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

Internal Report: 1999-V01

Motion and structure for vision-based navigation[1]

Sagüés C., Guerrero J.J.

If you want to cite this report, please use the following reference instead:
Motion and structure for vision-based navigation, Sagüés C., Guerrero J.J., *Robotica*, Vol. 17(4), pages 355-364, 1999.

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

Motion and structure for vision-based navigation

C. Sagüés & J.J. Guerrero †

Abstract

This paper is aimed to develop a complete algorithm to determine the robot motion and the scene structure using a monocular vision system. It is based on straight lines and significant points extracted on them. In this way, an agreement between the problems to extract or to match points and the limitations of infinite lines to compute structure and motion is established. Some plentiful geometrical relations of the lines in the scene are exploited to disambiguate the coupling between the rotation and the translation of the camera. Several real images have been used to validate the proposed method. The algorithm has been considered for navigation of a mobile robot running in man made environments.

Keywords

Robot vision, visual motion, motion estimation, robot navigation, motion and structure, straight lines, significant points.

1 Introduction

Mobile autonomous robots are actually able to execute tasks indoors, where the ground is assumed to be horizontal and the robot localization is obtained using specific landmarks. However, without these assumptions, powerful perception systems to estimate the 3D motion of the robot and the scene structure are required. Vision is a sensor broadly used.

The methods to recover structure and motion from vision have been widely studied and revised in the last years [1]. Two types of methods have been proposed: optical flow based and correspondence based. The methods based on correspondences allow higher disparities between images solving the problem in a better conditioned way. The infinite line is a feature broadly used to compute motion [2], [3], [4]. Working with lines, three images at least are needed to obtain both, the camera motion and the 3D scene structure. There are also works which explicitly consider points to compute motion [5]. Corresponding points allow to solve motion and structure problem more easily than corresponding lines, but extracting and matching points is normally more difficult. In our paper points are used, but associated to lines. In this way, an agreement between the problems to extract or to match points and the limitations of infinite lines to compute structure and motion is established [6].

In real situations, where noise is present, it is difficult to obtain good solutions, but the best is obtained with many features and some global nonlinear optimization [7]. As was mentioned above, points along lines are used in our work. We propose an anisotropic noise model for the location of the point that takes into account the noise of the projected line (the location error of the point is higher along the line than across it). We have considered a nonlinear optimization to obtain the motion assuming rigidity of the scene. The proposed model allows to weigh the measures taking into account the point and its line support in the image. As in other works [8], the vertical cue is also considered trying to provide relevant qualitative information about the structure of the scene to recover structure and motion robustly. This significant information allows to disambiguate the problem of coupling between translations along and rotations around axes parallel to the image plane. This assumption

†Departamento de Informática e Ingeniería de Sistemas, Centro Politécnico Superior, UNIVERSIDAD DE ZARAGOZA, María de Luna 3, E-50015 ZARAGOZA, SPAIN, Phone 34-976-761940, Fax 34-976-762111, email: csagues@posta.unizar.es, jguerrer@posta.unizar.es

has been previously exploited [4]. However, in our work the motion is globally computed using not only lines, but also characteristic points on them.

In our paper, a complete chain to obtain motion and structure from a camera on a mobile robot is presented. In the technical literature the aspects involved in this problem (extraction of features, matching, motion and structure computation, etc) have been treated in a separate way. We use a single camera without geometric map of the scene. Only some assumptions about the general aspect of the environment are taken into account. Thus, straight lines are expected to be plentiful and the 3D lines are supposed to be mainly vertical and horizontal, which are reasonable assumptions in man made environments.

The process starts with the extraction and matching of lines and characteristic points along them (§2). After that, the representation of the features used to compute 3D information is presented (§3). In §4 an estimation of the 3D direction of significant lines is obtained. Afterwards, in §5 the 3D motion of the camera and the structure of the environment of the robot are computed. The method uses significant lines and points in two images. Experimental examples using real images to support the algorithm are also presented along the paper.

2 Extraction and matching of features

In a correspondence based approach the features must be extracted from the images, and the correspondences must be computed. We use lines and characteristic points on them, that have been extracted and matched as explained below.

2.1 Extraction of lines

Once the image is acquired, straight lines are extracted using our implementation of the method proposed by Burns [9]. This method computes spatial brightness gradients to segment the image. Pixels having the gradient magnitude larger than a threshold are grouped into regions of similar direction of brightness gradient. These groups are named line-support regions (LSR). A planar brightness surface is fitted to each LSR by a least-squares approach, predicting the brightness in function of the image coordinates. The line is obtained as the intersection of this brightness plane and an horizontal plane of mean brightness in the LSR. The parameters of the line in the image are obtained with subpixel accuracy.

Using this line detector, we obtain in addition to the geometrical parameters of the lines, some attributes related with their brightness (contrast, average gray level, steepness) and some quality attributes (deviation from straightness). These attributes provide very useful information to select and identify them. After the extraction, the lines are selected in function of its length, its contrast and its deviation from straightness in order to have few but good lines.

2.2 Extraction of characteristic points

Characteristic points are usually attached to some edge. So, focusing the search of points on the extracted lines turns out easier than searching points on all the image. In a first approach, the tips obtained by the extractor of lines could be used as characteristic points. However, they are not good enough because edge detectors do not work properly to obtain points [10]. Besides that, there are tips of lines that correspond with well-defined points, but in other cases the tips stump unclear.

