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

Internal Report: 2001-V02

Uncalibrated vision based on lines for robot navigation[1]

Guerrero J.J., Sagüés C.

If you want to cite this report, please use the following reference instead:
Uncalibrated vision based on lines for robot navigation, Guerrero J.J., Sagüés C., *Mechatronics* , Vol. 11(6), pages 759-777, 2001.

[1]This work was partially supported by CICYT project TAP97-0992-C02-01 and DGA project P29/98.

Uncalibrated vision based on lines for robot navigation

Guerrero J.J. and Sagüés C. †

Abstract

In this paper, we study an uncalibrated monocular vision system based on lines for robot navigation in man-made environments. Neither environment a priori map is used nor euclidean reconstruction is obtained. The method is based on the extracted straight lines, supposing the robot moves on a plane ground. The effect in the image of camera rotation is computed from the homography of the line at infinity. From corresponding vertical lines in two uncalibrated images both, the robot heading and a region in the image, which corresponds to the free space ahead, are computed. This is the most relevant aspect of our method, because usually it has been considered that corresponding lines in two images do not constraint the structure and motion problem. However, using the visibility constraint and compensating the camera rotation, we have obtained the direction of translation and a qualitative free space ahead. A control scheme in image coordinates has been implemented to carry out the experiments with our robot moving in the lab. It allows the robot to advance along a corridor avoiding doors or pillars without collision. The computation of information from lines is accurate enough and although the robot has not previous map, it is autonomous when moving in this environment. The method supposes advantages in comparison with other works because neither image nor camera-wheel calibration are required.

Keywords: Uncalibrated vision, straight lines, motion and structure, vision based navigation, mobile robots.

1 Introduction

Mobile autonomous robots are actually able to execute tasks using specific landmarks or other systems which give global localization. However, without these systems, powerful perception systems are required to allow them to navigate. Vision is a sensor broadly used.

Many works on vision for robot navigation try to build accurate models of the scene, using accurately calibrated systems [2, 23]. Although these systems operate properly are computationally expensive and are conditioned by the fitting of the real scene to

†Dpto. 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 34-976-761861, email {jguerrer, csagues}@posta.unizar.es — Published in MECHATRONICS 11(2001) pp. 759-777

the model. Besides that, the calibration process is difficult, being the final results sensitive to calibration errors. Anyway, the problem of calibration for navigation is not just one of vision, since the entire eye-wheel system must be calibrated [22].

On the other hand, it has been shown that it is possible to recover information from uncalibrated cameras, based in their projective properties [14]. Many applications on non-metric vision has been proposed and tested [19]. By recovering non-metric visual data, the vehicle must be controlled in a non-metric space. As suggested in [4] there are two natural approaches: the first is to use visual servoing and the second is to compute a trajectory using non-metric reconstruction to carry out metric control at the final stage.

We use a monocular vision system without previous calibration, carrying out the robot control directly in image coordinates. Here, we do not assume that the intrinsic parameters of the camera are known (pixel aspect ratio, focal lengths, coordinates of the principal point) and we do not compute the extrinsic camera calibration with respect to the mobile robot. We consider that the camera looks approximately towards the direction of advance of the robot, and we only suppose that the image plane is approximately vertical. Anyway a vertical rectification of the image can circumvent this assumption [20].

Our approach takes straight lines in the image as key features, because in man-made environments most of the object boundaries are straight lines. The straight line has a simple mathematical representation, involve more information, and has many advantages as opposed to features like points [13]. In our work the lines are classified as vertical and non-vertical doing simple assumptions. From non-vertical lines, vanishing points are computed to correct camera orientation and to compute the homography of the infinity between images [9]. The vertical lines are matched and used as environment features.

As neither previous map nor robot heading are known, we face up to a structure and motion problem. The computation of the direction of translation and the time to collision based on the focus of expansion (FOE) is a classical result [5], [16]. In these approaches the basic visual information is usually the correspondence of points or the optical flow, and accurately calibrated cameras are considered. We compute both the focus of expansion and a region in the image, which corresponds to the free space in front of the robot, using corresponding vertical lines in two uncalibrated images. This is the most relevant aspect of our paper, because usually it has been considered that corresponding lines in two images do not constraint the structure and motion problem [11]. However, using the visibility constraint and compensating the camera rotation as proposed here, both the direction of translation and a qualitative free space ahead have been obtained.

The navigation along a straight corridor has been used as a case study to test our proposal. Odometry provides quite well the length advanced, but orientation and heading are not robust enough [12]. A control scheme in image coordinates allows to correct the robot heading avoiding collision. Each step, only two images are used to compute both the direction of translation and the depth of the obstacles, and no map of the environment is recorded. The commanded heading is obtained from the vanishing point (assuming the robot must advance along the dominant direction) or through the computed free space when the non-metric depth in front of the robot is small (it is near a wall, there is an obstacle like a pillar, etc). The method could

be applied in man-made environments, where straight lines and a dominant direction can be extracted from the images, for example corridors or streets. The experimental results have shown that the vanishing point is a key cue to navigate along a corridor. Besides that, it is possible to avoid collision computing the FOE and the free space from corresponding lines in two uncalibrated images. However, the computation of the FOE in real scenes is a nontrivial problem [18]. Thus, the extracted FOE has been filtered along time, assuming a smooth variation, because its computation from two images did not result accurate enough.

