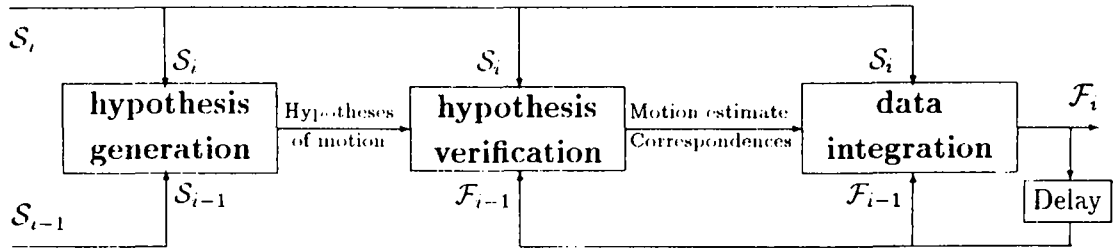



**Fig. 5:** Relation between motion estimation and data integration

preceding instant $\mathcal{S}_{i-1}$ are used at the first stage to generate hypotheses of motion between successive instants. We do not use the fused data $\mathcal{F}_{i-1}$ since the complexity of the generation algorithm in the worst case is $O(n^2m^2)$ ($n$ and $m$ are the number of primitives in $\mathcal{F}_{i-1}$ and $\mathcal{S}_i$, respectively) and $\mathcal{F}_{i-1}$ has more primitives than $\mathcal{S}_i$. $\mathcal{S}_i$ and $\mathcal{F}_{i-1}$ (at time $t_2$, $\mathcal{F}_1$ is the same as $\mathcal{S}_1$) are used at the verification stage which provides as its output an optimal estimate of the egomotion $\mathbf{d}_i$ and also segment correspondences between $\mathcal{S}_i$ and $\mathcal{F}_{i-1}$. The integration module then builds a more accurate and complete model $\mathcal{F}_i$ by combining all available information. We choose the coordinate system of the last observed frame $\mathcal{S}_i$ as that of the global model $\mathcal{F}_i$ being updated.

Fused segments have a label which indicates their number of instances up to the current instant. For example, if a segment in $\mathcal{F}_{i-1}$ with a label $k$ is matched to a segment in $\mathcal{S}_i$, then the fused segment in $\mathcal{F}_i$ has a label $k+1$. Unmatched segments in $\mathcal{S}_i$ are simply copied into $\mathcal{F}_i$ and are labeled with 1. These segments are always retained since they are likely to appear at the next instant. Unmatched segments in $\mathcal{F}_{i-1}$ with a label 1 may or may not be retained depending upon whether the number of primitives in $\mathcal{F}_i$ becomes too big or not. Those segments are less likely to appear at the next instant than those unmatched segments in $\mathcal{S}_i$. In fact, a segment which appeared only once during several preceding instants is very likely not to correspond to any real segment in space. It may have been reconstructed from false matches in stereo or its image contours have been poorly segmented. Other unmatched segments in $\mathcal{F}_{i-1}$ are retained in $\mathcal{F}_i$ by applying the motion estimate $\mathbf{d}_i$ and its covariance matrix $\Lambda_{\mathbf{d}_i}$.

There remains the problem of fusing the primitives in correspondence into a new primitive while taking into account $\mathbf{d}_i$ and its covariance matrix $\Lambda_{\mathbf{d}_i}$ (see Fig. 6).

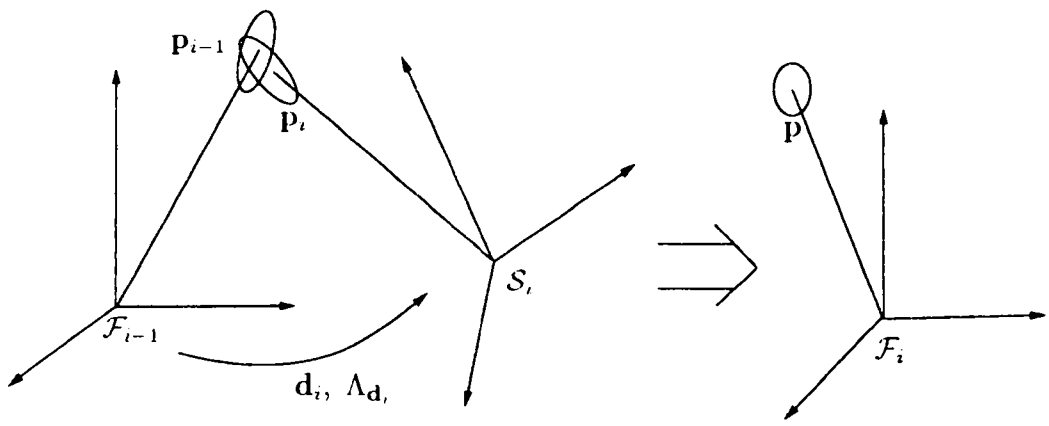



**Fig. 6:** Fusing data from two different views

## 4.1 Fusing General Primitives

We use a procedure based on the Kalman filter theory to integrate multiple observations. Suppose we have two independent observations $\mathbf{x}_1$ and $\mathbf{x}_2$ with their covariance matrices $\Lambda_1$ and $\Lambda_2$ of a state vector $\mathbf{x}$. The modified Kalman minimum-variance estimator says that the optimal estimate is given by[‡]

$$\begin{aligned} \hat{\mathbf{x}} &= [\Lambda_1^{-1} + \Lambda_2^{-1}]^{-1}[\Lambda_1^{-1}\mathbf{x}_1 + \Lambda_2^{-1}\mathbf{x}_2] \ , \\ \Lambda &= [\Lambda_1^{-1} + \Lambda_2^{-1}]^{-1} \ . \end{aligned} \tag{15}$$

That is, the optimal estimate is just a weighted average of the observations and the corresponding information matrix (i.e., the inverse of the covariance matrix) is the sum of the information matrices of the observations. The covariance matrix is always reduced in integrating an observation. Equation (15) can be easily proved using the standard Kalman filter

[‡]These formulae can be extended for $n$ independent observations $\mathbf{x}_i$ $(i = 1 \ldots n)$:

$$\begin{aligned} \hat{\mathbf{x}} &= (\Sigma_{i=1}^{n}\Lambda_i^{-1})^{-1}(\Sigma_{i=1}^{n}\Lambda_i^{-1}\mathbf{x}_i) \ , \\ \Lambda &= (\Sigma_{i=1}^{n}\Lambda_i^{-1})^{-1} \ . \end{aligned}$$

When $n = 2$, the following formulae are computationally less expensive

$$\begin{aligned} \hat{\mathbf{x}} &= \Lambda_2(\Lambda_1 + \Lambda_2)^{-1}\mathbf{x}_1 + \Lambda_1(\Lambda_1 + \Lambda_2)^{-1}\mathbf{x}_2 \ , \\ \Lambda &= \Lambda_1(\Lambda_1 + \Lambda_2)^{-1}\Lambda_2 \ . \end{aligned}$$

by setting the transformation matrix and the measurement matrix to identity matrices. Note that we can apply iteratively the above procedure if we have more than two observations. First we use (15) to compute a new estimate $\hat{\mathbf{x}}$ and its associated covariance matrix $\Lambda$, then for every other observation we compute an up-to-date estimate by integrating the old one with the observation based on (15).

One requirement to use the above procedure is that all observations are expressed in the same coordinate system. Consider Fig. 6. Let $\mathbf{p}_{i-1}$ and $\mathbf{p}_i$ be primitives in correspondence in $\mathcal{F}_{i-1}$ and $\mathcal{S}_i$ and $\mathbf{p}$ the new primitive to be included in $\mathcal{F}_i$ (we choose the coordinate system of $\mathcal{S}_i$ as that of $\mathcal{F}_i$). We see that $\mathbf{p}_{i-1}$ and $\mathbf{p}_i$ are represented in two different coordinate systems which are related by the motion vector $\mathbf{d}_i$ and its covariance matrix $\Lambda_{\mathbf{d}_i}$. In order to use the above procedure, we should first transform $\mathbf{p}_{i-1}$ in the coordinate system of $\mathcal{F}_i$, denoted by $\mathbf{p}'_{i-1}$, based on $\mathbf{d}_i$ and $\Lambda_{\mathbf{d}_i}$. Let ${}_{i-1}T_i = f(\mathbf{d}_i)$ be the transformation matrix from $\mathcal{F}_{i-1}$ to $\mathcal{F}_i$, the transformed primitive of $\mathbf{p}_{i-1}$ in Frame $\mathcal{F}_i$ is given by

$$\mathbf{p}'_{i-1} = {}_{i-1}T_i \mathbf{p}_{i-1} \ , \tag{16}$$

and its covariance matrix is given by

$$\Lambda_{\mathbf{p}'_{i-1}} = {}_{i-1}T_i \, \Lambda_{\mathbf{p}_{i-1}} \, {}_{i-1}T_i^T + \frac{\partial({}_{i-1}T_i \mathbf{p}_{i-1})}{\partial \mathbf{d}_i} \Lambda_{\mathbf{d}_i} \frac{\partial({}_{i-1}T_i \mathbf{p}_{i-1})}{\partial \mathbf{d}_i}^T \ . \tag{17}$$

See (Zhang, 1990) for the details of the computation of the derivative $\frac{\partial({}_{i-1}T_i \mathbf{p}_{i-1})}{\partial \mathbf{d}_i}$.

Another requirement is that the observations are independent. Strictly speaking, $\mathbf{p}'_{i-1}$ and $\mathbf{p}_i$ are not independent, although $\mathbf{p}_{i-1}$ and $\mathbf{p}_i$ can be reasonably assumed independent. This is because $\mathbf{p}'_{i-1}$ is computed from the original primitive $\mathbf{p}_{i-1}$ by applying the motion estimate $\mathbf{d}_i$ and both $\mathbf{p}_{i-1}$ and $\mathbf{p}_i$ contribute to the estimation of $\mathbf{d}_i$. This results the correlation between $\mathbf{p}'_{i-1}$ and $\mathbf{p}_i$. However this correlation is negligible since the motion vector $\mathbf{d}_i$ is usually estimated from more than 30 correspondences (150 correspondences, sometimes).

