

Extraction and Interpretation of Semantically Significant Line Segments for a Mobile Robot*

Xavier Lebègue and J. K. Aggarwal
Computer and Vision Research Center,
Dept. of Electrical and Computer Engr., ENS 520,
The University of Texas at Austin,
Austin, Texas 78712-1084, U.S.A.

Abstract

This paper describes new algorithms for detecting and interpreting linear features of a real scene as imaged by a single camera on a mobile robot. The low-level processing stages are specifically designed to increase the usefulness and the quality of the extracted features for a semantic interpretation. The detection and interpretation processes provide a 3-D orientation hypothesis for each 2-D segment. This in turn is used to estimate the robot's orientation and relative position in the environment and to delimit the free space visible in the image. Next, the orientation data is used by a motion stereo algorithm to fully estimate the 3-D structure when a sequence of images becomes available. From detection to 3-D estimation, a strong emphasis is placed on real-world applications and very fast processing with conventional hardware.

1 Introduction

This paper presents a new approach for extracting *semantically significant* line segments from monocular images, for estimating a mobile robot's orientation and relative position, and for identifying key objects associated with the free space, such as walls and doorways. The common idea is to design all the stages of image interpretation, including the lowest image processing level, such that they provide the higher stages with the most semantically useful features.

The tasks to be accomplished by the robot are usually specified in high-level semantic terms, such as "go down the hallway and go through the last door to the

left." In order to execute this task, the robot must be able to identify the objects of interest (here, hallways and doors) in its perception of the environment. One approach is to reconstruct a 3-D line segment description of the environment of the robot from several intensity images (see for example [1]), and then to match groups of those segments to selected object models. For this, several 3-D segments are selected according to orientation and position criteria [9]. A 3-D hypothesis of an object is generated and matched to the selected segments. The process is repeated for other segments and other hypotheses. The high-level semantic interpretation is then used for determining the free space or finding objects of interest.

Since the high-level interpreter is looking for line segments of particular orientations in 3-D, we designed our lower-level image-processing stages to take advantage of this information. We will show practical examples of how this top-down information can benefit the feature extraction stage by reducing the amount of unwanted features, increasing the sensitivity to good features, and drastically speeding the computation.

Section 2 of this paper discusses the *a priori* knowledge of a few prominent 3-D orientations for lines in the scene. Section 3 describes a very fast algorithm to extract useful line segments from a single image. The algorithm simultaneously hypothesizes the 3-D orientation of segments. Section 4 shows how the robot's orientation in the scene is extracted from a single image, and how the results of Section 3 may be used for a first semantic interpretation of the scene without the 3-D depth information. Finally, Section 5 extends these results to a sequence of *monocular* images. A Kalman filter is used for a recursive estimation of the position of 3-D segments. Unlike purely metrical approaches for the recursive estimation, the qualitative information derived independently for each image is used to enhance the quality and usefulness of the

*This research was supported in part by the DoD Joint Services Electronics Program through the Air Force Office of Scientific Research (AFSC) Contract F49620-89-C-0044, and in part by the Army Research Office under contract DAAL03-91-G-0050.

results while decreasing the computation time. The high-level semantic interpretation of the resulting 3-D maps is discussed. Experimental results with an indoor mobile robot are provided and analyzed.

2 Motivation and assumptions

2.1 Knowledge of prominent 3-D orientations

Estimating the 3-D structure of a scene from a single visual image is impossible without certain assumptions about the structure and the projection geometry. We chose to concentrate on objects which have parallel lines with known 3-D orientations in a world coordinate system. For example, in indoor scenes, rooms and hallways usually have a rectangular structure, and there are three prominent orientations for 3-D line segments: one vertical and two horizontal orientations perpendicular to each other. Outdoor urban scenes also display similar characteristics. In this paper, any 3-D orientation is permitted, as long as it is given to the algorithms. Therefore, more complex environments, such as polygonal buildings with angles other than 90 degrees, are handled as well if these angles are known. The assumption of known prominent 3-D orientations is consistent with the goal of identifying objects of interest for a mobile robot, such as walls, doorways, or entire buildings.