A summary of the paper follows. In Section 2 the extraction and matching of straight lines is shortly presented. Section 3 describes the method to compute the direction of translation, which includes the compensation of camera rotation. Section 4 explains how to compute the steps to collision to the objects in front of the robot. In Section 5 the proposal is particularised for navigation along a corridor, presenting the control scheme based on vision. Section 6 describes experimental results obtained with our mobile robot. Finally, conclusions are presented in Section 7.

2 Extraction and matching of lines

As was mentioned above, our approach takes straight lines in the image as key features. They are extracted with our version of the method proposed by Burns [6]. This extractor provides not only the geometric representation of the projected line, but also some attributes related with the brightness of the edge (contrast, average grey level, steepness).

Once straight edges are extracted in the image, they are classified as vertical and non-vertical. To do that we assume the image plane is approximately vertical, and therefore vertical lines appear parallel and vertical in the image. As mentioned, a vertical rectification of the image could be carried out to relax this assumption [20].

After that, the vertical lines in two images are matched. We determine correspondences between linear segments, without assumptions about the motion or the structure of the scene. 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 [3]. A constant velocity model has been heuristically selected to predict the features location in the following image. A nearest neighbour tracking approach as in [7] has been developed [10], in which two image brightness attributes of the line are also used in the tracking process. These attributes are the average grey level and the mean contrast.

In the proposed tracker [10], the covariance of the innovation is a block diagonal matrix where we can differentiate 3 sub-blocks, corresponding to the location parameters, the length and the brightness attributes respectively. The proposed matching technique can be seen as a variation of the nearest neighbour standard filter using the localization parameters, in which some measurements are excluded from the acceptance region if there are no length or brightness coherence. With this procedure, compensation between high precision in some parameters with high error in others is avoided. As was concluded [10], the brightness attributes are useful 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.

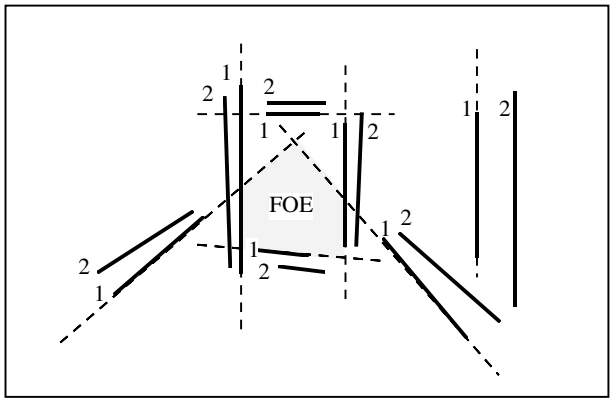


Figure 1: Vertical and non vertical corresponding lines in two images are shown (1 and 2 indicate the image number). Assuming pure translation and considering the visibility constraint, the direction of translation can be computed from lines of unknown depth. Every line constraints the FOE to belong to a half-plane. The intersection of all these half-planes provides a solution region.

3 Robot heading from lines with an uncalibrated camera

As we use lines in two images, the general problem of structure and motion cannot be solved [11]. A line in each image provides two parameters, but four parameters are also needed to determine its 3D position. The same happens with a vertical image plane and using only vertical lines moving on a horizontal plane. Each line provides one parameter in each image (its x coordinate), but two parameters are needed to locate the vertical line on the ground plane. Therefore, there are not enough information to compute motion and structure from lines in two images. However, in this reasoning the visibility constraint has not been considered.

3.1 Computation of the robot heading from lines

Using the visibility constraint (the scene observed is always in front of the camera), and assuming pure translational motion or general rigid motion with a bounded rotation it is possible to solve, at least qualitatively, the structure and motion problem from normal flow information [1], [21]. This idea can be extended to corresponding lines in two uncalibrated images.

When the relative motion between the camera and the scene is a pure translation, the projected motion field has one singular point which is a focus of expansion or contraction. The image trajectories of every point are linear and they intersect at that point, which corresponds with the projection in the image of the direction of translation.

The focus of expansion (FOE) is obtained from corresponding lines in two images using a voting scheme that exploits the visibility constraint. As the observed lines must be in front of the camera, and the robot makes a forward motion, all the image velocities must expand from the FOE. When a line moves from the left to the right, every point belonging to the half plane on the left of the line can be considered as a

possible FOE. On the other hand, the FOE will be on the right of a line when this line moves to the left.

A voting scheme based on the Hough transform is used to obtain it. Thus, each line votes to all the image points compatible with the sign of its projected motion. After the voting process, the image region having a higher number of votes is selected (Fig. 1). This method provides an image region as FOE, whose size depends on the number and distribution of the lines used. It is interesting to have as many matched lines as possible in order to obtain a narrow solution region.