## 4.2 Fusing Line Segments

A 3D line segment is represented by ($\psi$, $\mathbf{m}$, $l$) and ($\Lambda_\psi$, $\Lambda_\mathbf{m}$) (see Sect. 3). No modelization of the uncertainty on the segment lengths has been carried out (we do not need it). Let the *transformed* segment from $\mathcal{F}_{i-1}$ be represented by ($\psi_1$, $\mathbf{m}_1$) with their covariance matrices ($\Lambda_{\psi_1}$, $\Lambda_{\mathbf{m}_1}$) and the segment from $\mathcal{S}_i$ be represented by ($\psi_2$, $\mathbf{m}_2$) with their covariance

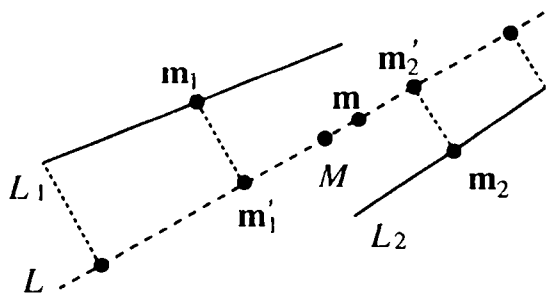



**Fig. 7:** Union of two matched segments

matrices ($\Lambda_{\psi_2}$, $\Lambda_{\mathbf{m}_2}$). Then the parameters of the fused segment are given by

$$\begin{aligned} \boldsymbol{\psi} &= [\Lambda_{\psi_1}^{-1} + \Lambda_{\psi_2}^{-1}]^{-1}(\Lambda_{\psi_1}^{-1}\boldsymbol{\psi}_1 + \Lambda_{\psi_2}^{-1}\boldsymbol{\psi}_2) , \\ \Lambda_{\psi} &= [\Lambda_{\psi_1}^{-1} + \Lambda_{\psi_2}^{-1}]^{-1} , \\ \mathbf{m} &= [\Lambda_{\mathbf{m}_1}^{-1} + \Lambda_{\mathbf{m}_2}^{-1}]^{-1}(\Lambda_{\mathbf{m}_1}^{-1}\mathbf{m}_1 + \Lambda_{\mathbf{m}_2}^{-1}\mathbf{m}_2) , \\ \Lambda_{\mathbf{m}} &= [\Lambda_{\mathbf{m}_1}^{-1} + \Lambda_{\mathbf{m}_2}^{-1}]^{-1} . \end{aligned} \tag{18}$$

They give the orientation and position of the fused segment.

In the case of the discontinuity of $\phi$ (see Sect. 3.2), care must be taken before we use the above procedure. The idea is the following. If a segment is represented by $\boldsymbol{\psi} = [\phi, \theta]^T$, it is also represented by $[\phi - 2\pi, \theta]^T$. Therefore, when fusing two segments, we perform the following tests and actions. If $\phi_1 < \pi/2$ and $\phi_2 > 3\pi/2$, then set $\phi_2 = \phi_2 - 2\pi$; else if $\phi_1 > 3\pi/2$ and $\phi_2 < \pi/2$, then set $\phi_1 = \phi_1 - 2\pi$; else do nothing. Notice that adding a constant to a random variable does not affect its covariance matrix. Using $\phi_1$ and $\phi_2$, we can compute a $\phi'$ using the above procedure. The $\phi$ of the fused segment is then equal to $\phi'$ if $\phi' \geq 0$, and equal to $2\pi + \phi'$ if $\phi' < 0$.

Due to the reasons described in Sect. 3, we can conclude that a reconstructed segment is only part of a real segment in space. Two segments are considered as being matched if they have a common part. Their corresponding segment in space can be expected not to be shorter than either of the two segments. That is, the union of the two segments can be reasonably considered as a better estimate of the corresponding segment in space.

Consider two segments to be fused, $L_1$ and $L_2$ in Fig. 7. First an infinite line $L$ is computed using (18). Then the midpoints of the two segments are projected on $L$, and we get $\mathbf{m}'_1$ and $\mathbf{m}'_2$. The lengths of the projected segments are also computed, and are denoted by $l'_1$ and $l'_2$. Now we want to compute the real midpoint $M$ and the length $l$ of the fused segment $L$. After studying all cases, we have the following algorithm:

**Algorithm 1: Union of Two Segments**

- **Input:** $(\psi_1, \mathrm{m}_1, l_1)$ and $(\psi_2, \mathrm{m}_2, l_2)$
- **Output:** $(\psi, \mathrm{m}, l)$
- **Define** local variables: $\mathrm{m}'_1$, $\mathrm{m}'_2$, $l'_1$, $l'_2$ and $M$

- Compute, using (18), the infinite line $L$: $(\psi, \mathrm{m})$
- Project $\mathrm{m}_1$ and $\mathrm{m}_2$ on $L$: $\mathrm{m}'_1$ and $\mathrm{m}'_2$
- Project $l_1$ and $l_2$ on $L$: $l'_1$ and $l'_2$
- Compute the real midpoint $M$ and length $l$ of $L$:

  →**if** $(\|\overrightarrow{\mathrm{m}'_1\mathrm{m}'_2}\| + l'_2/2 < l'_1/2)$

  **then** $\hookrightarrow M = \mathrm{m}'_1, \quad l = l'_1$

  **else if** $(\|\overrightarrow{\mathrm{m}'_1\mathrm{m}'_2}\| + l'_1/2 < l'_2/2)$

  **then** $\hookrightarrow M = \mathrm{m}'_2$

  $\hookrightarrow l = l'_2$

  **else** $\hookrightarrow M = \mathrm{m}'_1 + \dfrac{\|\overrightarrow{\mathrm{m}'_1\mathrm{m}'_2}\| + l'_2/2 - l'_1/2}{2} \dfrac{\overrightarrow{\mathrm{m}'_1\mathrm{m}'_2}}{\|\overrightarrow{\mathrm{m}'_1\mathrm{m}'_2}\|}$

  $\hookrightarrow l = l'_1 + (\|\overrightarrow{\mathrm{m}'_1\mathrm{m}'_2}\| + l'_2/2 - l'_1/2)$

  ←**endif**
- Set $\mathrm{m} = M$

where $\overrightarrow{\mathrm{m}'_1\mathrm{m}'_2} = \mathrm{m}'_2 - \mathrm{m}'_1$.

The covariance matrix of $M$ can be extrapolated from that of $\mathbf{m}$, $\Lambda_{\mathrm{m}}$, and that of the unit direction vector $\mathbf{u}$, $\Lambda_{\mathbf{u}}$. Indeed, $M$ can always be represented by

$$M = \mathbf{m} + s\mathbf{u} \ , \tag{19}$$

where $s$ is an arbitrary scalar. The above equation can be interpreted as the addition of a *biased* noise to $\mathbf{m}$, therefore the covariance matrix of $M$ is given by

$$\Lambda_M = \Lambda_{\mathbf{m}} + s^2(\Lambda_{\mathbf{u}} + \mathbf{u}\mathbf{u}^T) \ . \tag{20}$$

## 4.3 Example

In this section, we give an example of fusing two segments.

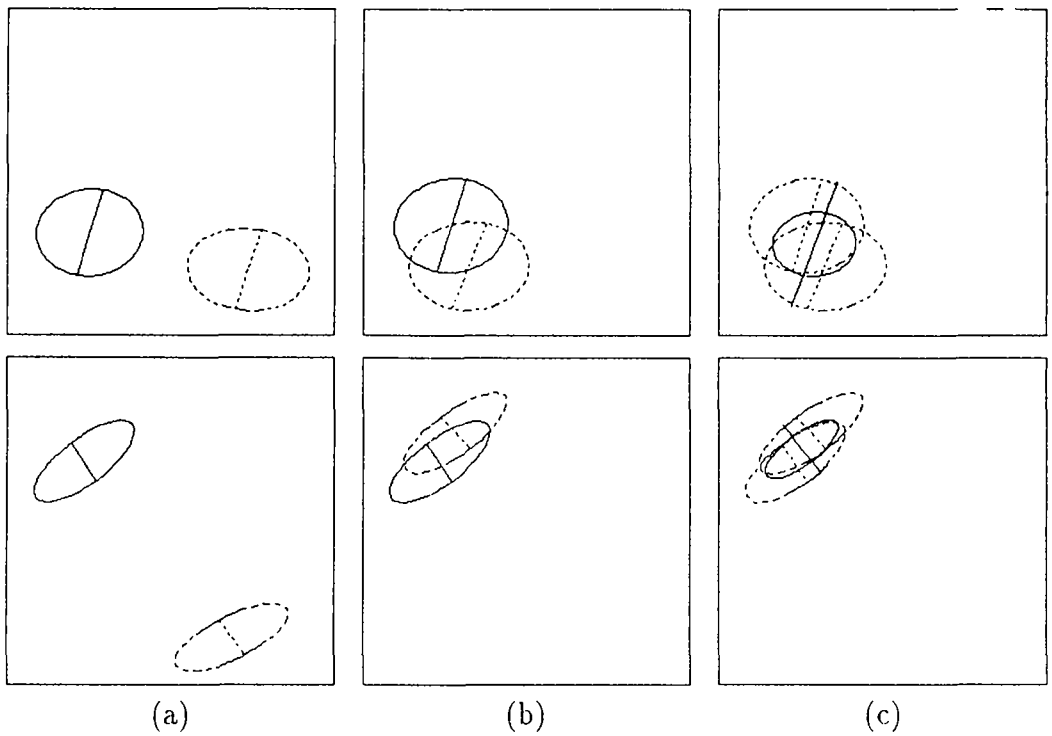



**Fig. 8:** Fusing two segments: evolution of the uncertainty in the midpoint

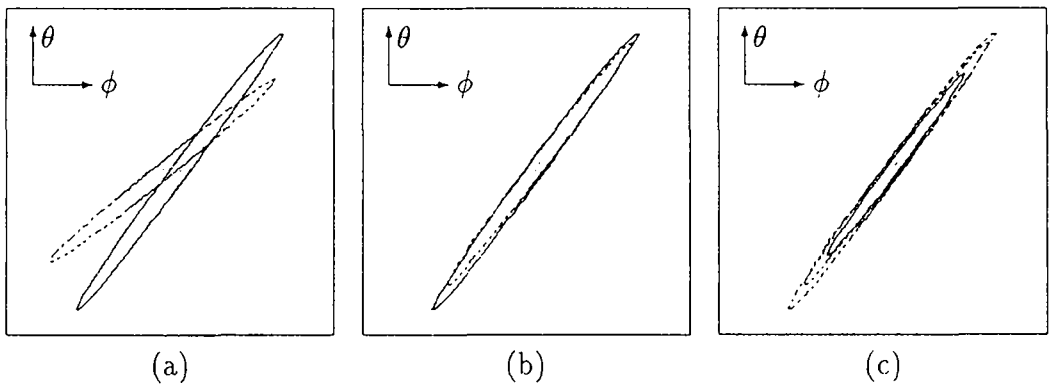



**Fig. 9:** Fusing two segments: evolution of the uncertainty in the orientation

Figure 8 shows two segments which are taken from two real stereo frames. Pictures in the first row are the front view and those in the second row are the top view. Figure 8a shows the superposition of the two original segments and Figure 8b shows their superposition after applying the estimated motion to the first segment (in dashed line). The solid segment in Fig. 8c is the fused one which is, for reason of comparison, superimposed with its original segments (in dashed line). The uncertainties of the midpoints are represented by ellipses. Figure 9 shows the evolution of uncertainty in the orientation parameters $\psi$. Figure 9a shows the superposition of the two original $\psi$'s (represented by points) and their uncertainty ellipses, and Figure 9b shows their superposition after applying the estimated motion to the first $\psi$ (in dashed line). Figure 9c displays the $\psi$ and its uncertainty ellipse of the fused segment (in solid line) with its original observations (in dashed line). We can observe how the uncertainty is reduced.