There are two kinds of methods to obtain points [11]. In the first class, points with maximum curvature on an edge chain are searched. In the second class, points are searched on the intensity image by using heuristic techniques such as the Moravec operator [12] or by measuring brightness variations [13]. Normally, the later methods have higher computational cost [14], but higher precision.

We obtain the points from brightness variations, but we search points along the line. In this way, the computational cost is drastically reduced, and isolated noise points are avoided. We consider as characteristic points those whose gradient multiplied by the curvature is a maximum [13].

Taking the derivative of the orientation of the gradient (E_x, E_y), in the direction orthogonal to the gradient $n_\perp$, we have

a b

Figure 1: a.- Image of the laboratory with the lines extracted (filtered in gradient and length). b.- Characteristic points selected along the lines.

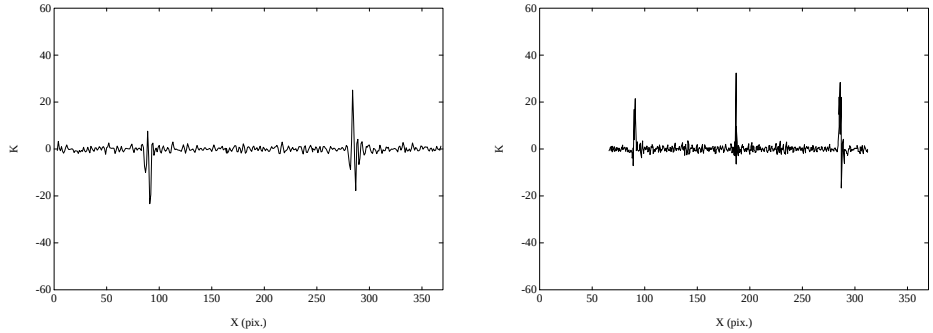

Figure 2: Examples of the K measurement along two lines in a real image. In the first example two points are easily selected. In the second one, three characteristic points are obtained.

$$c = \frac{d\left(\arctan\frac{E_y}{E_x}\right)}{d\left(n_\perp\right)} = \frac{2\,E_{xy}\,E_x\,E_y - E_y^2\,E_{xx} - E_x^2\,E_{yy}}{\left(E_x^2 + E_y^2\right)^{\frac{3}{2}}}$$

where each subindex indicates a spatial derivative.

The measure of cornerness is obtained multiplying the above expression by the modulus of the gradient

$$K = \frac{2\,E_{xy}\,E_x\,E_y - E_y^2\,E_{xx} - E_x^2\,E_{yy}}{\left(E_x^2 + E_y^2\right)} \tag{1}$$

We search along the straight lines for points with maximum K. The cornerness operator K supplies (with little computational cost) a quantitative measurement of point goodness.

In Fig. 1 it can be seen the extracted lines and the points selected along them from a scene of our laboratory. In Fig. 2 examples of the cornerness measurement K along two straight lines in a real image are represented. The K operator is null on straight lines, and it can be observed some maximum values of K, which are related to well-defined points.

2.3 Matching

The algorithm proposed in this paper solves motion and structure from at least two images and needs the correspondence of features between them. To make easier the matching problem, some intermediate images are taken. We have treated the correspondence problem by tracking straight lines in the image with a predict-match-update loop using a Kalman filter [15]. A constant velocity

model has been heuristically selected to predict the features representation in the following image. A nearest neighbor tracking approach as in [16], has been developed [17]. However, besides the classical location values, two image brightness attributes of the line are used in the tracking process. These attributes are the average gray level and the mean contrast. The bright attributes are crucial to match lines when neither the structure nor the camera motion are known, because geometrical constraints are only valid locally and they must be imposed in an heuristic way. Besides that, the matching using these bright attributes is made nearly in parallel to the geometrical matching. More details about this tracker are given in a previous paper [18].

We show in Fig. 3 an example of corresponding lines in two images using the proposed algorithm. The computation of corresponding straight lines gives indirectly corresponding points on them.