In this work only vertical lines and planar motion is assumed, which simplifies the computation. Therefore, the accumulator of votes has one dimension and only the x coordinate of the FOE is computed. Naming x_1, x_2 the location of corresponding vertical lines in first and second images respectively, the algorithm can be outlined as follows:

begin FOEComputation
    for all matched lines
        if $x_2 > x_1$ vote to all $x_{ac} < x_1$
        if $x_2 < x_1$ vote to all $x_{ac} > x_1$
    end for
    x^{FOE} = the most voted in x_{ac}
end

where x_{ac} is the accumulator array.

3.2 Rotation compensation

The method to compute the robot heading just presented assumes that the camera does not rotate. Thus, the locations of the lines in the first image must be corrected to compensate the rotation, before applying the Hough transform. As the vanishing point (VP) corresponds to a point at the infinite, its location in the image changes only due to rotations but not due to translations. In a first approach, the effect of the relative camera rotation can be compensated from vanishing points obtained in both images. An approximate compensation of the rotation can be obtained from the difference between the x coordinates of the vanishing point in the first and second images $(x_1^{VP} - x_2^{VP})$. Therefore $x_1^{corrected} = x_1 - (x_1^{VP} - x_2^{VP})$. This approximation is more valid near the VP.

A better rotation compensation can be done obtaining the homography of the line at infinity [14]. In this way, we consider the general situation of two images taken with a projective camera, even being uncalibrated.

Let $\mathbf{H}$ be the homography matrix relating the coordinates in both images of vertical lines at the infinity

$$\begin{pmatrix} \lambda x_2 \\ \lambda \end{pmatrix} = \mathbf{H} \begin{pmatrix} x_1 \\ 1 \end{pmatrix} \tag{1}$$

In our case, $\mathbf{H}$ is a 2x2 matrix defined with a scale factor. The overall scale factor is arbitrary, so $h_{22} = 1$ is chosen. Thus, we have

$$\begin{aligned} \lambda\, x_2 &= h_{11}\, x_1 + h_{12} \\ \lambda &= h_{21}\, x_1 + 1 \end{aligned}$$

Eliminating λ, each corresponding vertical line provides a linear equation to obtain the three independent elements of the homography matrix $\mathbf{H}$, that is

$$x_1\, h_{11} - x_1\, x_2\, h_{21} + h_{12} = x_2 \tag{2}$$

Using the vanishing point in both images and at least two corresponding vertical lines at the infinity, the homography matrix can be computed solving a linear set of equations like (2). To compute it, we have heuristically selected as corresponding lines (supposed to be at the infinity) those having a disparity similar to the disparity of the vanishing point ($x_1^{VP} - x_2^{VP}$).

Once matrix $\mathbf{H}$ computed, the lines in the first image are corrected in the following way

$$x_1^{corrected} = \frac{h_{11}\, x_1 + h_{12}}{h_{21}\, x_1 + 1}$$

This expression can be rearranged to obtain the rotation compensation x^{rc} as

$$x_1^{corrected} = x_1 + x^{rc} = x_1 + \frac{h_{11}\, x_1 + h_{12} - h_{21}\, x_1^2 - x_1}{h_{21}\, x_1 + 1} \tag{3}$$

3.3 Filtering the FOE

As it is known, determining the FOE in real scenes is not a trivial problem [18]. The algorithm to compute the FOE described in section 3.1 only performs when it is inside the field of view. This is the normal situation in a mobile robot, where the heading of the robot and the focal axis of the camera are nearly parallel. As was mentioned, the previous algorithm obtains an image region as solution, whose size depends on the number, distribution and depth of the lines used.

However, in order to compute it more accurately, two considerations have been established:

- Each FOE region computed is characterized as a probabilistic measure, whose mean value is the center of the region solution, and whose variance is obtained in function of its width ($width^{FOE}$). Assuming a 95% of probability of the real FOE being into the solution region, and considering a Gaussian distribution for the measure, we have $0.5\, width^{FOE} = 1.96\, \sigma_{FOE}$. Therefore the variance is obtained as

$$\sigma_{FOE}^2 = \left(\frac{width^{FOE}}{3.92} \right)^2 \tag{4}$$

- We propose to estimate the FOE in each step, based on the measurements of previous steps. The camera is rigidly attached to the mobile robot, which moves slowly doing only slight changes of heading. Thus, it can be assumed that the FOE position is stable in the image. It can change, but changes are smooth in time. Thus, we propose to estimate it along the sequence of images, using a Kalman filter [3]. Heuristically a constant velocity model for the evolution of the current heading has been considered.

It is well known that the Kalman filter equations are computed directly from the linear space state model of a stochastic system, that is

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{F}\mathbf{x}(k) + \mathbf{v}(k) && (5) \\ \mathbf{z}(k) &= \mathbf{H}(k)\mathbf{x}(k) + \mathbf{w}(k) && (6) \\ \mathbf{E}\left[\mathbf{v}(k)\mathbf{v}(j)'\right] &= \mathbf{Q}(k)\delta_{kj} && (7) \\ \mathbf{E}\left[\mathbf{w}(k)\mathbf{w}(j)'\right] &= \mathbf{R}(k)\delta_{kj} && (8) \end{aligned}$$

Equations (5) and (7) represent the evolution of the FOE throughout time. Different models can be adjusted to its evolution. Heuristically we have selected a constant velocity model (that is a reasonable assumption due to the dynamics of the mobile robot), in which the state vector includes the position and the velocity. In this way, the state is,

$$\mathbf{x} = \left(x^{FOE}, \dot{x}^{FOE}\right)^T$$

We define the dynamic matrix as

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

However, the constant velocity can be considered valid only locally. There are some variations which can be modelled with the assumption of the current heading having a constant acceleration in each sample time. If we model the acceleration as a white noise without mean and q covariance, then process noise covariance is [3]

$$\mathbf{Q} = \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix} q$$

where q must be tuned experimentally.