### 4.4 Summary of the Fusion Algorithm

A fused 3D line segment has a label called the *instance* which indicates the number of instances, i.e., the number of original segments which have been fused, until the current instant. Algorithm 2 gives a more formally description on how the fusion process works. The fused frame $\mathcal{F}_1$ at $t_1$ is set to be the observed stereo frame $\mathcal{S}_1$.

## 5 Experimental Results

In this section, we provide three experimental results to measure both quantitatively and qualitatively the performances of the integration process. The first one gives a quantitative analysis. The second shows the result of fusing two stereo views, and the third shows that of fusing 35 stereo views of a room.

### 5.1 Quantitative Analysis

In this experiment we used three data sets, each containing ten noisy 3D frames. Each 3D frame consists of four 3D line segments, which was generated as follows. The noise-free segments form a square with side equal to 500 millimeters and are 3 meters before the stereo setup. We have projected them on each camera using the real calibration parameters obtained for our trinocular stereo system, and therefore obtained four triplets of 2D segments. The projection of the sqaure on a camera is almost a rectangle of width about 120 pixels

**Algorithm 2: Integration of Multiple Stereo Views**

- **Input:** Stereo frames $\mathcal{S}_{i-1}$ and $\mathcal{S}_i$ and previous fused frame $\mathcal{F}_{i-1}$
- **Output:** Updated fused frame $\mathcal{F}_i$

- Generate motion hypotheses between $\mathcal{S}_{i-1}$ and $\mathcal{S}_i$
- → **for** each hypothesis generated, using $\mathcal{F}_{i-1}$ and $\mathcal{S}_i$
  - ↪ Compute the optimal motion estimate $\mathcal{D}$
  - ↪ Find line segment matches $\{\mathcal{M}\}$

  ← **endfor**
- Choose the best hypothesis which gives $\mathcal{D}_{\text{best}}$ and $\{\mathcal{M}_{\text{best}}\}$
- Get $\mathcal{F}'_{i-1}$ by applying $\mathcal{D}_{\text{best}}$ to $\mathcal{F}_{i-1}$

  `/* the last frame has been chosen as the global reference */`
- Consider segments which have been matched

  → **for** each pairing $(S', S)$ in $\{\mathcal{M}_{\text{best}}\}$
  - ↪ Fuse them as $S_{\text{F}}$ `/* Algorithm 1 */`
  - ↪ Retain $S_{\text{F}}$ in $\mathcal{F}_i$ and set its *instance* to that of $S'$ plus 1

  ← **endfor**
- Copy unmatched segments from $\mathcal{F}'_{i-1}$ to $\mathcal{F}_i$
- Copy unmatched segments in $\mathcal{S}_i$ into $\mathcal{F}_i$
  and initialize their *instances* all to 1

**Table 1:** Quantitative results of fusion

| *Data set* | | No. 1 | | No. 2 | | No. 3 | |
|---|---|---|---|---|---|---|---|
| *Segment* | | angle | distance | angle | distance | angle | distance |
| No. 1 | average | 9.67 | 39.69 | 7.73 | 21.67 | 9.99 | 31.18 |
| | fusion | 5.23 | 9.15 | 0.16 | 7.03 | 2.12 | 10.09 |
| No. 2 | average | 7.60 | 47.81 | 8.82 | 28.51 | 9.60 | 32.90 |
| | fusion | 2.97 | 6.34 | 5.98 | 3.42 | 5.52 | 16.15 |
| No. 3 | average | 19.68 | 37.55 | 6.54 | 33.05 | 8.45 | 33.66 |
| | fusion | 8.84 | 4.31 | 2.63 | 13.28 | 2.25 | 2.88 |
| No. 4 | average | 13.85 | 46.30 | 15.42 | 55.83 | 11.72 | 31.37 |
| | fusion | 1.46 | 16.24 | 5.43 | 2.02 | 3.40 | 7.60 |

and of height about 150 pixels (image resolution: 512×512). We then added independent Gaussian noise to the endpoints of the projected segments. The noise for each projected segment has two independent components: one component parallel to the segment and another perpendicular to the segment. The parallel noise term was added such that the length of the segment was shortened. The parallel component is a random scalar with mean zero and standard deviation $\sigma_{\parallel} = 20$ pixels; the perpendicular one is a random scalar with mean zero and standard deviation $\sigma_{\perp}$ ranging from 0.5 pixels to 2.5 pixels. Such noises were added to both endpoints of a segment. This partly models the precision of our edge-detection and polygonal approximation processes: more uncertainty is present along the segment than in its normal direction. At this point, we have generated a noisy image triplet. It was then supplied to the trinocular stereo system, and a realistic 3D frame was finally reconstructed.

In the first set of ten frames, the $\sigma_{\perp}$'s are equal to 0.5, 1.0, 1.4, 1.7, 1.9, 2.1, 2.2, 2.3, 2.4 and 2.5, respectively. In the second set, the $\sigma_{\perp}$'s of the first five frames are all equal to 1 pixel, and those of the last five frames, 2 pixels. In the third set, the $\sigma_{\perp}$'s of the first five frames are all equal to 2 pixels, and those of the last five frames, 1 pixel. We then ran the integration process to fuse the ten frames in each data set. To measure the quality of the fusion result, we define two criteria. The first is the angle between the fused segment and its corresponding noise-free segment. The second is the distance of the midpoint of the noise-free segment to the fused one. The result are given in Table 1, where for reason of comparison we also show the average of the angles between each noise-free segment and its corresponding noisy ones, and the average of the distances of the midpoint of each noise-free segment to its corresponding noisy ones. The angles are in degrees and the distances are in millimeters. As can be observed, considerable improvement in the accuracy of the

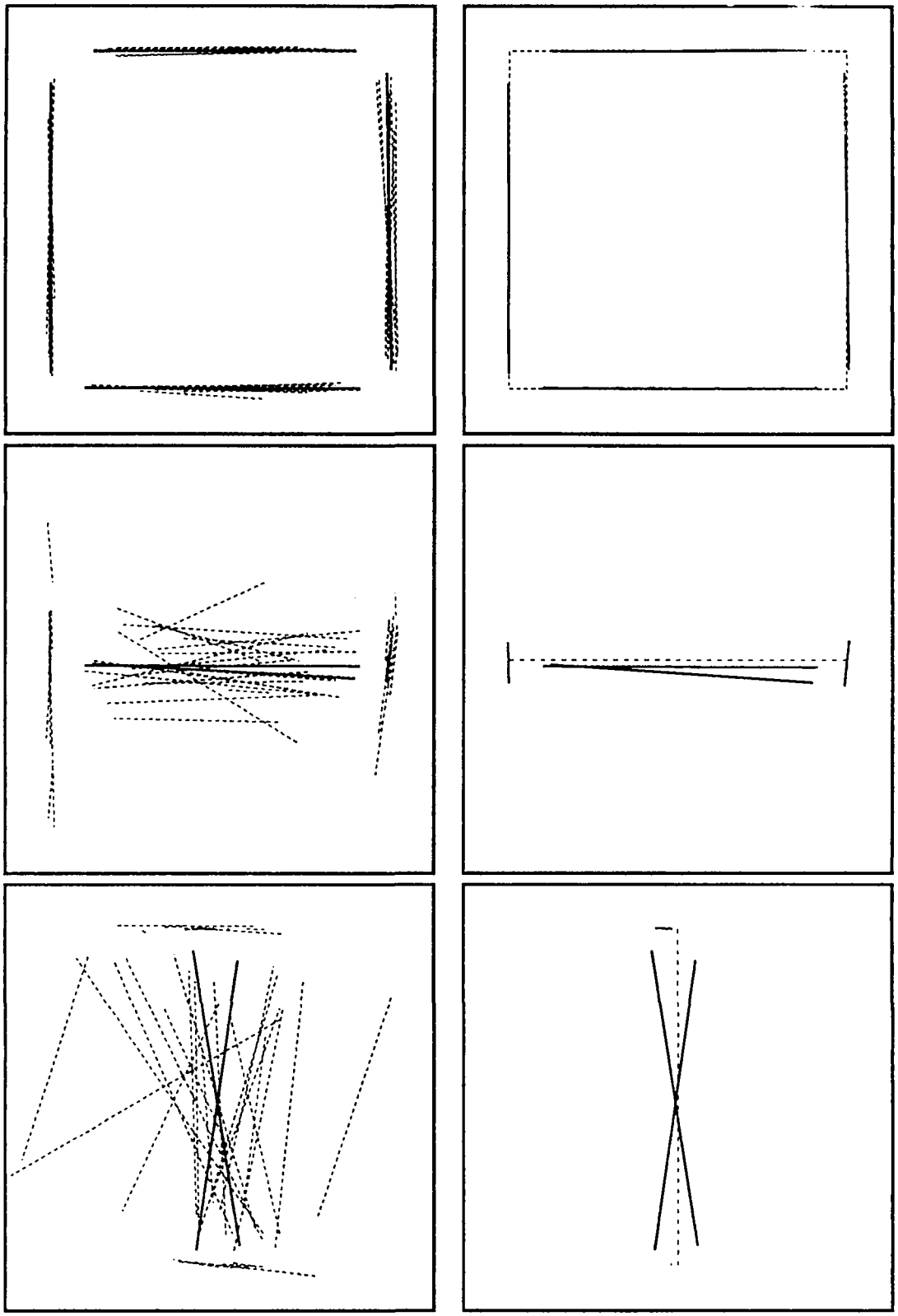


Fig. 10: Fusion results with synthetic data

measurements has been achieved.