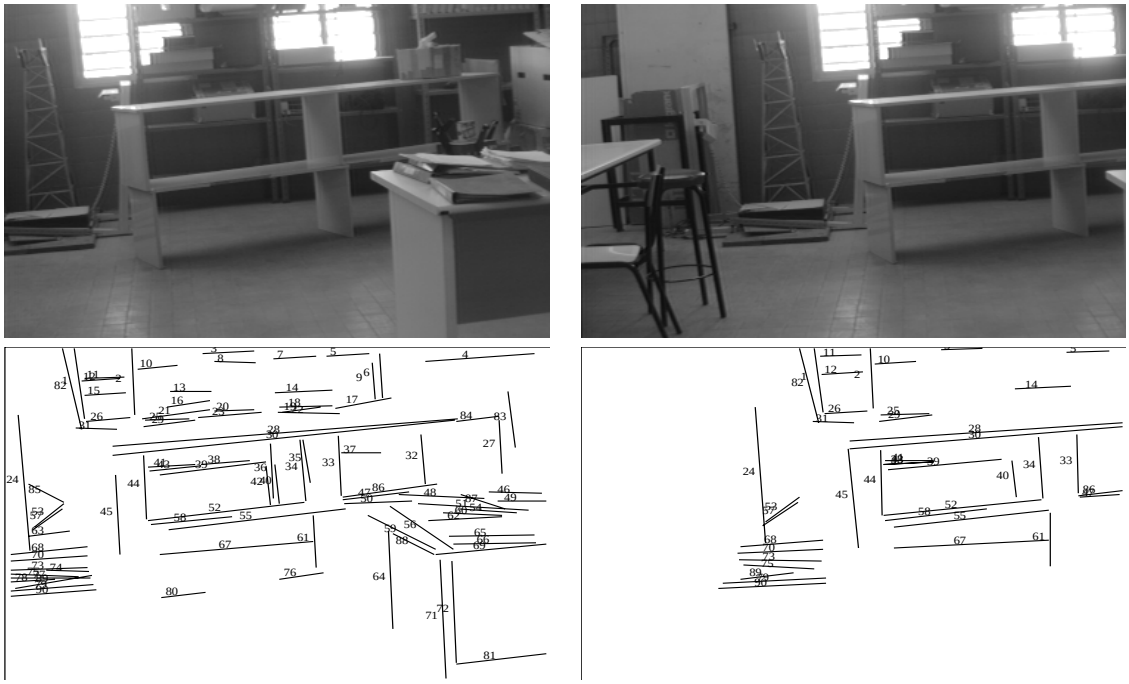

Figure 3: Example of matched straight lines in two images. In this case, six intermediate images have been used to track them.

3 Geometric representation of the features

To compute the camera motion and the scene structure, the projected features must be represented and related to the 3D space according to the projective nature of the vision process. The classical pinhole camera model is assumed. The origin of the camera reference system is on the projection center. The Z axis is aligned with the focal axis and the focal length is considered to be the unit. In this way, a normalized retina (located in front of the camera) is defined. The location of a feature in the digitalized image can be easily transformed to the normalized retina from the parameters obtained in the camera calibration process [19].

We represent a infinite line in the image with two parameters (Fig. 4). These are the ϕ_l and θ_l angles defining the normal $\mathbf{n}$ of the projection plane of the line,

$$\mathbf{n} = (cos\phi_l cos\theta_l, sin\phi_l cos\theta_l, -sin\theta_l)^T$$

The angle ϕ_l describes the orientation of the line with respect to y axis of the normalized retina. As the focal length is the unit, the distance in the normalized retina from the origin to the line can be expressed as $tan\theta_l$.

Another parameter ψ_i for each characteristic point (i) defines its location along its associated line, in such a way that the unit vector in the direction of the projection line of the point, expressed in the

camera reference system is

$$\mathbf{p}_i = \begin{pmatrix} cos\phi_l sin\theta_l cos\psi_i + sin\phi_l sin\psi_i \\ sin\phi_l sin\theta_l cos\psi_i - cos\phi_l sin\psi_i \\ cos\theta_l cos\psi_i \end{pmatrix}$$

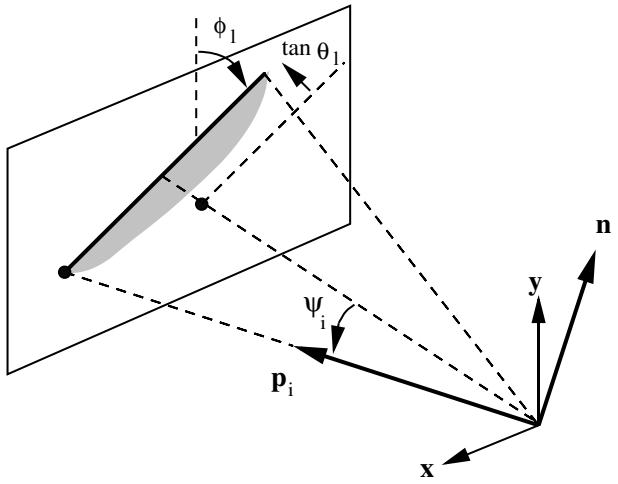

Figure 4: Representation of a projected line with a characteristic point on it.

From this representation, if a point (with coordinates x_i and y_i in the normalized retina) belongs to a projected line (with parameters ϕ_l and θ_l), we have

$$x_i\, cos\phi_l + y_i\, sin\phi_l - tan\theta_l = 0$$

that is the equation of the line applied to the point. The parameter that we use to represent the location of the point along the line can be obtained as,

$$\psi_i = atan2\left(x_i\, sin\phi_l - y_i\, cos\phi_l, \frac{1}{cos\theta_l}\right)$$

We take ϕ_l in the 2π range in such a way that the normal to the plane of projection ($\mathbf{n}$) goes towards the direction of the spatial brightness gradient, from dark to light. The angle θ_l takes values from $-\frac{\pi}{2}$ to $+\frac{\pi}{2}$. As real cameras have a small field of view, the angle θ_l will be small for all lines that appear in the image. The angle $\psi_i \in [-\frac{\pi}{2}, +\frac{\pi}{2}]$ and it will also be less than the camera field of view. The sole singularity of this representation appears when $\theta_l = \pm\frac{\pi}{2}$, and therefore, we are far from the singularities with lines and points that can appear in the image.

4 Direction of significant lines

The vertical cue provides information which gives robustness to the computation of motion and structure [8]. We look for a good vertical cue making simple assumptions about the environment. Normally in man made environment vertical and horizontal lines are dominant. So, once the lines are extracted and matched, an initial classification of lines as vertical and horizontal is carried out, using the supposed vertical direction.