Equations (6) and (8) model the measurement of the FOE given by the algorithm presented above (section 3.1). The matrix $\mathbf{H} = (1\ 0)$ selects the values corresponding to the position because the derivatives are not measured. The variance of the measurements is

$$\mathbf{R}(k) = \sigma^2_{FOE}$$

which is obtained as proposed (4).

Once defined the model, the Kalman filter equations to estimate the heading of the robot, based on current and previous measurements, are well known [3].

4 Computation of steps to collision

Once the robot heading has been computed, a relative depth map can be obtained assuming only the epipolar geometry. The computation is equivalent to the time to collision computation obtained from optical flow measures or corresponding points in calibrated cameras [5], [16]. However, in our proposal it is obtained from corresponding vertical lines in two uncalibrated images.

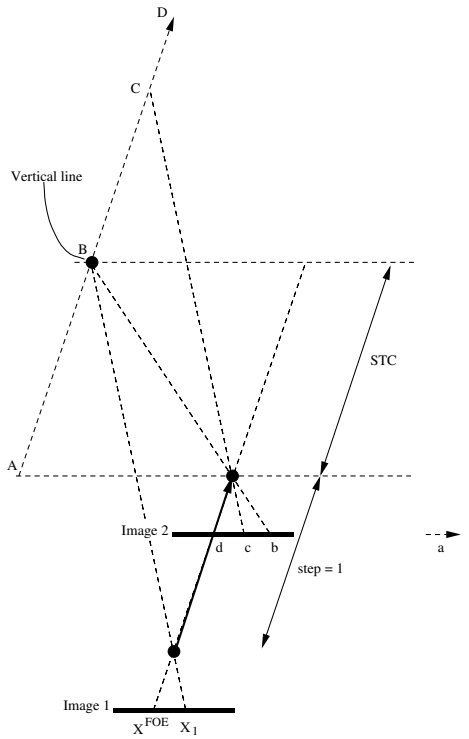


Figure 2: Geometry involved in the problem assuming pure translation and vertical images. Top view of the camera motion and scene, where the images are lines and the vertical edges are points.

Assuming that the robot advances with constant velocity, the relative depth can be obtained in steps or time to collision units. The formulation to obtain it can be achieved based on a projective invariant. The geometry involved in this problem, assuming vertical image planes and vertical extracted lines can be seen in Fig. 2. It is known that the cross-ratio of four points on a line is preserved under projective transformations [17]. Thus from four collinear points (a, b, c, d) we can define a projective invariant Cr that is

$$Cr(a, b, c, d) = \frac{(x^c - x^a)(x^d - x^b)}{(x^c - x^b)(x^d - x^a)}$$

where $(x^c - x^a)$ is the distance between points c and a.

To derive the expression that provides the steps to collision of a observed vertical line, we define four aligned points in the second image. They are:

- the point at infinity (a),
- the projected location of the vertical line in the second image (b),
- the projected location of the vertical line in the first image (c),
- the FOE location (d) .

If we name x_1 the location of the line in the first image and x_2 its location in the second one, it can be observed that $x^b = x_2$ and $x^c = x_1$. Therefore the cross-ratio of these four points turns out to be

$$Cr(a,b,c,d) = \frac{\infty\,(x^{FOE} - x_2)}{(x_1 - x_2)\,\infty} = \frac{x^{FOE} - x_2}{x_1 - x_2} \tag{9}$$

On the other hand, these four points can be considered the projection of four collinear points on the scene (points A, B, C, D in Fig. 2). They are on a line parallel to the translation, which passes through the observed vertical edge (point B). Their cross-ratio is

$$Cr(A,B,C,D) = \frac{(1 + STC)\,\infty}{1\,\infty} = \frac{1 + STC}{1} \tag{10}$$

where the distance advanced by the robot between two images is considered the unit, and STC is the number of steps to collision of the line.

Therefore, the cross-ratio of these four points and the cross-ratio of their perspective projection must be equal. Equating (9) and (10) we arrive at

$$STC = \frac{x_1 - x^{FOE}}{x_2 - x_1} \tag{11}$$

which allows to obtain the steps to collision to a vertical line knowing the FOE and the locations of the line in two images.

This computation is derived considering pure translation, and therefore, the effect of the camera rotation must be previously compensated as explained in section 3.2.

4.1 Errors in STC computation

As was said above, the depth of some lines in the scene can be computed in steps to collision units, using expression (11). However, there are many sources of error that can cause this value to be useless. Here, we analyze the effect of possible error sources in the computed STC.