Figure 10 gives a visual presentation of the fusion results with the second data set. In the pictures on the left, we show the superposition of the fused segments (in solid lines) and their original noisy segments (in dashed lines). In the pictures on the right, we show the superposition of the fused segments (in solid lines) and their original noise-free segments (in dashed lines). The first picture in each column shows the projection to a plane almost parallel to the plane passing through the cameras optical centers (front view). The second shows the projection to the ground plane (top view) and the third shows the projection to a plane perpendicular to the above two planes (side view). We observe that we can obtain precise measurements from very noisy data through fusion.

## 5.2 Fusion of Two 3D Views

Figures 11 and 12 show two stereo frames reconstructed by our mobile robot in two different positions. The triangle in each picture represents the optical centers of the cameras of our trinocular stereo system. We have 261 segments in the first frame and 250 segments in the second. Note that there is a large displacement between these two positions (about 10 degrees of rotation and 75 centimeters of translation) which can be noticed by superposing the two frames (see Fig. 13). The maximum shift in the 2D images is about 95 pixels (the image resolution is 512×512) (we show in Fig. 14 the images taken by the first camera).

Applying our hypothesis generation procedure to these two frames, we obtain 12 hypotheses. All these hypotheses are propagated to the whole frame to match more segments and to update the motion estimate. In the end, 9 hypotheses correctly give the estimate of the displacement. The one which matches the largest number of segments and gives the minimal matching error is kept as the best one. To determine how good this estimate is, we apply it to the first frame and superimpose the transformed one on the second, i.e., in the coordinate system of the second frame (see Fig. 15). The shift of the triangle in this figure displays in fact the displacement of the robot. Figure 16a shows the superposition of the matched segments after applying the estimated motion to the first frame. One observes a very good accuracy of the motion estimate. We obtain, at the same time, 170 matches between the two frames. Among the recovered matches, 14 are multiple matches. Our algorithm admits multiple matches, that is, a segment in the first frame can have two or more correspondences in the second frame, and *vice versa*. Two broken segments and a long segment can be matched by our algorithm. Segments of multiple matches are fused into a single segment, so we get 156 fused segments in total. The fusion result is displayed in Fig. 16b. Comparing Fig. 16b with Figs. 11 and 12 (or Fig. 16a), we observe the improvement in the accuracy of segment measurements (examining, for example, segments on the right).

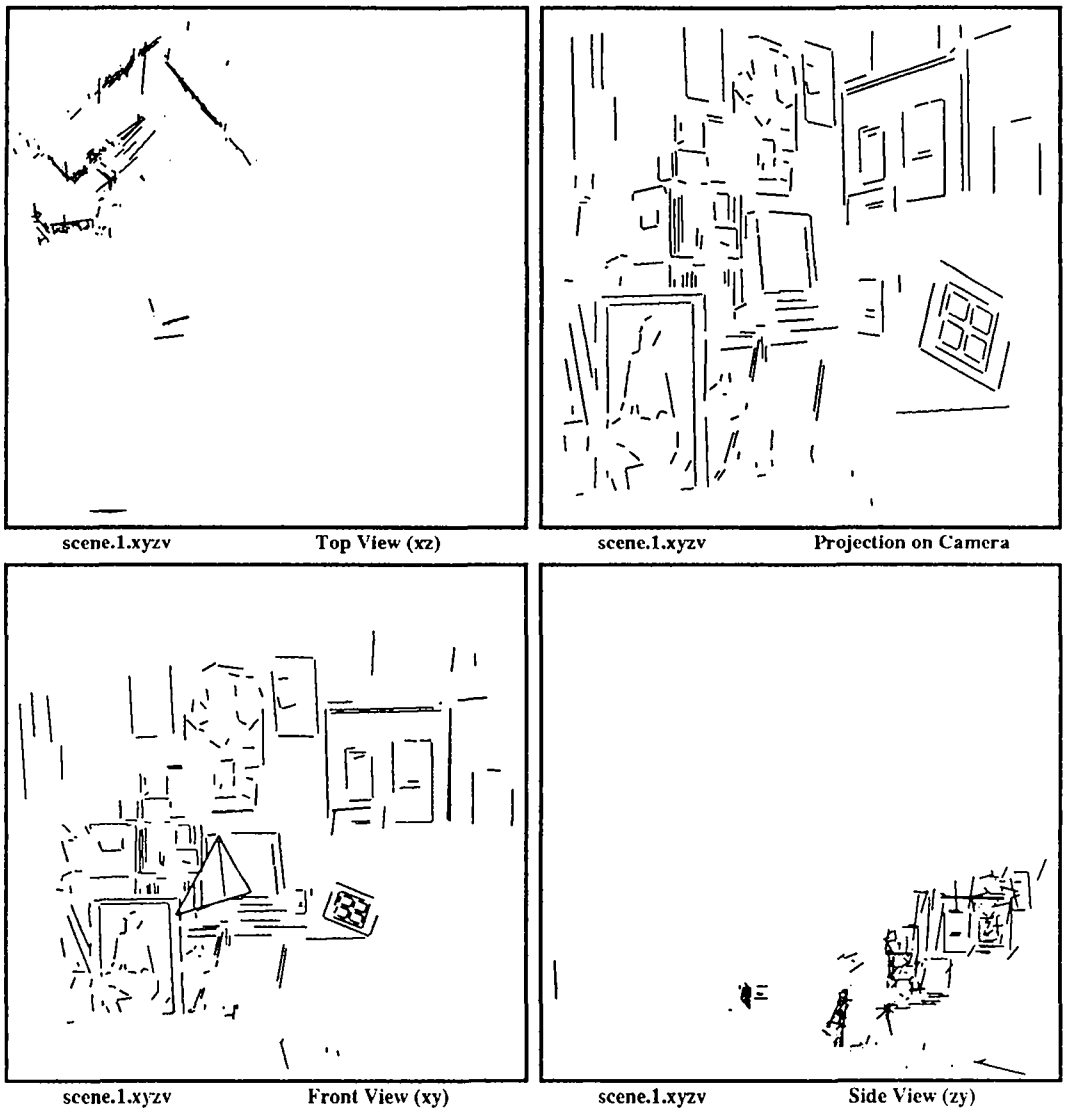



Fig. 11: Different views of stereo frame 1 (uniform scale)

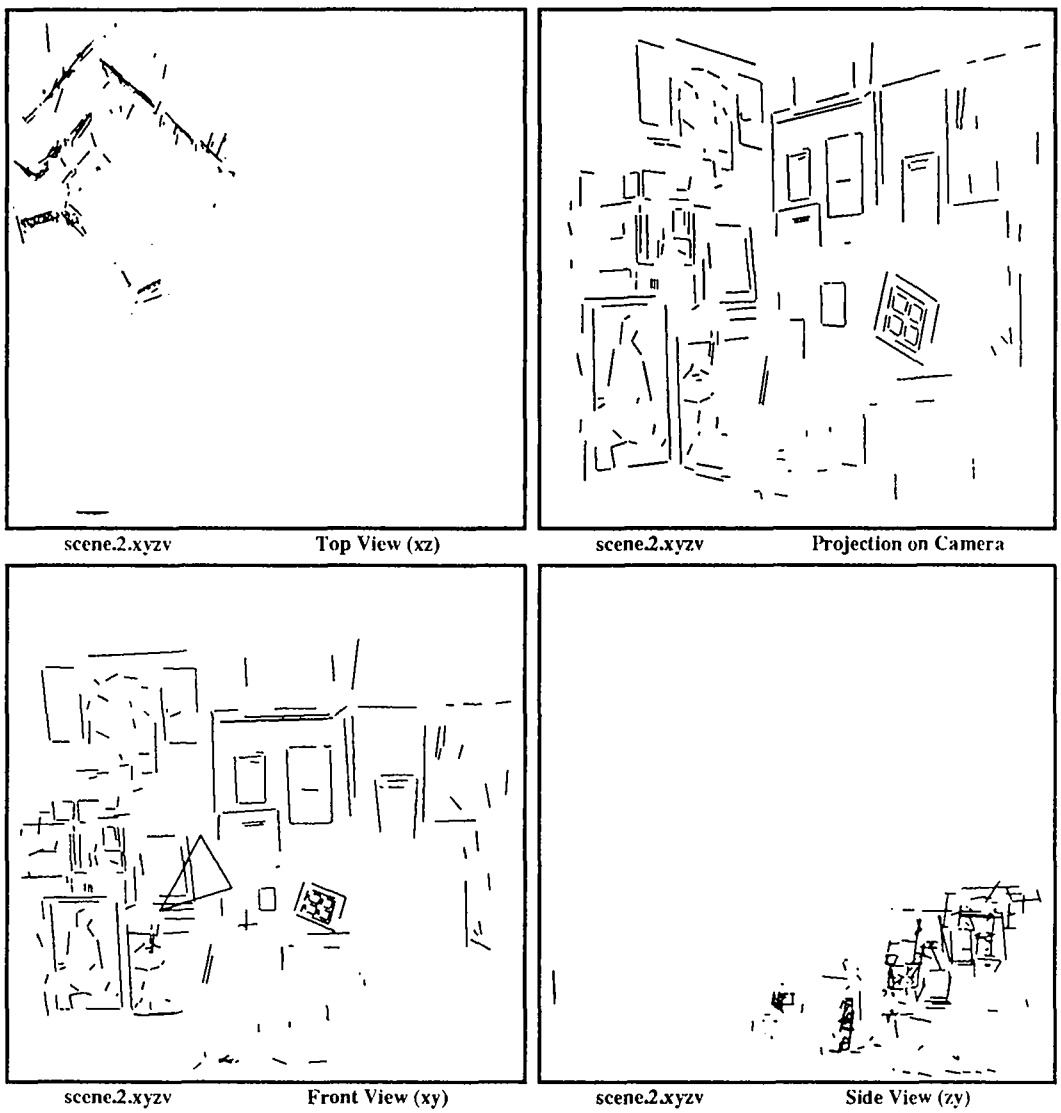



Fig. 12: Different views of stereo frame 2 (uniform scale)