2.2 Orientation of the camera relative to the scene

Vertical lines constitute an interesting special case for two reasons: they are especially common in man-made scenes, and their 3-D orientation can easily be known in the 3-D camera coordinate system. If a 2-axis inclinometer is mounted on the mobile robot and properly calibrated, a 3-D vertical vector can be expressed in the 3-D coordinate system aligned with the 2-D image coordinate system. Inexpensive commercial inclinometers have a precision better than 0.01 degrees. Although vibrations and calibration errors may corrupt this measurement, it can serve as a good estimate of the camera's roll and pitch. We will assume that either the robot is equipped with an inclinometer, or that it drives on an approximately planar horizontal surface.

We estimate the third angular degree of freedom of the camera from odometer readings of the robot. Provided that the odometer is constantly corrected

by vision (which it will be in Section 4), the odometer does not drift without bounds. The orientation of the camera with respect to the scene follows from the orientation of the robot and the eye/wheel calibration data.

2.3 Coordinate systems and transformations

The coordinate systems, as shown in Figure 1, are W (the World coordinate system, with a vertical z-axis), R (the Robot coordinate system), C (the Camera coordinate system), and P (the coordinate system used for the perspective projection on the retina).

The homogeneous coordinate transformation matrix from W to R is:

$$T_{WR} = T_{roll}^{robot} T_{pitch}^{robot} T_{heading}^{robot} \begin{bmatrix} 1 & 0 & 0 & -x_W^{robot} \\ 0 & 1 & 0 & -y_W^{robot} \\ 0 & 0 & 1 & -z_W^{robot} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

x_W^{robot} , y_W^{robot} , and z_W^{robot} are approximated by the odometer but are not used in this section.

T_{RC} , the coordinate transformation matrix from R to C, needs to be completely determined through eye/wheel calibration. T_{CP} , the perspective projection matrix, is also determined through calibration. Therefore, we have

$$\begin{bmatrix} su \\ sv \\ s \\ 1 \end{bmatrix}_P = T_{CP} T_{RC} T_{WR} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_W$$

where u and v represent the coordinates in pixels of a point on the image plane.

2.4 Projecting segments of known 3-D orientation

In this section, we draw inferences about the likely 3-D orientation of the line segments from their 2-D projections in the image plane. With a pinhole perspective projection model, lines parallel to each other in the 3-D scene will converge to a vanishing point in the 2-D projection (refer to Figure 2). The knowledge of vanishing points provides useful information about the structure of the scene and about the camera's orientation with respect to the scene [2, 3, 10]. In particular, if the orientation of the camera relative to the scene is known, a vanishing point can be computed for each given 3-D orientation before the image

is processed. All the lines that have a given orientation in 3-D *must* pass through the associated vanishing point when projected. Conversely, if a line does not pass through a vanishing point, it cannot have the 3-D orientation associated with that vanishing point. In practice, if a line does pass through a vanishing point when projected, it is *likely* to have the associated 3-D orientation. The results of Section 3 will show how likely this is in practical situations.

To summarize, the line detection algorithm of Section 3 knows in each point of the image plane the orientation that a projected line segment would have if it had one of the predefined 3-D orientations. Therefore, the basic idea is to detect the 2-D segments with one of the possible orientations, and mark them with the associated 3-D orientation hypothesis.