A rotation with two degrees of freedom is computed to estimate the vertical direction. The rotation $\mathbf{R}_{rc} = Rot(z, \phi_z^r) \cdot Rot(x, \psi_x^r)$ is obtained using the supposed vertical lines, in such a way that the supposed vertical lines would appear in a rectified image as parallel and vertical (in the direction of the y axis). We compute the angles ϕ_z^r, ψ_x^r that minimize

$$\sum^{j \in vert} [\hat{\mathbf{y}} \cdot \mathbf{R}_{rc} \mathbf{n}_j]^2 \qquad (2)$$

where $\hat{\mathbf{y}} = (0, 1, 0)^T$, $\cdot$ is the dot product between vectors, and $\mathbf{n}_j$ is the normal to the plane of projection of each supposed vertical line.

After that, a vertical rectification of the projected features is carried out (Fig. 5). The rectification is made rotating (by $\mathbf{R}_{rc}$) the $\mathbf{n}_j$ and $\mathbf{p}_i$ vectors, that represent the projected features in the camera reference system. This rectification will be used to make more robust the computation of the 3D direction of each significant line.

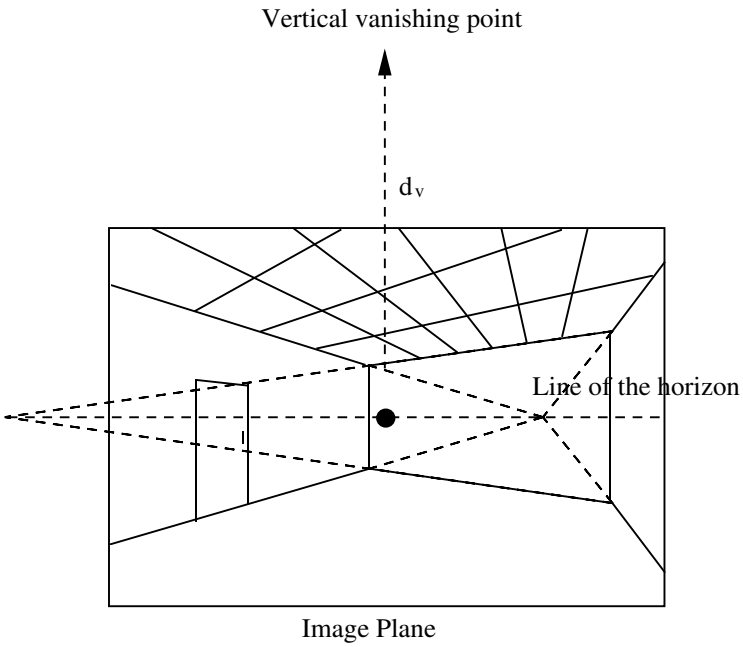

Figure 5: The rectification process transforms a general view in a vertical image, in which the line of the horizon is in the image center and all projected vertical lines appear parallel, also being perpendicular to the line of the horizon.

Once the rectified image has been obtained, the line of the horizon will be horizontal and it will be on the image center (Fig. 5). The final aim of this process is to provide the vanishing point of the significant lines. Using the line representation in the rectified image, an estimation of the 3D direction of each significant line is obtained (Fig. 6). The vertical lines vanish in $\mathbf{d}_j = \hat{\mathbf{y}}$. For horizontal lines the vanishing point is obtained as the intersection of the projected line and the line of the horizon. Thus, taking the representation of the line in the rectified image, the 3D direction can be estimated as

$$\mathbf{d}_j = \frac{\hat{\mathbf{y}} \times \mathbf{n}_j}{\| \hat{\mathbf{y}} \times \mathbf{n}_j \|}$$

where $\times$ is the cross product of vectors.

5 Motion and structure computation

As was mentioned above, we compute motion using identified points along lines. In this way, an agreement between the problems to extract or to match points and the limitations of infinite lines to compute structure and motion is established [6]. An anisotropic noise model for the location of the point that takes into account the noise of the projected line has also been proposed [20].

Besides that, to have information of relative depth, the constraints in the direction of significant lines provided by the previous process (§4) are considered. This information allows to uncouple translations along an axis with rotations around its perpendicular axis in the image plane. This is very important because a little error in the rotation computation brings about large errors in the computation of structure and translation.

Figure 6: The intersections of significant lines and the line of the horizon provide an estimation of their 3D direction.

5.1 Motion computation

We take the first camera reference system as the basic reference system. The matrix of rotation from the first to the second camera reference system is named $\mathbf{R}$. The vector $\mathbf{t}$ expresses the translation of the camera from the first camera location to the second.

The problem is posed as the estimation of the camera motion given a discrete description of the image deformation from one image to the next. The corresponding features in two images previously computed, are used as the description of the image changes due to motion. The image measurements are complemented by a measurement of their uncertainty. The location uncertainty of a point (i) along a line (l) is represented by a covariance matrix. It is composed of the line orientation covariance $\sigma^2_{\phi_l}$, the line position covariance $\sigma^2_{\theta_l}$, and the location covariance of the point along the line $\sigma^2_{\psi_i}$ which is supposed to be the biggest.

From two corresponding points in two images the epipolar constraint can be formulated [21]. If we express it in the first reference system, we can write for each point (i),

$$\mathbf{p}_{1i} \cdot (\mathbf{t} \times \mathbf{R}\,\mathbf{p}_{2i}) = 0 \tag{3}$$

where the subscripts 1 or 2 indicate the first or second image frame. This equation expresses that the translation vector must be coplanar with the two projection lines of the point in both images (Fig. 7).