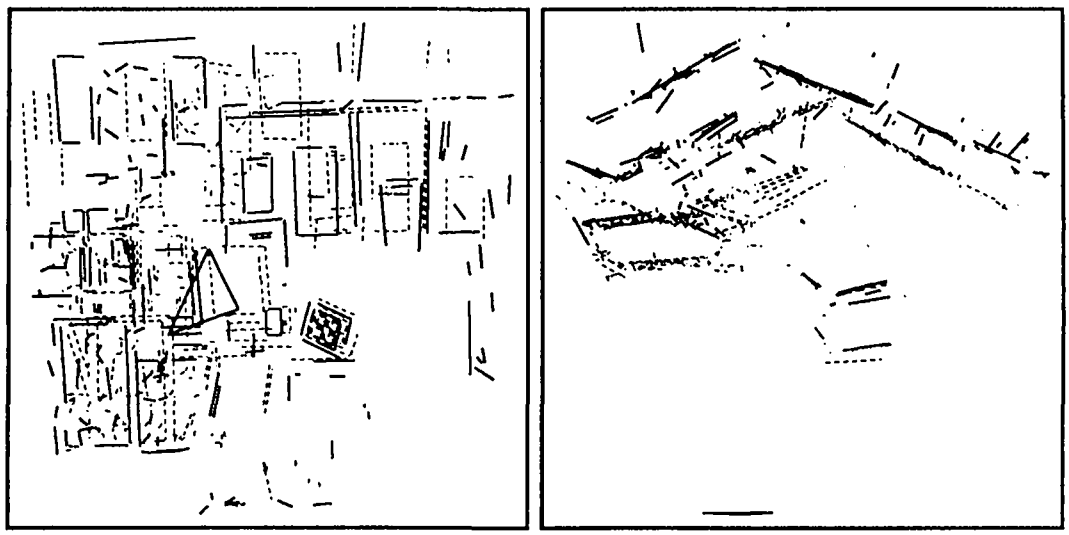


**Fig. 13:** Superposition of the two original frames: segments of Frame 1 are represented in dashed lines and those of Frame 2 in solid lines

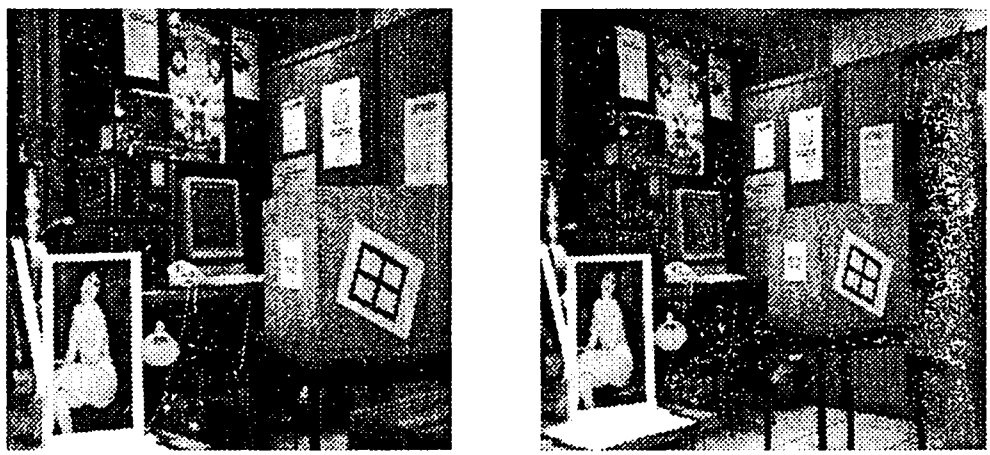


**Fig. 14:** Images taken by the first camera at two different instants

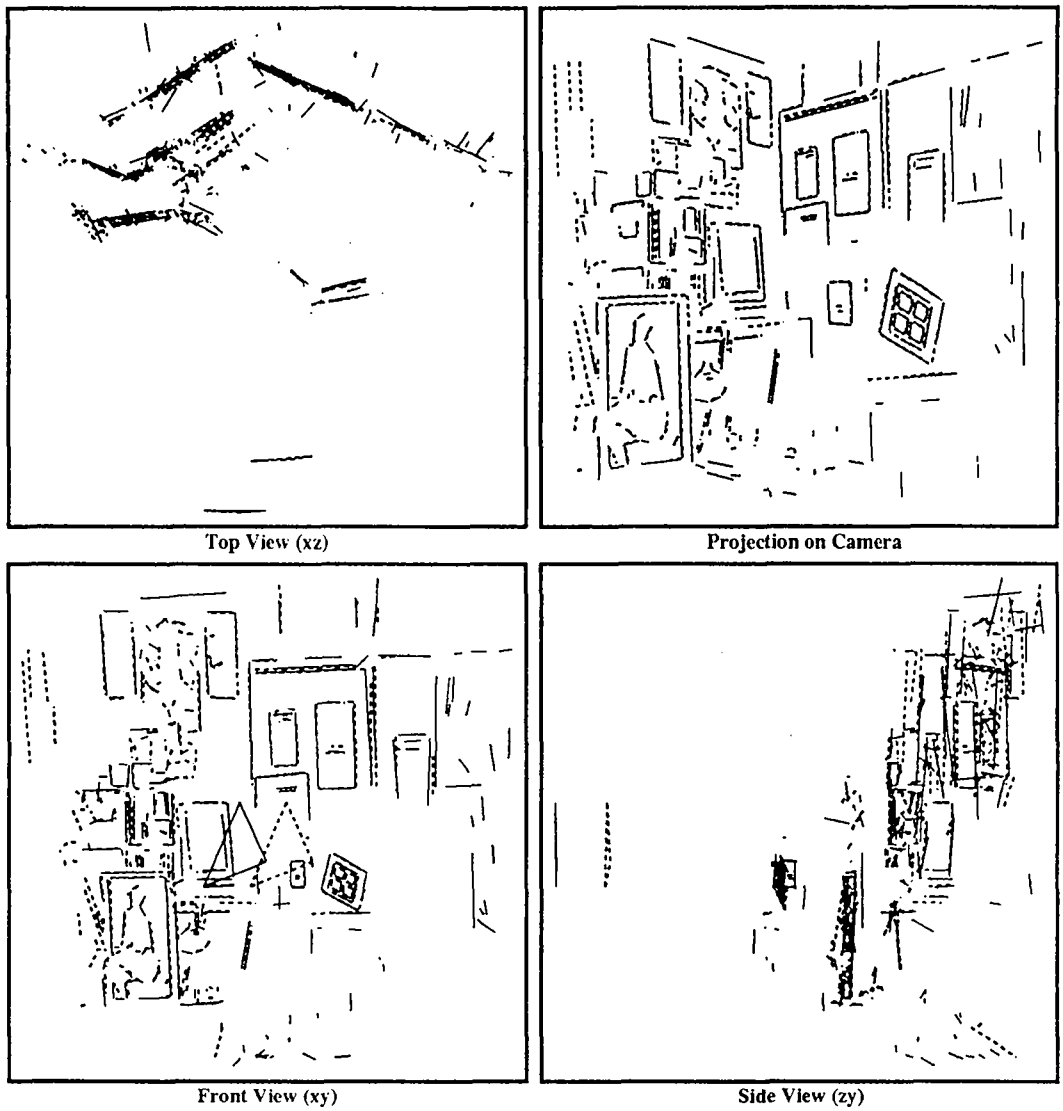
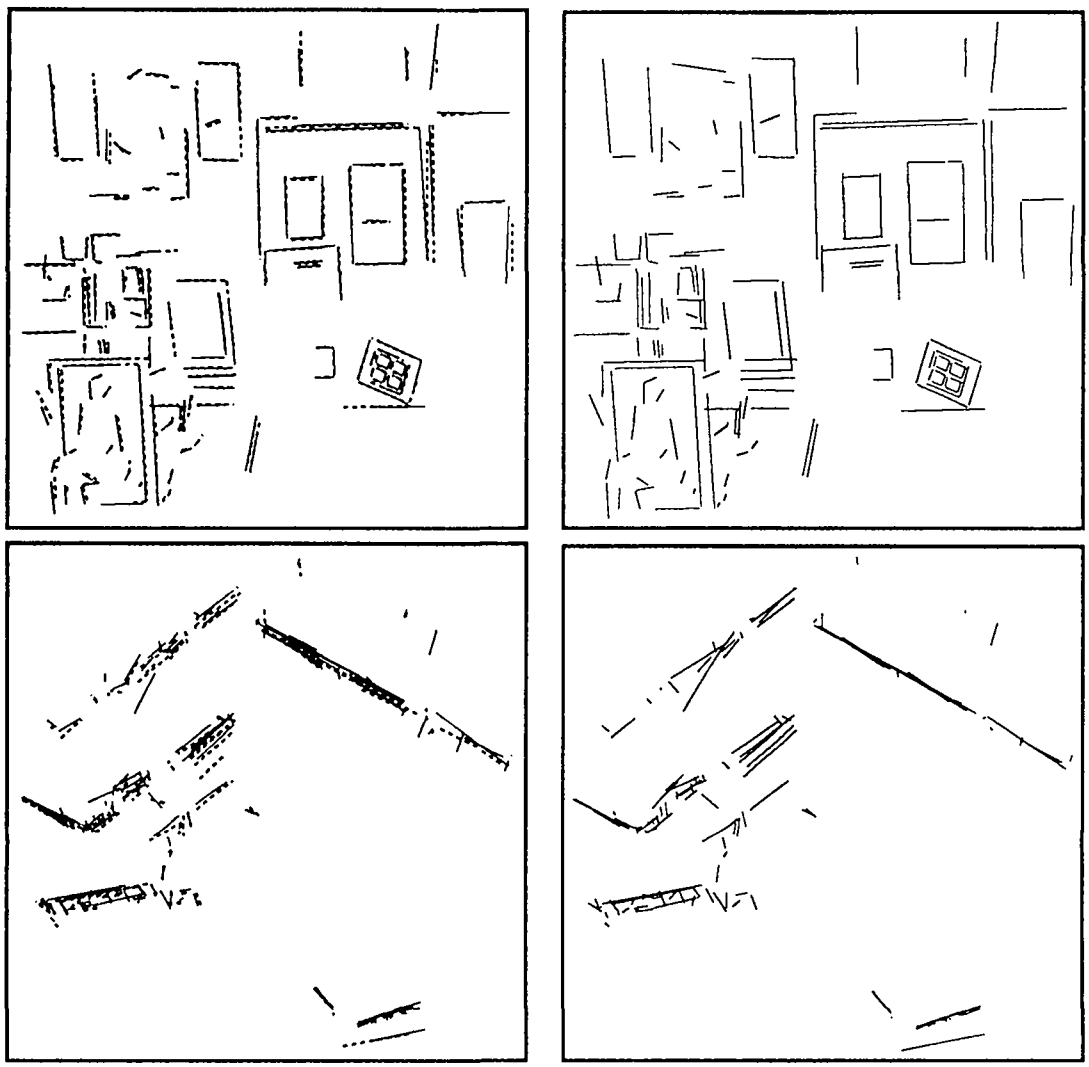