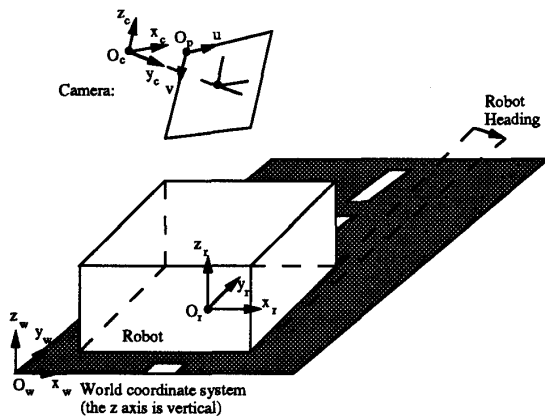


Figure 1: The coordinate systems

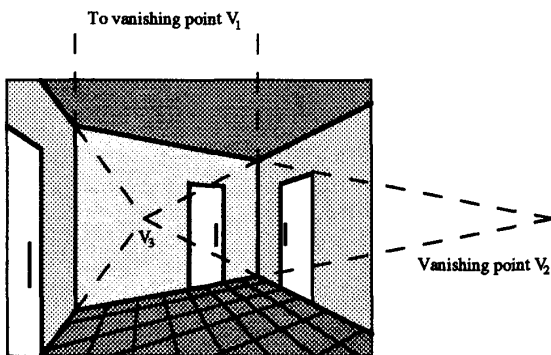


Figure 2: The effects of the heading on segment orientations

3 Detecting segments and estimating their 3-D orientation

3.1 Overview of the algorithm

The processing can be outlined as follows (see [7] for details):

1. Line support region extraction: compute the angle between the intensity gradient at each pixel and the expected direction of the projection of each 3-D orientation. Use a loose threshold to allow for noise in the gradient orientation. Reject improper pixels and 3-D orientations. Using the formulation of [7], the entire line support extraction is reduced to 8 additions and 11 multiplications per pixel and per 3-D orientation.
2. Non-maxima suppression: keep only the local gradient maxima along the estimated perpendicular to the line.
3. Pixel linking: create chains of pixels using a partial neighborhood search in the direction of the estimated vanishing points. This creates noisy linear chains.
4. Line fitting: perform a least-squares fit of line segments to the pixel chains. The pixel chains which cannot be closely approximated with a line segment are broken recursively into smaller chains.
5. Global orientation check: compute the match between each line and each 3-D orientation, like in the line support extraction step but with a much tighter threshold.

If the robot's *a priori* roll, pitch and heading are very uncertain, the lines will be extracted with loose thresholds, the true angles will be estimated, and the algorithm can then be run again with tight thresholds for the correct categorization.

3.2 Results

The algorithm was implemented in C on an HP 730 workstation, and tested on hundreds of indoor images obtained by our mobile robot (see [8] for a video presentation of the results). The predefined 3-D orientations are the vertical and the two horizontal orientations perpendicular to each other and aligned with the axes of our building. Figures 3 to 5 show the results of line extraction for one image in a sequence (The intensity image is deformed because the camera is equipped with a 6 mm wide-angle lens, as described

in Section 5). The processing time is only 1.2 seconds for each 512 by 480 image, from the intensity image to the list of categorized segments. The fast speed can be explained partly by the absence of multi-cycle floating-point instructions from the line orientation equations, when properly expressed. The lines are not broken up easily by a noisy gradient orientation, and the output quality does not degrade abruptly with high image noise, provided that the thresholds for local gradient orientations are loosened. A few misclassifications occur in some parts of the images, but are marked as ambiguities.