Hypotheses of direction for several lines in each image have been made, and therefore a second constraint which affects to the direction of these significant lines and to the camera rotation can be considered. Thus, the normal to the projection plane of a line in one image must be perpendicular to the hypothesized 3D direction of the line in the other image. For the ideal case it can be expressed as

$$\mathbf{d}_{1j} \cdot \mathbf{R}\,\mathbf{n}_{2j} = 0\,; \;\; \mathbf{d}_{2j} \cdot \mathbf{R}^T \mathbf{n}_{1j} = 0 \tag{4}$$

where $\mathbf{d}_{1j}$ and $\mathbf{d}_{2j}$ are the direction in each image of the j-th line, obtained in §4.

It is clear that in the presence of noise there is not a set of motion parameters ($\mathbf{R}$, $\mathbf{t}$) that can exactly satisfy these constraints for all the features. So, we try to find a correction for the given observations in such a way that the points satisfy the epipolar constraint (3), and the significant lines satisfy the direction estimate (4). This correction is minimized taking into account the weighing of different errors. We formulate a constrained least-squares, that can be solved using Lagrangian multipliers [22]:

$$\begin{aligned} J_d = \;& \textstyle\sum_{i,j} \; \delta\mathbf{p}_{1i}^T \boldsymbol{\Gamma}^{-1}_{\delta\mathbf{p}_{1i}} \delta\mathbf{p}_{1i} + \delta\mathbf{p}_{2i}^T \boldsymbol{\Gamma}^{-1}_{\delta\mathbf{p}_{2i}} \delta\mathbf{p}_{2i} \\ + \;& \lambda_i (\mathbf{p}_{1i} + \delta\mathbf{p}_{1i}) \cdot (\mathbf{t} \times \mathbf{R}(\mathbf{p}_{2i} + \delta\mathbf{p}_{2i})) \\ + \;& \delta\mathbf{n}_{1j}^T \boldsymbol{\Gamma}^{-1}_{\delta\mathbf{n}_{1j}} \delta\mathbf{n}_{1j} + \lambda_{1j} \mathbf{d}_{2j} \cdot \mathbf{R}^T (\mathbf{n}_{1j} + \delta\mathbf{n}_{1j}) \\ + \;& \delta\mathbf{n}_{2j}^T \boldsymbol{\Gamma}^{-1}_{\delta\mathbf{n}_{2j}} \delta\mathbf{n}_{2j} + \lambda_{2j} \mathbf{d}_{1j} \cdot \mathbf{R} (\mathbf{n}_{2j} + \delta\mathbf{n}_{2j}) \end{aligned} \tag{5}$$

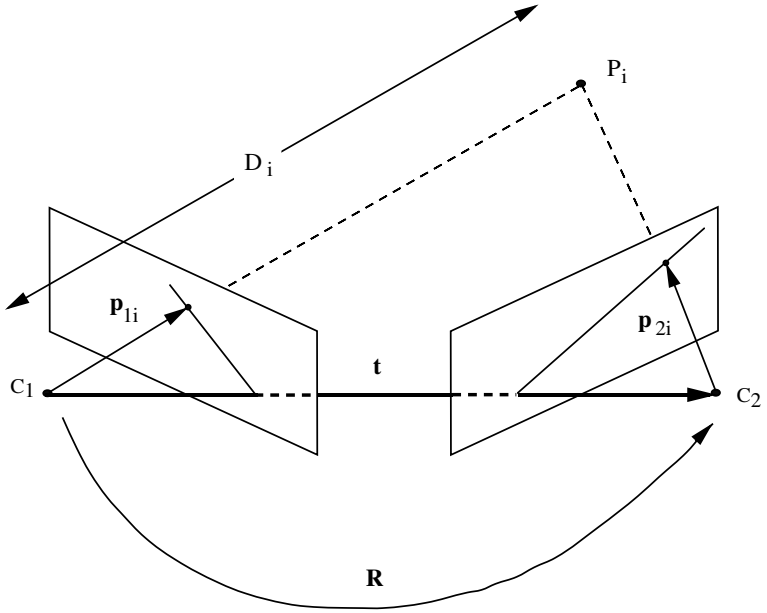

Figure 7: The translation of the camera and the lines of projection of two corresponding points must be coplanar to satisfy the epipolar constraint.

where $\mathbf{\Gamma}$ correspond with the covariance matrix of the observations uncertainty (Appendix).

Setting the derivatives of this expression equal to zero for the unknowns, and solving the set of equations to eliminate the local variables λ_i, λ_j, $\delta\mathbf{p}_{1i}$, $\delta\mathbf{p}_{2i}$, $\delta\mathbf{n}_{1j}$, $\delta\mathbf{n}_{2j}$, an equivalent expression, depending only on the motion parameters, can be obtained. There are nonlinearities and the results are too complicated to do anything useful with them. As in [23] the second order terms of the noise are eliminated and then the derivatives are taken. In this way, the proposed expression to minimize in function of the motion parameters is