Naming x_1^o, x_2^o the coordinates of a corresponding line observed in both images, and naming x^{rc} the effect of the rotation compensation,

$$x^{rc} = \frac{h_{11}\,x_1^o + h_{12} - h_{21}\,(x_1^o)^2 - x_1^o}{h_{21}\,x_1^o + 1}$$

the expression (11) turns out

$$STC = \frac{x_1^o + x^{rc} - x^{FOE}}{x_2^o - x_1^o - x^{rc}} \tag{12}$$

Three main sources of error can be considered: error in the computation of the FOE (x^{FOE}), error in matching ($x_2^o - x_1^o$), and error in the rotation compensation (x^{rc}). To analyze the influence of these sources of error in the computed depth (STC), we make the partial derivatives,

$$\begin{aligned} \frac{\delta\,STC}{\delta\,x^{FOE}} &= \frac{-1}{ddt} \\ \frac{\delta\,STC}{\delta\,(x_2^o - x_1^o)} &= \frac{-STC}{ddt} \\ \frac{\delta\,STC}{\delta\,x^{rc}} &= \frac{1 + STC}{ddt} \end{aligned} \tag{13}$$

where $ddt = x_2^o - x_1^o - x^{rc}$ is the disparity due to translation.

From these expressions it can be seen that the ddt must be large to have small influence of the three errors, in the computation of the depth. The problem appears near the FOE, where the ddt is very small, and therefore all possible errors have high influence. Thus, in practice the computed STC of lines near the FOE is not useful.

Large ddt makes difficult the matching step. From the expression of ddt, it can be deduced that if x^{rc} was large and $x_2^o - x_1^o$ was small, then matching would be easy and ddt would be large. However, there is no a rotation that allows compensating disparity in all the image when the camera translates in a direction close to the focal axis. Therefore, there is a trade-off between matching difficulty and depth accuracy. To circumvent it, we track the lines in intermediate images .

From the three error sources considered above, we have observed experimentally that the error in x^{FOE} is the most severe. Fortunately, the expressions (14) indicate that the influence of errors due to FOE are smaller than the influence of other errors, because the observed STC are usually larger than one. The influence of errors in x^{rc} turns out to be the largest. Fortunately, when the environment has a dominant direction x^{rc} is accurately computed because it is based on vanishing points.

5 Case study. Navigating along a corridor

As commented, our proposal could be applied in man-made environments where a dominant direction can be extracted from the images. In this section, we present as case study the concepts previously proposed, for the navigation of a robot along a corridor. Our system provides a local correction of the heading, assuming the robot is going along a straight corridor. Nevertheless, a global navigation system could provide a desired heading in a more general situation.

5.1 Dominant scene direction

Desired heading depends on the goal to be reached by the robot. Normally, it will be imposed by the user's task, or provided by a global navigation system. However, if the robot moves in a corridor or a room with a well defined main direction, the desired heading can be obtained from the main scene direction assuming that the robot must advance along it. This can be easily computed from a vanishing point using non-vertical lines.

With the non-vertical lines, two vanishing points may appear. One is the corresponding to the main direction of the environment and therefore corresponds to lines parallel to the desired heading. The other appears out of the field of view, and therefore its computation is ill-conditioned. We have used the first one. When the scene depth in front of the camera is high, the commanded heading is computed from this vanishing point.

Many works to extract vanishing points have been proposed [15]. We have obtained the vanishing point from non-vertical lines using the Hough transform. To do it, an accumulator array of one dimension is considered. Each non-vertical line votes to the x position corresponding to the intersection with the line of the horizon. The element of the accumulator where more intersections are found gives the vanishing point.

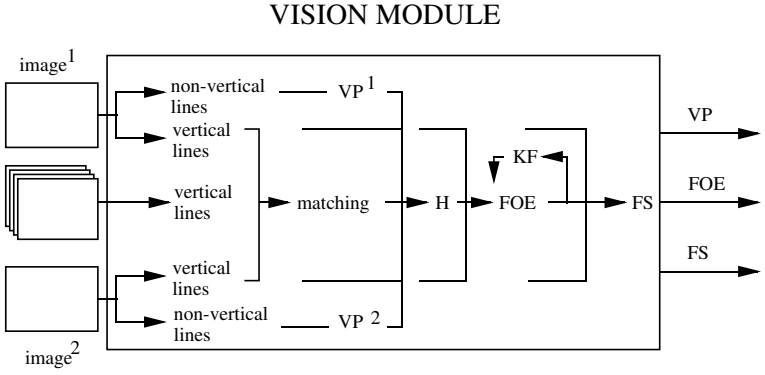


Figure 3: Outline of the vision module.

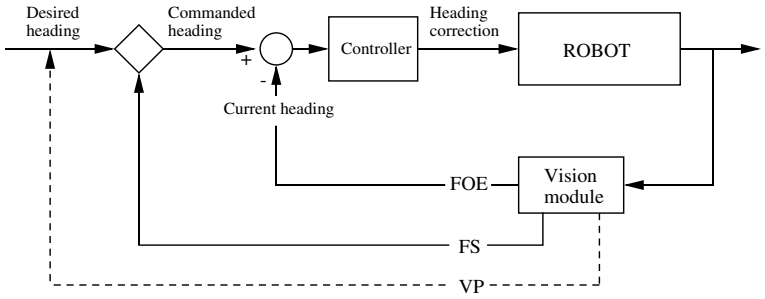


Figure 4: Outline of the heading control scheme based on vision.