**Fig. 15:** Superposition of the transformed segments of Frame 1 (in dashed lines) and those of Frame 2 (in solid lines) in the coordinate system of Frame 2 (nonuniform scale)



(a) (b)

**Fig. 16:** (a) superposition of the matched segments after applying the estimated motion to the first frame, (b) fused segments (unmatched segments are not displayed)

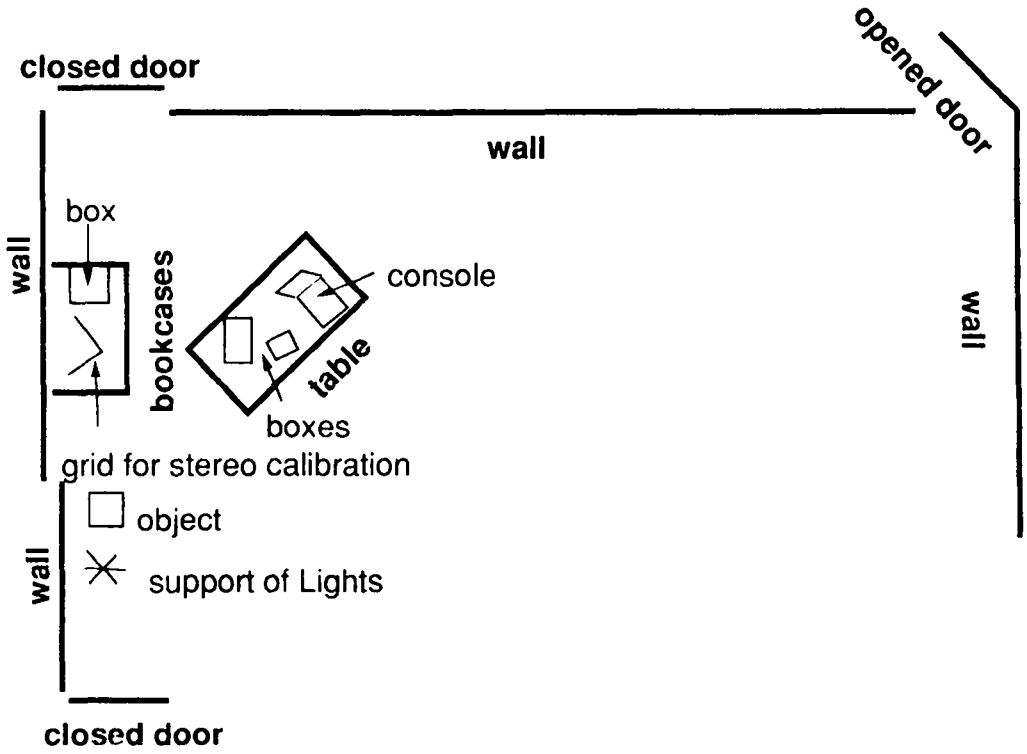
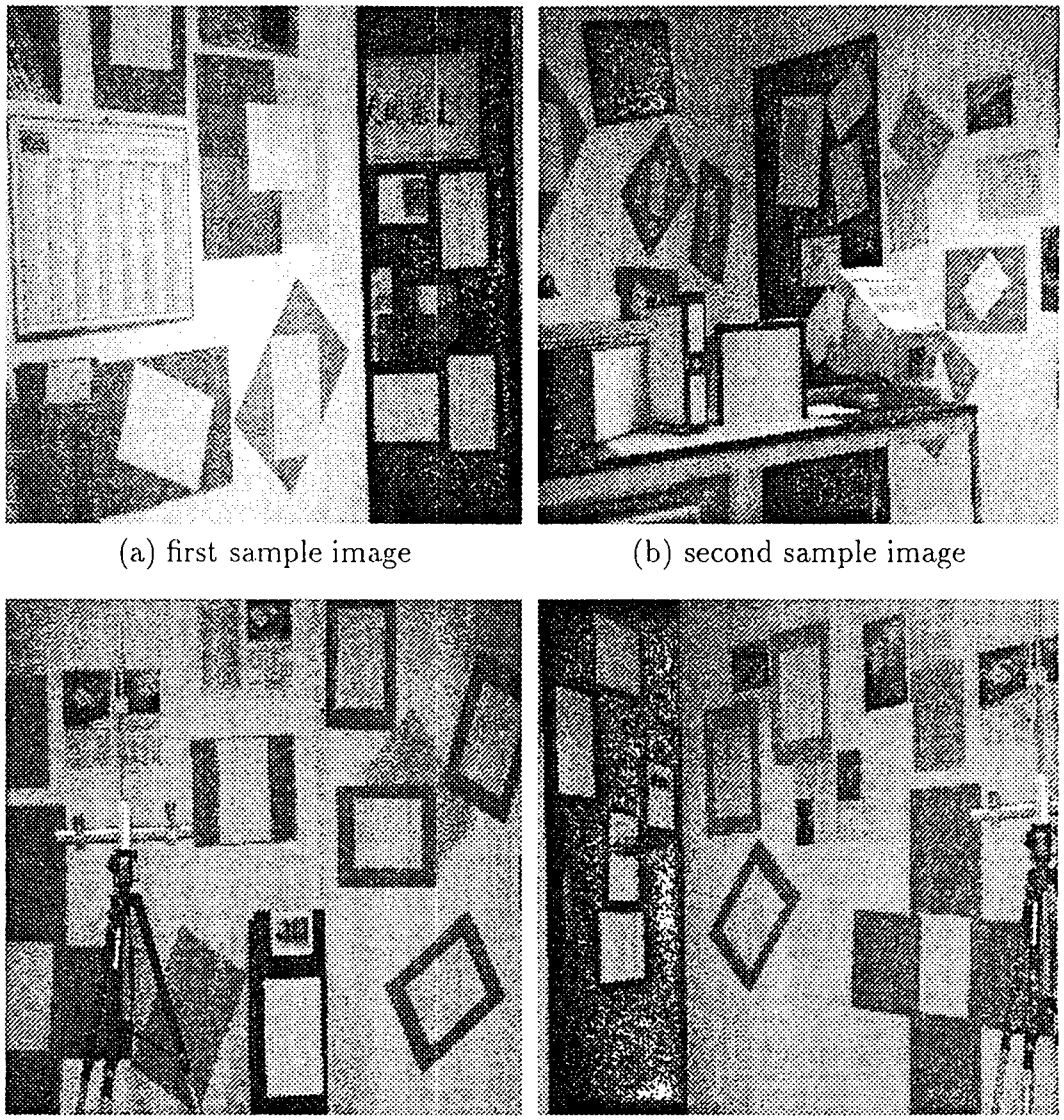



**Fig. 17:** Semantic description of the room



(a) first sample image (b) second sample image

(c) third sample image (d) fourth sample image

**Fig. 18:** Four sample views of the room

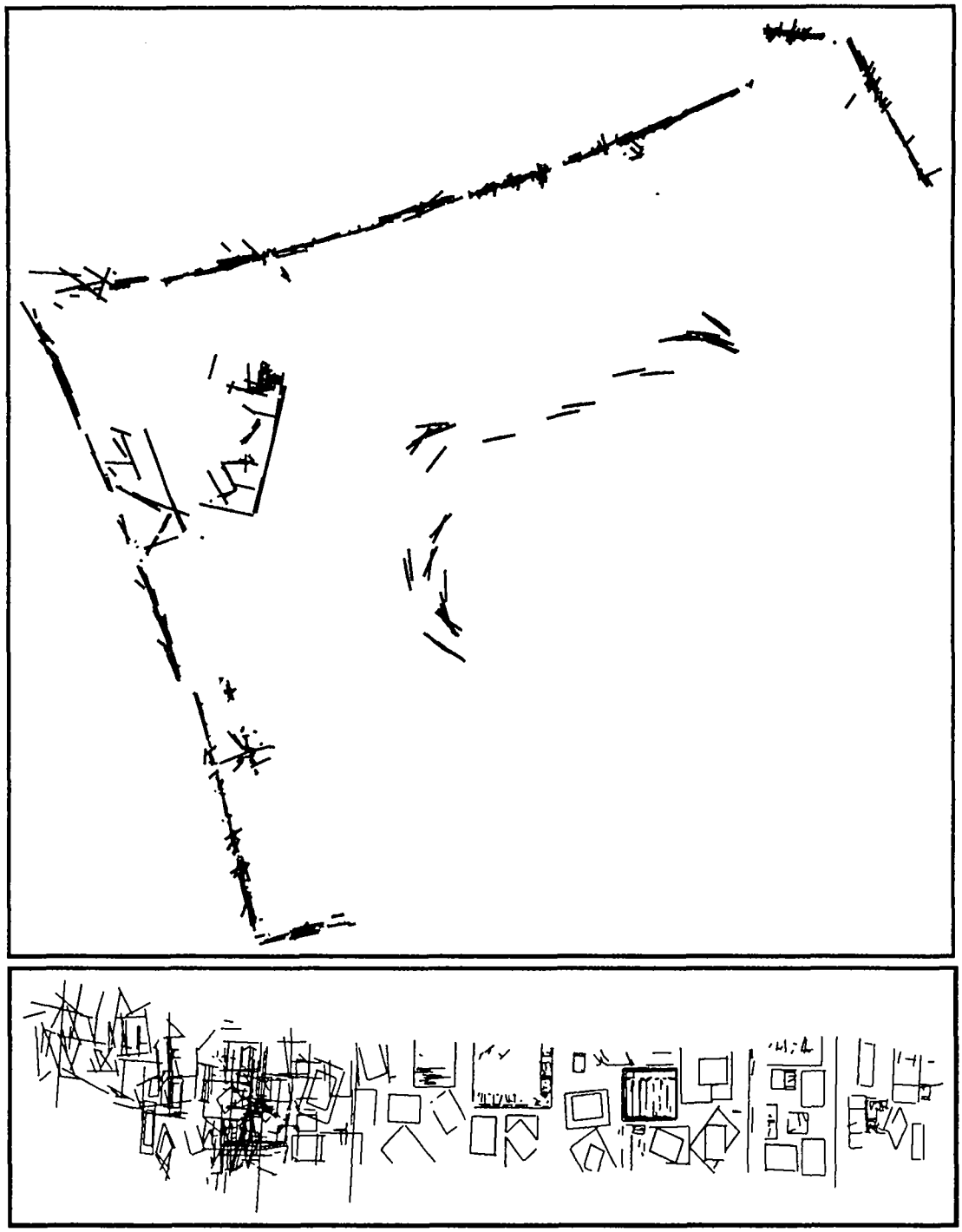


**Fig. 19:** The final 3D map of the room by integrating 35 3D frames obtained from stereo: top and front views (line segments in the middle part of the top view indicate the positions of the cameras)