$$\begin{aligned} J_d &= \sum_{i,j} \frac{(\mathbf{p}_{1i}\cdot(\mathbf{t}\times\mathbf{R}\mathbf{p}_{2i}))^2}{\mathbf{q}_{1i}^T\mathbf{\Gamma}_{\delta\mathbf{q}_{1i}}\mathbf{q}_{1i}+\mathbf{q}_{2i}^T\mathbf{\Gamma}_{\delta\mathbf{q}_{2i}}\mathbf{q}_{2i}} \\ &+ \frac{(\mathbf{d}_{2j}\cdot\mathbf{R}^T\mathbf{n}_{1j})^2}{(\mathbf{R}\mathbf{d}_{2j})^T\mathbf{\Gamma}_{\delta\mathbf{n}_{1j}}\mathbf{R}\mathbf{d}_{2j}} \\ &+ \frac{(\mathbf{d}_{1j}\cdot\mathbf{R}\mathbf{n}_{2j})^2}{(\mathbf{R}^T\mathbf{d}_{1j})^T\mathbf{\Gamma}_{\delta\mathbf{n}_{2j}}\mathbf{R}^T\mathbf{d}_{1j}} \end{aligned} \tag{6}$$

where

$$\mathbf{q}_{1i} = \mathbf{t}\times\mathbf{R}\,\mathbf{p}_{2i}\ ;\ \ \mathbf{q}_{2i} = \mathbf{R}^T(\mathbf{t}\times\mathbf{p}_{1i})$$

Iterated methods [22] allow to solve the motion with an scale factor for translations. We have used an estimation method such as Levenberg-Marquardt [24]. The proposed algorithm needs an initial guess of camera motion to solve the problem. We take the motion guess from odometry. We eliminate the scale factor considering the length advanced by the robot also from odometry. As it is known, odometry provides quite well the length advanced by the robot, but orientation and heading are not robust enough [4].

5.2 Structure computation

Once the camera motion is computed, the structure is easily obtained by triangulation, obtaining each 3D line as the intersection of its two projection planes. The 3D direction of each line can be obtained as

$$\frac{\mathbf{n}_{1j}\times\mathbf{R}\,\mathbf{n}_{2j}}{||\mathbf{n}_{1j}\times\mathbf{R}\,\mathbf{n}_{2j}||} \tag{7}$$

A 3D point (i) of the j-th line has been obtained as the intersection of the projection line of the point in the first image with the projection plane of the line in the second image. Thus the distance D_i from the first camera to the 3D point is evaluated (Fig. 7), and therefore the coordinates of the 3D point are

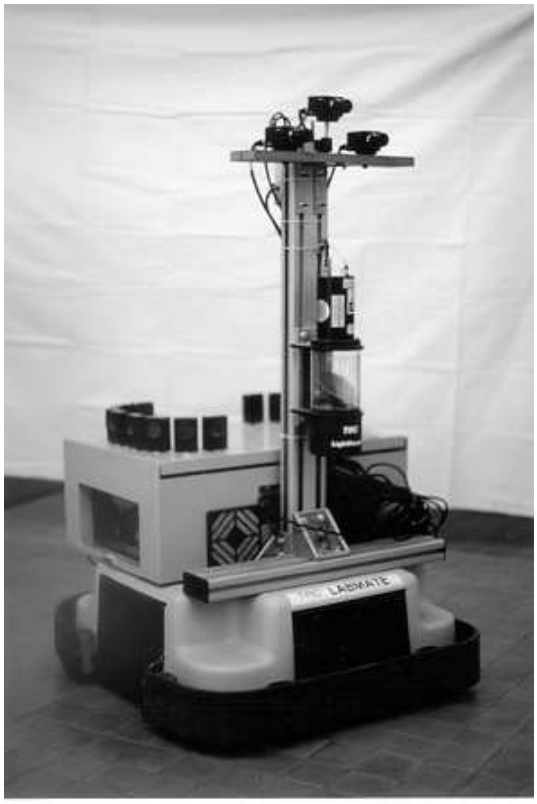

Figure 8: Mobile robot in its actual configuration

$$D_i \, \mathbf{p}_{1i} = \frac{\mathbf{t} \cdot \mathbf{R} \, \mathbf{n}_{2j}}{\mathbf{p}_{1i} \cdot \mathbf{R} \, \mathbf{n}_{2j}} \, \mathbf{p}_{1i} \tag{8}$$

However, bad results of structure are obtained when the projection plane of a line is nearly parallel to the translation vector, because the two projecting planes of the line are nearly parallel. In this case, it is better to obtain the structure using the points directly. The distance from the origin of the first frame to the 3D point can be evaluated, and therefore the coordinates of the 3D point are

$$D_i \, \mathbf{p}_{1i} = \frac{(\mathbf{t} \times \mathbf{R}\mathbf{p}_{2i}) \cdot (\mathbf{p}_{1i} \times \mathbf{R}\mathbf{p}_{2i})}{||\mathbf{p}_{1i} \times \mathbf{R}\mathbf{p}_{2i}||^2} \, \mathbf{p}_{1i} \tag{9}$$

6 Experiments

At the end of each section we have presented experimental results with several images showing the most relevant ideas involved in each step. The proposed method is aimed to help our mobile robot to navigate. We have mounted on a Labmate robot a ring of ultrasound sensors, a scanner laser, and three cameras with and acquisition and preprocessing board (Fig. 8). The computational resource is based on a SPARC computer which is also aboard and it is connected by ethernet via radio.

In this section we obtain a reconstruction of a real scene (Fig. 9) from lines with points and a translation motion. We show the reconstructed structure applying the algorithm proposed. The results of intermediate steps are avoided to emphasize the final result.

A good guess of the camera rotation is needed to make the motion algorithm to converge. It has been observed that the translation obtained is usually deviated towards the focal axis. The rotation computation is a very critical step, because a little error in rotation makes translation and structure to degenerate. When constraints about line orientation are not used, the algorithm has some problems to disambiguate rotations around the vertical direction from translations along the horizontal axis parallel to the image plane. Using the direction constraints with lines nearly parallel to the focal axis, the results improve. In Table 1 an example of the motion computed is given.