5.2 Image based control scheme

Our proposal to compute 3D information from vision is outlined in the vision module of Fig. 3. Firstly, straight lines are extracted and classified as vertical and non-vertical. From the non-vertical lines a vanishing point (VP) is obtained in each image. On the other hand, the vertical lines are tracked and matched. After that a rotation compensation is carried out using vanishing points and other vertical lines to compute the homography (H) of the line at infinity (section 3.2). The vertical lines, after rotation compensation, are used to compute the FOE of the last step, which is filtered with a Kalman filter (KF), as was explained. Once the FOE has been obtained, the relative depth of vertical lines is evaluated in steps to collision units (STC obtained in section 4).

This depth information is used to determine an image region, which corresponds with the free space (FS) in front of the robot. It is delimited in the image by two vertical lines (IZQ on the left, DER on the right). To obtain them, we firstly apply a threshold to the STC of the lines to distinguish the near lines from the far ones. We assume that only three regions can appear (one region of free space and two occupied regions with close lines). To eliminate outliers, we apply a morphological closing, which consists on a dilation followed by an erosion using the same mask [8]. In the corridor this simple solution usually provides reasonable results.

The outputs of the vision module are: the vanishing point which is the desired heading, the FOE which corresponds with the current heading, and an image region of free space in front of the robot. All of these measurements are x positions along the horizontal direction, and they are expressed in pixel coordinates in the last image.

Figure 5: Estimating the vanishing point and the current heading. The extracted non-vertical lines are enlarged to the line of the horizon (large horizontal line in the image), showing the vanishing point. The large vertical line indicates the x estimate of the current heading.

To control the robot heading without camera calibration, the ideas previously presented are combined into a control scheme expressed in image coordinates. The robot is supposed to move forward, making correction of heading step by step. The goal is to advance along the corridor avoiding collisions.

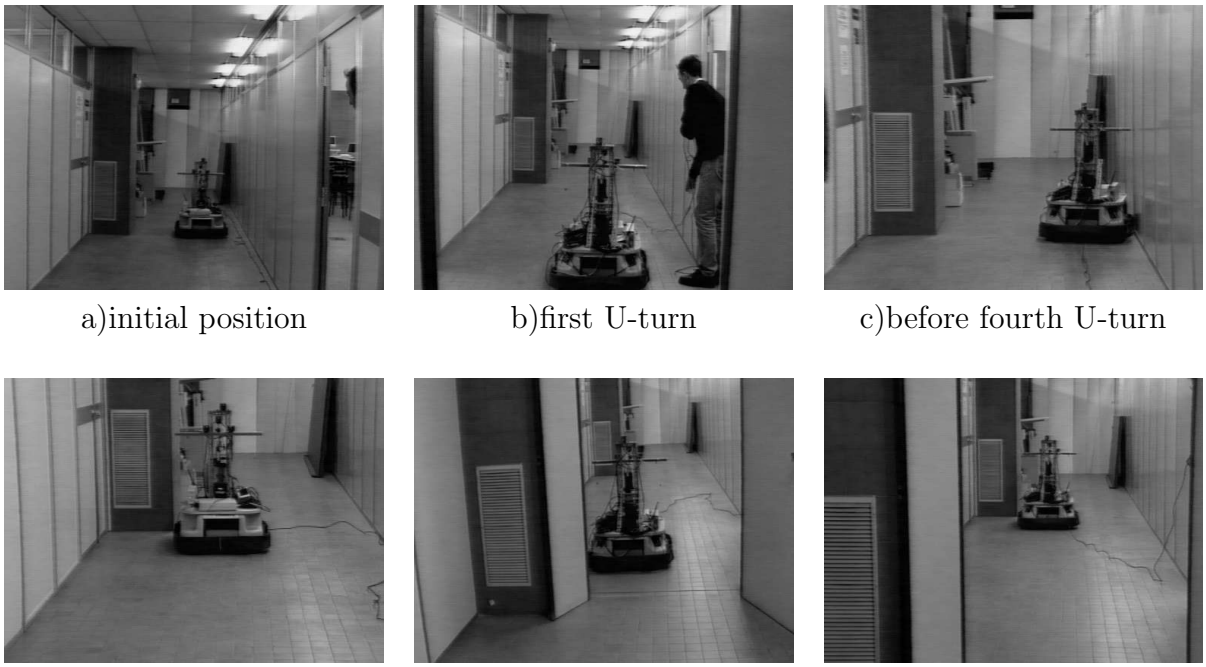
The image based control scheme is outlined in Fig. 4. The commanded heading is obtained from the desired heading (in the corridor VP) when there is enough free space in front of the robot. When the desired heading is not into the free space, the commanded heading is the center of the free space. A controller allows to correct the heading in function of the difference between the commanded heading and the current heading (measured both in the image). Experimentally, a proportional controller has been tuned.