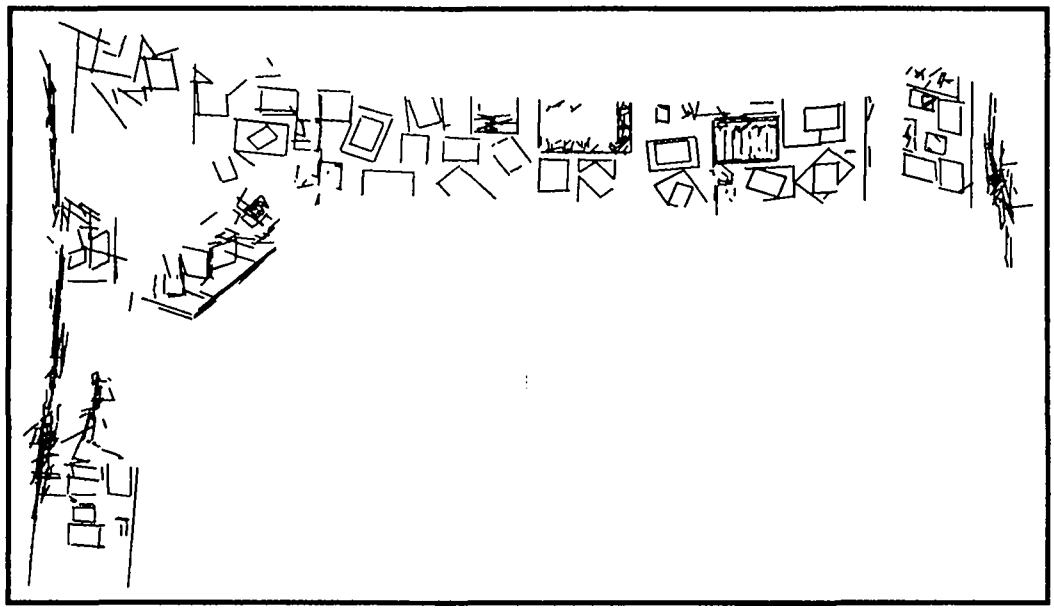


Fig. 20: First perspective view of the global map

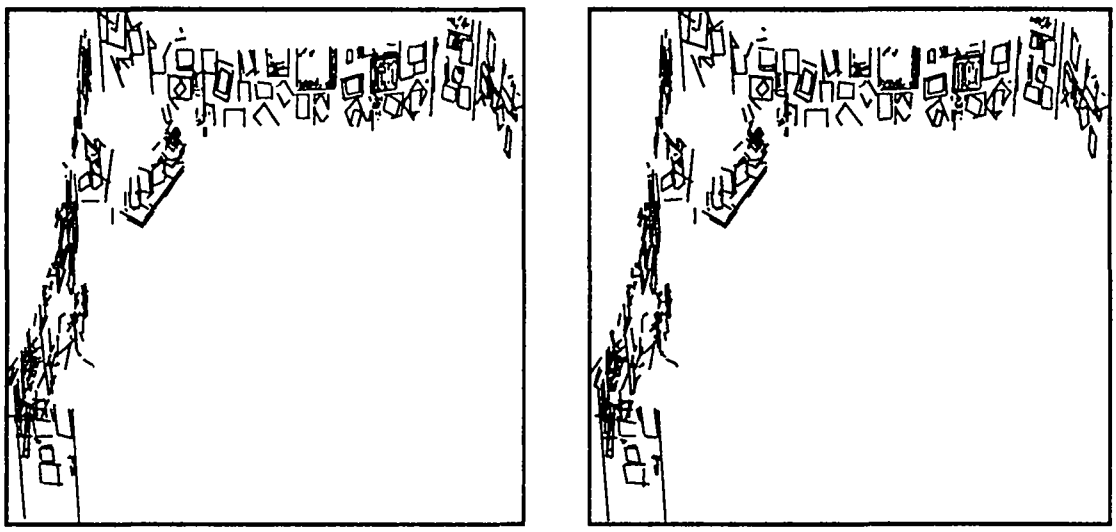


Fig. 21: A stereogram of the first perspective view of the global map

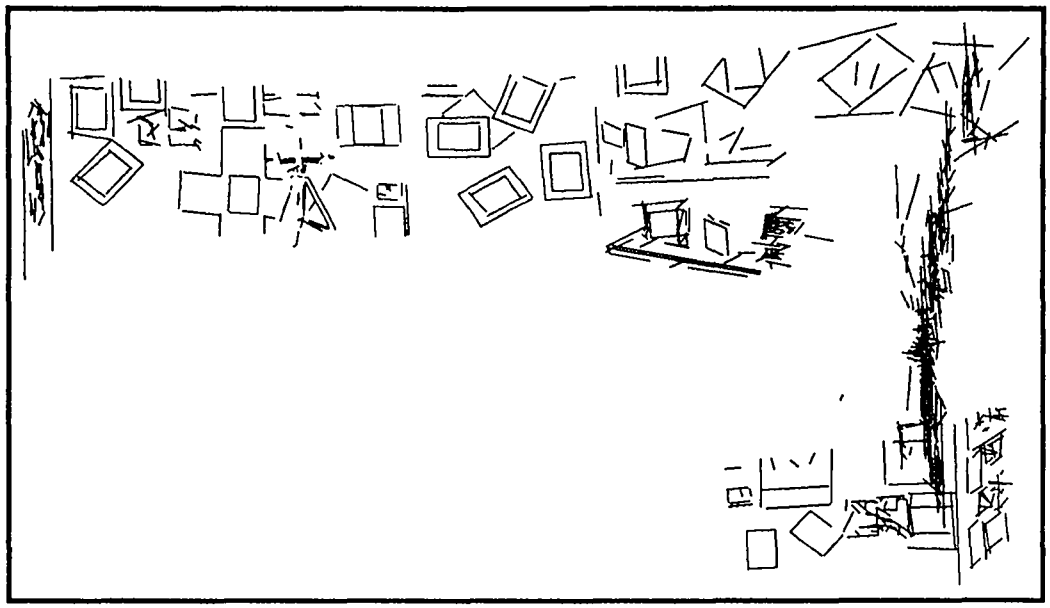


**Fig. 22:** Second perspective view of the global map

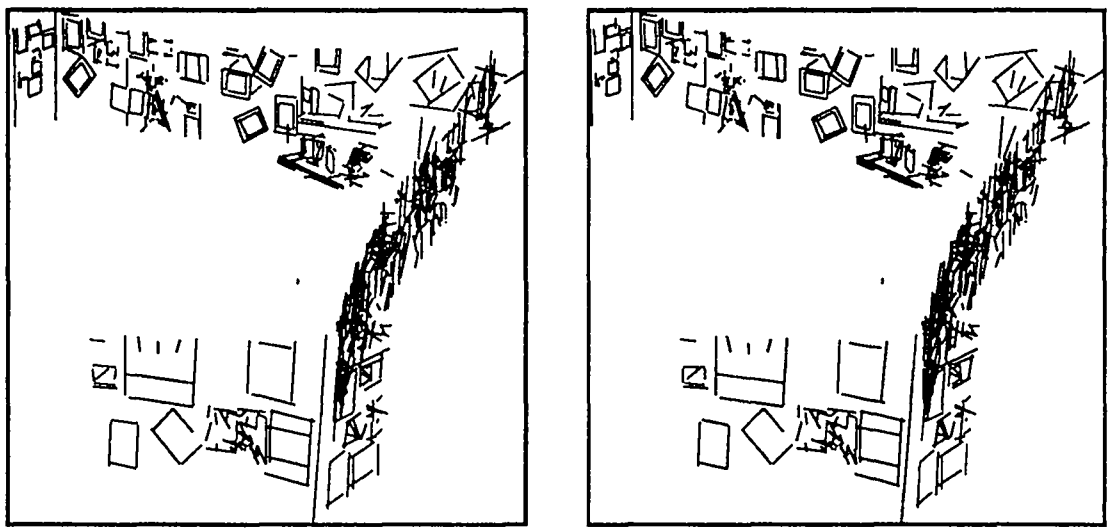


**Fig. 23:** A stereogram of the second perspective view of the global map

## 5.3 Fusion of a Long Sequence

We now describe the result of the integration of 35 stereo views taken when the robot navigates in a room. Merged segments are represented in the coordinate system related to the last position. As described in Sect. 2, the displacement of the robot is manually controlled with the aide of a graphic interface, such that there exists some common part of the view fields in the successive positions (the necessary condition that the motion algorithm succeeds). A labeled map of the room is shown in Fig. 17. In Fig. 18, we show four sample images taken by the first camera. Figure 19 shows the final 3D map obtained by integrating 35 stereo views of the room. One can easily establish the relation between the descriptions in Fig. 19 and in Fig. 17. In order to give an idea about the displacements of the robot in this experiment, we also show the position of the cameras at each instant in the room in the top view which is represented by a line segment in the middle part. Amongst all the displacements effectuated, the biggest rotation is of 17.4 degrees and the largest translation is of 683 millimeters.