Figure 9: Scene of our laboratory used to compute motion and structure

	α	W_x	W_y	W_z
Commanded	-	0.0	0.0	0.0
With Direction Constraints	8.25	-8.86E-3	4.79E-1	-3.69E-1
Without Direction Constraints	31.38	9.79E-1	1.48E-1	-7.72E-1

Table 1: Deviation angle between computed and commanded translation (α), and rotation computed (W_x, W_y, W_z) in one experiment. In the first case the direction constraints have been used, in the second case they have not been considered. The values are expressed in degrees.

A reconstruction of the larger lines of the scene, projected in the first camera reference system, can be seen in Fig. 10. The top view of this reconstruction is shown in Fig. 11. In this case we have a good reconstruction. The lines in the wall are approximately in a plane and the big table is reconstructed nearly perpendicular to the wall. However, the two lines of the little table in front of the wall are difficult to reconstruct, because their planes of projection are nearly parallel to the camera translation. They must be reconstructed using two characteristic points. As the table is occluded by the stool, the depth of the points on that side is not accurate.

In the previous experiment a pure translational motion has been commanded. The method also works when there is rotation of the camera, but the translation must be enough large. When the translation is small the motion and depth results are not valid.

With respect to the computation time required by the different steps of the algorithm, the most critical steps are the extraction of the features in the images (which takes about 1 second per image) and the solution of the nonlinear least-squares problem. Due to the characteristics of this last problem, the computation time depends on its conditioning, initial guess and tolerances. The matching and other steps like reconstruction take short time (about 0.1 seconds), which also depend on the number of features.

As was mentioned above, we have solved for five motion parameters, three of them represent the camera rotation and two of them represent the direction of translation (the scale factor $|\mathbf{t}|$ is obtained from odometry). If we know the vertical direction very well, two degrees of freedom will be reduced using rectified images to make easier the convergence of the algorithm (the rotation ($\mathbf{R}$) will only be around the y axis).

7 Conclusions

We have presented a complete algorithm to obtain the camera motion and the scene structure using straight lines with points. It has been assumed that the lines are vertical and horizontal. This assumption is normally achieved in man made environments.

The motion computation algorithm uses lines and characteristic points on them. A noise model that considers the point to belong to the line has been proposed. To improve its robustness the hypothesized 3D direction of some significant lines are used. This leads to a partial correction of the coupling between the rotation and the translation in the computation of motion. When motion is

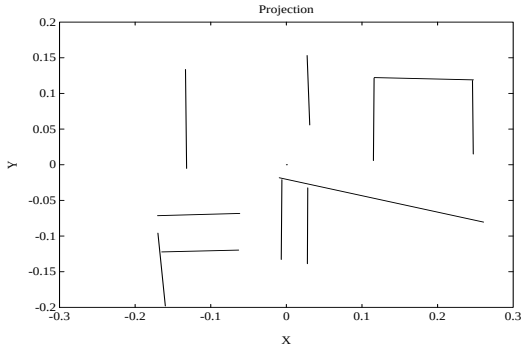

Figure 10: Reconstructed features projected in the first camera location

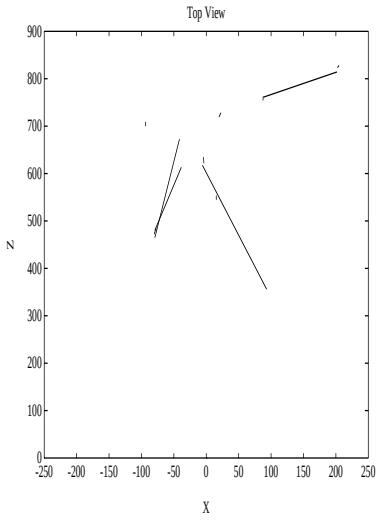

Figure 11: Reconstruction from the motion computed. Top view

computed the 3D localization of the features is obtained.

The experiments have shown the difficulties of the complete motion and structure paradigm in real situations, but partial interesting conclusions have been obtained. The extractor of straight lines works well. However sometimes the lines are broken and there exist some difficulties in regions with specular reflections. The cornerness operator combined with the extractor of lines allows to obtain points associated to them with a small computational cost. The correspondence problem has been solved by tracking. As it is based on a extended description of the feature, the results are good enough, specially when intermediate and close images are used. The rectification and vanishing point detection works correctly when a good vertical cue is available. The motion and structure computation needs a good rotation guess to converge, because a nonlinear minimization must be solved. When there are spurious matched features or the scene is all far away from the camera the results are not good. Nevertheless, it has been proved that the significant information allows to disambiguate the coupling between rotation and translation. Besides that, it has been showed that motion and structure can be automatically computed from at least two images of man made environments, and this information can be used for navigation of a mobile robot.