6 Experimental results

Some experiments have been carried out with our mobile robot. We have mounted on a Labmate platform a ring of ultrasound sensors, a scanner laser, and three cameras with an acquisition and pre-processing board. The computational resource is also aboard. The method proposed in this paper uses an only camera, with its image plane approximately vertical. We have moved the robot on a corridor of our lab.

As an example, the extraction of main scene direction from lines in an image can be seen in Fig. 5. The computation of the vanishing point turns out accurate enough to control the robot. It has a maximum error of a few pixels which supposes less than a degree of error in orientation with respect to the environment.

In a first experiment the robot is commanded to go and return along the corridor making U-turns (rotations of 180 degrees). Fig. 6 shows the results obtained in our lab. Using only odometry the robot collides with the wall after a small number of U-turns, because of odometry drifts. Using the VP as commanded heading the robot turns round several times, maintaining its orientation in the corridor. We can conclude

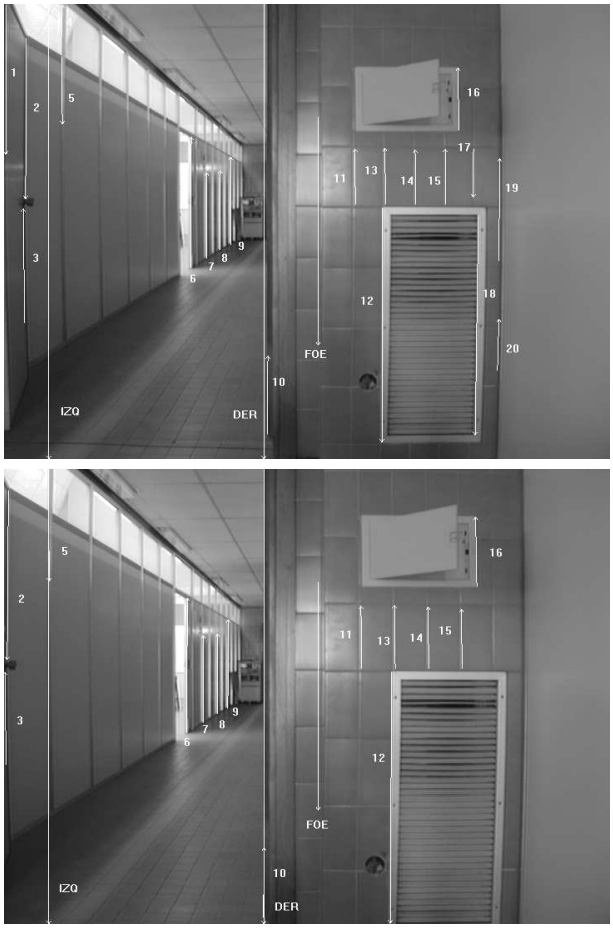


a)initial position b)first U-turn c)before fourth U-turn

d)initial position e)first U-turn f)before eighth U-turn

Figure 6: When the robot turns round some times using only odometry it loses the orientation with respect to the corridor, and it collides with the wall (images a,b,c). In this case, the robot collides with the wall before doing the fourth U-turn (rotation of 180 degrees). Using the vanishing point as commanded heading the robot turns round several times without collision (images d,e,f). In this example, after eight U-turns the robot is nearly at the initial position.

Figure 7: An example of the trajectory of the robot navigating along the corridor and avoiding a pillar. The method turns out robust to people walking in front of the robot, because it is based on straight lines in prominent directions.



Line	STC
2	6.44
3	6.38
5	7.84
6	12.88
7	13.16
8	13.40
9	13.64
10	5.50
11	2.48
12	2.84
13	2.78
14	3.00
15	3.14
16	3.15

Figure 8: From corresponding lines in two images, the FOE (marked as a line) and the free space are computed. The free space corresponds with the region between the IZQ and DER large lines. Here only the lines used to obtain the free space are printed. To obtain the FOE more corresponding lines have been used. The depth of the lines in steps to collision (STC) are given in the table.

that the VP is a key cue to navigate along a corridor.

The robot can also correct its lateral position to cross a door or to avoid a pillar in the corridor using the proposed approach. In Fig. 7 an example of the trajectory of the robot can be seen. In this case, the robot advances autonomously on the corridor, making heading corrections to cross the door without collision. A heading correction is achieved each step, based on the difference between the current heading and the commanded heading (Fig. 4). Initially the robot advances following the vanishing point. After that, the robot moves toward the free space. Once the pillar has been avoided and the robot crosses the door, it advances again in the direction of the corridor.

Two characteristic examples have been selected to emphasize how the complete algorithm works and how the robot avoids the pillar when it is necessary. In Fig. 8 we show the first example of the FOE and the FS computed from corresponding vertical lines. Supported in the images, it can easily tested in a qualitative way that the STC are satisfactory for all the lines. In this case, the commanded heading is obtained from the center of the FS (between IZQ and DER lines) to avoid the pillar, because the FS does not include the desired heading.

Line	STC
1	4.58
2	4.84
3	4.32
5	4.58
6	13.41
7	14.54
9	3.64
10	22.64
11	21.12
12	8.54
13	7.76
14	7.04
15	6.86
16	5.42

Figure 9: From corresponding lines in two images, the FOE (marked as a line) and the scene depth are computed. The depth, in steps to collision units, is given in the table. In this case only the line number 9 has a bad STC due to mismatching. The morphological closing filters it, obtaining the free space (region between the IZQ and DER large lines).