From Fig. 19, several remarks can be made. Consider the upper left corner of the room in the top view. Large uncertainty can be observed, which is due to the fact that the robot is distant from that corner. Consider now the same corner in the front view. In fact, the front view displays the height information observed by the robot. Since the robot is more distant from the corner than from the other parts, it observes things in the corner higher than in the other parts of the room. The low part of the corner, however, is not seen by the robot, because the table in front of it hides it. The left wall is in fact composed of two noncoplanar parts. Figures 20 and 22 give two perspective views of the final 3D map to observe better the details of the room. We provide also two stereograms, each corresponding a perspective view (Figs. 21 and 23), which allow the reader to perceive the depth through cross-eye fusion.

There are 3452 segments in total in all stereo frames and we have only 839 segments seen at least twice in the final map (segments which appear only once are not counted). This shows that fusion is also very useful in reducing the memory requirements. Although the result is very encouraging, one can observe some distortion in the global map, very probably due to the stereo calibration or 3D reconstruction.

# 6 Conclusion

We have described in this article a system to incrementally build a world model of an unknown environment with a mobile robot. The model is, for the moment,

segment-based. A trinocular system is used to build a local map of the environment. A global map is obtained by integrating a sequence of 3D frames taken by a stereo system when the robot navigates in the environment. The emphasis has been on the integration of segments from multiple views. Our approach to the integration of multiple stereo views is very similar to those reported in the literature based on Bayesian estimation (Smith and Cheeseman, 1987; Ayache and Faugeras, 1987; Porrill, 1988; Moutarlier and Chatila, 1989; Grandjean and de Saint Vincent, 1989), although there is a slight difference in technical details. An important characteristic of our integration strategy is that a segment observed by the stereo system corresponds only to one part of the segment in space if it exists, so the union of different observations gives a better estimate on the segment in space. We have succeeded in integrating 35 stereo frames taken in our robot room. Although the results are very encouraging, several points need to be further investigated:

- The distortion in the global map should be analyzed in detail. It may result from camera calibration, 3D reconstruction or motion estimation.
- The implementation of the current system is based on 3D line segments. They are rather limited in their descriptive power and are usually not sufficient to describe complex scenes, e.g., outdoor scenes. The method described in this article can easily be extended to include stable points such as corners, but this is not sufficient, either. We are currently working on the modelization of curved objects.
- Geometric constraints (parallelism of segments, coplanarity of segments, etc.) can be imposed on the segment set to improve the accuracy of measurements.
- The resulting world model needs to be represented in a more symbolic manner, identifying, for example, walls, tables and doors in it.
- The analysis and decision module should be developed in order for the mobile robot to build *automatically* the world model. This is something we are doing (Buffa et al., 1990).

The primary goal in (Buffa et al., 1990) is obstacle avoidance and trajectory planning for an indoor mobile robot. We first project, on the ground plane, 3D line segments reconstructed by stereo to obtain a two-dimensional map. Those 2D segments are used to construct a tessellation of the ground plane through the Delaunay triangulation. We then determine free space by marking those triangles which are empty and generate collision free trajectories. As the mobile robot navigates, more 2D segments are available, and a technique

adapted from that developed in this article is used to fuse 2D line segments and to build incrementally a global 2D map. The process is iterated until the task is accomplished.

Several similar ideas can be found in the *3D Mosaic* scene understanding system (Herman, 1986). That system is intended for incrementally generating a 3D model of a complex scene from multiple images. The primitives used are edges and vertices. It differs from the work described in this article in at least the following points. Firstly, no extended experiments have been carried out using this system. Only the merging result of a 3D frame obtained from a pair of stereo aerial images and a manually generated 3D frame has been reported. Secondly, the motion (coordinate transformation) between successive views has been assumed to be known, which makes the matching problem trivial. Finally, uncertainty in the model and in the measurements has not been systematically addressed. One important point of the 3D Mosaic system from our point of view is that part of the knowledge of planar-faced objects has been explicitly formulated. Such knowledge may constitute a good starting point for us to interpret fused data towards derivation of a symbolic model.

# References


Ayache, N. 1991. *Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception.* Cambridge: MIT Press.

Ayache, N., and Faugeras, O. 1987 (June, London, UK). Building, registrating and fusing noisy visual maps. *Proc. First Int'l Conf. Comput. Vision*, pp. 73–82.

Ayache, N., and Faugeras, O. D. 1989. Maintaining Representations of the Environment of a Mobile Robot. *IEEE Trans. RA*, 5(6):804–819.

Blostein, S., and Huang, T. 1984 (December, Denver, CO). Estimation 3-D motion from range data. *Proc. First Conf. Artif. Intell. Applications*, pp. 246–250.

Blostein, S., and Huang, T. 1987. Error analysis in stereo determination of a 3-D point position. *IEEE Trans. PAMI*, 9(6):752–765.

Brady, M. 1985. Artificial intelligence and robotics. *Artif. Intell.*, 26:79–121.

Buffa, M., Faugeras, O., and Zhang, Z. 1990 (July, Science University of Tokyo, Japan). Obstacle avoidance and trajectory planning for an indoors mobile robot using stereo vision and Delaunay triangulation. *Proc. Roundtable Discussion on Vision-Based Vehicle Guidance*, pp. 12–1–12–8.

Chen, H., and Huang, T. 1987. An algorithm for matching 3-D line segments with application to multiple-object motion estimation. *Proc. IEEE Workshop Comput. Vision*, pp. 151–156.

Chen, H., and Huang, T. 1988. Maximal matching of 3-D points for multiple-object motion estimation. *Pattern Recog.*, 21(2):75–90.

Durrant-Whyte, H. 1988a. Sensor models and multisensor integration. *Int'l J. Robotics Res.*, 7(6):97–113.

Durrant-Whyte, H. 1988b. Uncertain geometry in robotics. *IEEE J. RA*, 4(1):23–31.

Faugeras, O., and Hebert, M. 1986. The representation, recognition, and locating of 3D shapes from range data. *Int'l J. Robotics Res.*, 5(3):27–52.

Faugeras, O., Ayache, N., and Faverjon, B. 1986 (April, San Francisco, CA). Building visual maps by combining noisy stereo measurements. *Proc. Int'l Conf. Robotics Automation*, pp. 1433–1438.

Faugeras, O., Ayache, N., and Zhang, Z. 1988 (Rome, Italy). A preliminary investigation of the problem of determining ego- and object motions from stereo. *Proc. 9th Int'l Conf. Pattern Recog.*, pp. 242–246.

Grandjean, P., and de Saint Vincent, A. R. 1989. 3-D modeling of indoor scenes by fusion of noisy range and stereo data. Technical Report 89068, LAAS, Toulouse, France.

Grossmann, P. 1989 (San Diego, CA). From 3D line segments to objects and spaces. *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pp. 216–221.

Hager, G. 1988. Active reduction of uncertainty in multi-sensor systems. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

Herman, M. 1986. Representation and incremental construction of a three-dimensional scene model. *Techniques for 3-D Machine Perception*, ed. A. Rosenfeld. North-Holland: Elsevier, pp. 149–183.

Jazwinsky, A. 1970. *Stochastic Processes and Filtering Theory.* New York: Academic.

Jezouin, J., and Ayache, N. 1990 (December, Osaka, Japan). 3D structure from a monocular sequence of images. *Proc. Third Int'l Conf. Comput. Vision*, pp. 441–445.

Kim, Y., and Aggarwal, J. 1987. Determining object motion in a sequence of stereo images. *IEEE J. RA*, 3(6):599–614.

Lozano-Pérez, T., and Wesley, M. 1979. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of ACM*:560–570.

Lustman, F. 1987. Vision stéréoscopique et perception du mouvement en vision artificielle. Dissertation, University of Paris XI, Orsay, Paris, France.

Matthies, L., and Elfes, A. 1988 (Philadelphia, PA). Integration of sonar and stereo range data using a grid-based representation. *Proc. Int'l Conf. Robotics Automation*, pp. 727–733.

Matthies, L., and Shafer, S. A. 1987. Error modeling in stereo navigation. *IEEE J. RA*, 3(3):239–248.

Moutarlier, P., and Chatila, R. 1989 (August 28–31, Tokyo, Japan). Stochastic multisensory data fusion for mobile robot location and environment modelling. *Proc. Int'l Symposium Robotics Res.*, pp. 207–216.

Porrill, J. 1988. Optimal combination and constraints for geometrical sensor data. *Int'l J. Robotics Res.*, 7(6):66–77.

Roberts, K. 1988 (June, Ann Arbor, Michigan). A new representation for a line. *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pp. 635–640.

Robles, J. 1988. Planification de trajectoires et évitement d'obstacles pour un robot mobile équipé de capteurs à ultrasons. Rapport de DEA, University of Paris XI, Orsay, Paris, France.

Smith, R., and Cheeseman, P. 1987. On the representation and estimation of spatial uncertainty. *Int'l J. Robotics Res.*, 5(4):56–68.

Thonnat, M. 1988. Semantic interpretation of 3-D stereo data: finding the main structures. *Int'l J. Pattern Recog. Artif. Intell.*, 2(3):509–525.

Tournassoud, P. 1988. Planification de trajectoires en robotique: complexité et approche pratique. Dissertation, University of Paris XI, Orsay, Paris, France.

Tsuji, S., and Zheng, J. 1987 (Milan, Italy). Visual path planning by a mobile robot. *Proc. Int'l Joint Conf. Artif. Intell.*, pp. 1127–1130.

Zhang, Z. 1990. Motion analysis from a sequence of stereo frames and its applications. Dissertation, University of Paris XI, Orsay, Paris, France.

Zhang, Z., and Faugeras, O. 1989 (March, Irvine, CA). Calibration of a mobile robot with application to visual navigation. *Proc. IEEE Workshop Visual Motion*, pp. 306–313.

Zhang, Z., and Faugeras, O. 1991. Determining motion from 3D line segments: a comparative study. *Image and Vision Computing*, 9(1):10–19.

Zhang, Z., Faugeras, O., and Ayache, N. 1988 (December, Tampa, FL). Analysis of a sequence of stereo scenes containing multiple moving objects using rigidity constraints. *Proc. Second Int'l Conf. Comput. Vision*, pp. 177–186.



Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique.

ISSN 0249 - 6399