Appendix

Small errors are assumed and therefore we can consider the first order approximation to obtain the covarianze of the errors of the projecting vectors. Therefore, the covarianze matrix of the point and the line vectors can be expressed as

$$\mathbf{\Gamma}_{\delta \mathbf{p}_i} = \mathbf{J}^{\delta \mathbf{p}_i}_{\psi_i,\theta_l,\phi_l} \begin{bmatrix} \sigma^2_{\psi_i} & 0 & 0 \\ 0 & \sigma^2_{\theta_l} & 0 \\ 0 & 0 & \sigma^2_{\phi_l} \end{bmatrix} {\mathbf{J}^{\delta \mathbf{p}_i}_{\psi_i,\theta_l,\phi_l}}^T$$

$$\mathbf{\Gamma}_{\delta \mathbf{n}_j} = \mathbf{J}^{\delta \mathbf{n}_j}_{\theta_l,\phi_l} \begin{bmatrix} \sigma^2_{\theta_l} & 0 \\ 0 & \sigma^2_{\phi_l} \end{bmatrix} {\mathbf{J}^{\delta \mathbf{n}_j}_{\theta_l,\phi_l}}^T$$

where

$$\mathbf{J}^{\delta \mathbf{p}_i}_{\psi_i,\theta_l,\phi_l} = \begin{bmatrix} -o_x & n_x cos\psi_i & -a_y \\ -o_y & n_y cos\psi_i & a_x \\ -o_z & n_z cos\psi_i & 0 \end{bmatrix}$$

$$\mathbf{J}^{\delta \mathbf{n}_j}_{\theta_l,\phi_l} = \begin{bmatrix} -n_x tan\theta_l & -n_y \\ -n_y tan\theta_l & -n_x \\ -cos\theta_l & 0 \end{bmatrix}$$

being

$$\begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} = Rot(z,\phi_l) Rot(y,\theta_l) Rot(x,\psi_i)$$

Acknowledgments

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

References

[1] T.S. Huang and A. N. Netravali. Motion and structure from feature correspondences: A review. *Proceedings of the IEEE*, 82(2):252–268, 1994.

[2] Y. Liu and T.S.Huang. Estimation of rigid body motion using straight line correspondences. *Computer Vision, Graphics and Image Processing*, (43):37–52, 1988.

[3] M.E. Spetsakis and Y. Aloimonos. Structure from motion using line correspondences. *International Journal of Computer Vision*, (4):171–183, 1990.

[4] X. Lebégue and J.K. Aggarwal. Significant line segments for an indoor mobile robot. *IEEE Transactions on Robotics and Automation*, 9(6):801–815, 1993.

[5] J. Weng, T.S. Huang, and N. Ahuja. *Motion and Structure from Image Sequences*. Springer-Verlag, Berlin-Heidelberg, 1993.

[6] J.J. Guerrero, C. Sagüés, and A. Lecha. Motion and structure from straight edges with tip. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 2459–2464, San Antonio, USA, Oct 1994.

[7] J. Weng, N. Ahuja, and T.S. Huang. Optimal motion and structure estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(9):864–884, 1993.

[8] T. Viéville, E. Clergue, and P. Facao. Computation of ego-motion and structure from visual and inertial sensors using the vertical cue. In *Fourth International Conference on Computer Vision*, pages 591–598, Berlin, May. 1993.

[9] J.B. Burns, A.R. Hanson, and E.M. Riseman. Extracting straight lines. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(4):425–455, 1986.

[10] J. Cooper, S. Venkatesh, and L. Kitchen. Early jump-out corner detectors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(8):823–828, 1993.

[11] R. Deriche and G. Giraudon. A computational approach for corner and vertex detection. *International Journal of Computer Vision*, 10(2):101–124, 1993.

[12] D.H. Ballard and C.M. Brown. *Computer Vision*. Prentice Hall, Englewood Cliffs, N.J., 1982.

[13] L. Kitchen and A. Rosenfeld. Gray level corner detection. *Pattern Recognition Lett.*, 1:95–102, 1982.

[14] X. Xie, R. Sudhakar, and H. Zhuang. Corner detection by a cost minimization approach. *Pattern Recognition*, 26(8):1235–1243, 1993.

[15] T. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press In., 1988.

[16] R. Deriche and O. Faugeras. Tracking line segments. In *First European Conference on Computer Vision*, pages 259–268, Antibes, France, 1990.

[17] J.J. Guerrero and J.M. Martínez. Determination of corresponding segments by tracking both geometrical and brightness information. In *International Conference on Advanced Robotics*, pages 703–709, Barcelona, September 1995.

[18] J.J. Guerrero and C. Sagüés. Tracking features with camera maneuvering for vision-based navigation. *Journal of Robotic Systems*, 15(4):191–206, 1998.

[19] R.Y. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. In *Computer Vision and Pattern Recognition*, pages 364–374, 1986.

[20] C. Sagüés and J.J. Guerrero. Motion and structure from significant segments in man made environments. In *IFAC, Intelligent Autonomous Vehicles*, pages 337–342, Espoo, Finland, June 1995.

[21] O. Faugeras. *Three-Dimensional Computer Vision. A Geometric Viewpoint*. The MIT Press, Massachusetts, 1993.

[22] P.E. Gill, W. Murray, M.A. Saunders, and M.H. Wright. Constrained nonlinear programming. In G.L. Nemhauser, A.H.G. Rinnoy Kan, and M.J. Todd, editors, *Handbooks in Operations Research and Management Science. OPTIMIZATION*, volume 1, pages 171–210. Nort-Holland, Amsterdam, 1989.

[23] Minas E. Spetsakis. Models of statistical visual motion estimation. *CVGIP: Image Understanding*, 60(3):300–312, 1994.

[24] J.J. Moré. The levenberg-marquardt algorithm: Implementation and theory. In *Lecture Notes in Mathematics 630*. Springer, 1977.