The result of FOE and STC computed from infinite lines of second example can be seen in Fig. 9. In this case only a line has a bad STC due to mismatching. The morphological closing gives a valid FS. In this case, there is enough free space in front of the robot to take the VP as commanded heading.

In both examples the length of the step was 40 cm approximately. As was analyzed in section 4.1, the computation of depth of the lines near the FOE is not too accurate, because the triangulation is ill conditioned (the baseline and the projection lines are nearly parallel). Thus, the lines near the FOE are excluded to compute the FS.

The most difficult step in our approach is the computation of the FOE from lines. To have valid measurements, many lines must be extracted and matched. Experimentally, we have observed that about 30 lines are enough. Sometimes there are no enough matched lines, and the FOE is not accurate in spite of the filter implemented. At the moment in 10% of the times the robot collides when it is not centered in the corridor or there are some obstacles (door, pillar) in front of it, being not qualified to avoid them in all situations. Anyway, the results allow to confirm that it is possible to compute both the robot heading and the depth of the scene from vertical lines in two uncalibrated

images to make the robot to navigate.

7 Conclusion

In this paper, we have presented an uncalibrated monocular vision system based on lines for robot navigation. A control scheme in image coordinates computes directly corrections to the heading of the robot, without previous map of the environment. Using the epipolar and visibility constraints, it is possible to compute both the current heading and a depth map in steps to collision units. We have computed this information from corresponding infinite lines in two uncalibrated images, which can be considered the most relevant aspect of our method.

The experiments have shown that the method works in man-made environments. We have carried out them on a corridor where the desired heading is automatically computed. These have confirmed that the VP is a key cue in this kind of situations. The computation of information from lines is good enough and although the robot has not previous map, it can be autonomous moving here. The experiments have also shown that the robot avoids doors or pillars without collision, based on the information computed from corresponding lines in two uncalibrated images.

The most critical step is the computation of the focus of expansion. To have valid measurements many lines must be extracted and matched, which may be difficult in some scenes. Another drawback of our method is that it is difficult to detect obstacles projected near the FOE. However, this is a general limitation of systems based on monocular vision. Anyway, our method supposes advantage in comparison with other works because neither image nor camera-wheel calibration are required. Actually we are working to obtain corrections for the robot heading, when there is no a dominant direction. This will allow extending our proposal to more general situations, because the rotation compensation could be made without vanishing points.

Acknowledgements

This work was partially supported by CICYT project TAP97-0992-C02-01 and DGA project P29/98.

References

[1] Y. Aloimonos and Z. Duric. Estimating the heading direction using normal flow. *Int. Journal of Computer Vision*, 13(1):33–56, 1994.

[2] N. Ayache. *Artificial Vision for Mobile Vision. Stereo Vision and Multisensory Perception.* MIT Press, London, 1991.

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

[4] P.A. Beardsley, I.D. Reid, A. Zisserman, and D.W. Murray. Active visual navigation using non-metric structure. Repport of research OUEL 2047/94, University of Oxford, Oxford OX1 3PJ, U.K., 1994.

[5] W. Burger and B. Bhanu. *Qualitative Motion Understanding*. Kluwer Academic, Boston, 1992.

[6] 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.

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

[8] R.C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, Reading, Mass., 1993.

[9] J.J. Guerrero and C. Sagüés. Navigation from uncalibrated monocular vision. In *IFAC, Intelligent Autonomous Vehicles*, pages 210–215, Madrid, March 1998.

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

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

[12] 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.

[13] 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.

[14] Q.T. Luong and O. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17(1):43–76, 1996.

[15] E. Lutton, H. Maitre, and J. Lopez-Krahe. Contribution to the determination of vanishing points using hough transform. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(4):430–438, 1994.

[16] E. De Micheli, V. Torre, and S. Uras. The accuracy of the computation of optical flow and of the recovery of motion parameters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(5):434–447, 1993.

[17] J.L.S. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision*. MIT Press, Boston, 1992.

[18] D. Nair and J.K. Aggarwal. Dynamic multi-sensor data fusion systems for intelligent robots. *IEEE Transactions on Robotics and Automation*, 14(3):404–416, 1998.

[19] L. Robert, C. Zeller, O. Faugeras, and M. Hebert. Applications of non-metric vision to some visually guided robotics tasks. Rapport de recherche RR-2584, I.N.R.I.A., Sophia-Antipolis, France, 1995.

[20] C. Sagüés and J.J. Guerrero. Motion and structure for vision-based navigation. *Robotica*, 17(4):355–364, 1999.

[21] D. Sinclair, A. Blake, and D. Murray. Robust estimation of egomotion from normal flow. *International Journal of Computer Vision*, 13(1):57–69, 1994.

[22] Z. Zhang and O. Faugeras. *3D Dynamic Scene Analysis*. Springer, Berlin, Heidelberg, 1992.

[23] Z. Zhang and O. Faugeras. A 3d world model builder with a mobile robot. *International Journal of Robotics Research*, 4(11):269–284, 1992.