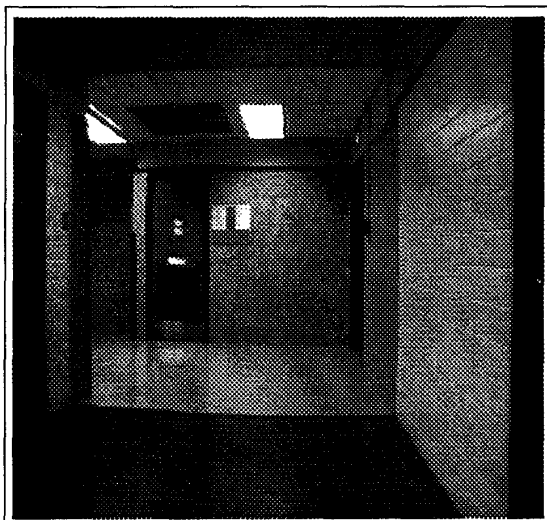


Figure 3: The input intensity image

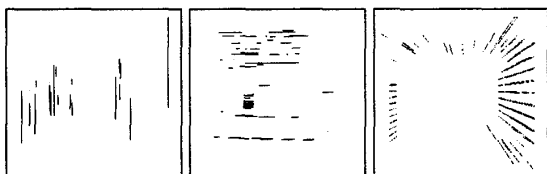


Figure 4: The line segments associated with each 3-D orientation

Table 1 compares the real and computed 3-D orientation of 1439 detected segments from eight images in three different environments. The presence of people in some scenes, as well as noise in the radio transmission of images, did not seem to generate many misclassifications. The percentages in each column sum to more than 100 % because of the multiple classification of ambiguous segments.

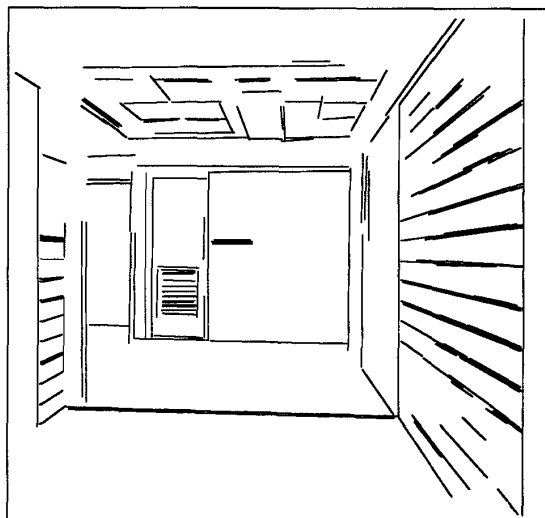


Figure 5: All of the 2-D segments

How detected segments are classified				
Classified as:	Real 3-D orientation:			
	Vertical	Hor. 1	Hor. 2	Other
Vertical	100 %	0 %	1.1 %	27 %
Horiz. 1	0 %	100 %	0.3 %	0 %
Horiz. 2	0.4 %	0 %	99.8 %	73 %
Total	100.4 %	100 %	101.2 %	100 %
No. of seg.	217	88	1123	11

Table 1: Hypothesizing 3-D orientations: success and error rates

4 Using information from a single image

4.1 Computing the robot's orientation

4.1.1 Algorithm

In this section we show how the robot's roll, pitch and heading may be computed *a posteriori* from the 2-D segments extracted in Section 3. To achieve the best precision, each angle should be evaluated from the vanishing point which varies the most with that angle. We begin with the roll, which is best derived from the 3-D vertical lines if the camera is looking approximately forward. Because of our definition of the heading (a rotation around the vertical axis of W), the vanishing point associated with the vertical lines does not depend on the heading. If all the 2-D segments classified as 3-D vertical are parallel in the image, the

vanishing point lies at infinity, the pitch p is zero, and the roll r is the 2-D orientation of the segments in the image. Otherwise, we first estimate the location (u_v, v_v) of the vanishing point by a weighted least-squares method. Let $T_{i,j}^{\text{RP}}$ be the element i, j of the matrix T_{RP} . After algebraic manipulations, we find:

$$r = \arctan \frac{\begin{pmatrix} (T_{2,3}^{\text{RP}} T_{3,2}^{\text{RP}} - T_{3,3}^{\text{RP}} T_{2,2}^{\text{RP}})u_v + \\ (T_{3,3}^{\text{RP}} T_{1,2}^{\text{RP}} - T_{1,3}^{\text{RP}} T_{3,2}^{\text{RP}})v_v + \\ (T_{1,3}^{\text{RP}} T_{2,2}^{\text{RP}} - T_{2,3}^{\text{RP}} T_{1,2}^{\text{RP}}) \end{pmatrix}}{\begin{pmatrix} (T_{2,1}^{\text{RP}} T_{3,2}^{\text{RP}} - T_{3,1}^{\text{RP}} T_{2,2}^{\text{RP}})u_v + \\ (T_{3,1}^{\text{RP}} T_{1,2}^{\text{RP}} - T_{1,1}^{\text{RP}} T_{3,2}^{\text{RP}})v_v + \\ (T_{1,1}^{\text{RP}} T_{2,2}^{\text{RP}} - T_{2,1}^{\text{RP}} T_{1,2}^{\text{RP}}) \end{pmatrix}}$$

The pitch of the robot may also be computed from the vanishing point of vertical lines, but for a better precision, we use another vanishing point more dependent on pitch and heading. In our implementation, we use the vanishing point which lies closest to the center of the image. We obtain the pitch and the heading with a method identical to the computation of roll described above.

4.1.2 Results

The quality of the results depends on the error on the vanishing point extraction and the precision of calibration. To illustrate the performance of heading computation from individual images, we plotted the path of the robot around a building floor. In Figure 6, the heading extracted by the vision algorithm was combined to the translation given by odometry to estimate the path of the robot. At the beginning and end of the experiment, the robot was approximately aligned with the center of a hallway. The robot stopped about five meters before the departure point. The total traveled distance was 125 meters, and 97 images were used to compute the headings. The algorithm seemed unaffected by people moving in the field of view and by a significant noise due to the transmission of the images by radio. The processing time for a *posteriori* heading estimation is negligible compared to the time needed for line extraction.

4.2 Image interpretation

Interpreting a single segment image is necessary when no 3-D segment information is available. This occurs for the first frames of a motion stereo sequence, or when the baseline of a stereo setup is too small compared to the depth of the scene. Ideally, interpreting a single segment image should not be too different from

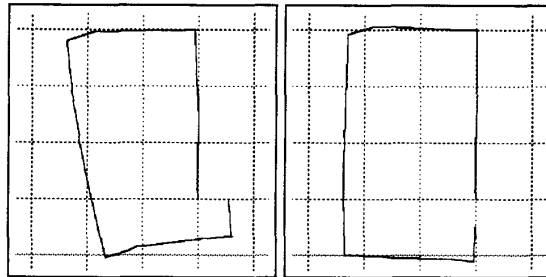


Figure 6: The path of the robot (a) as estimated by odometry, and (b) after correction by vision

interpreting a 3-D segment map to maintain consistency when the latter representation becomes available. Identifying objects by searching for segments of a particular 3-D orientation is a good step to achieve this consistency.

The interpretation of the individual segments in a single 2-D image requires *a priori* knowledge about the type of scene. It can be achieved by heuristic methods. The knowledge of the orientation of 3-D segments is particularly well suited to finding the floor and the walls in indoor images. Extracting the floor is essential for map-making, for determining the free space, and for computing the position of the robot relative to the visible environment. In most cases, the outline of the floor can be described entirely by segments of predefined horizontal 3-D orientation. A heuristic method can extract an approximate outline of the floor from a single segment image. In real applications, the floor may not be perfectly extracted, but this first approximation can be corrected after a sequence of images becomes available, by estimating recursively the 3-D position of the segments.

5 Estimating the position of 3-D segments

5.1 Introduction

The recursive estimation of 3-D segments from a sequence of images is the next logical step toward scene understanding. It aims at estimating the depth information that is absent from 2-D images. Existing approaches include [1, 4, 5], to name only a few. Viéville and Faugeras [11] concentrate on estimating motion and structure from *monocular* segment images and present results on the reconstruction of a calibration pattern. Jezouin and Ayache [6] review different

segment-tracking techniques using monocular images and present results on 3-D reconstruction from synthetic aerial images.

Applications in autonomous mobile robots imply the ability to deal with real indoor scenes while using limited computation resources. As in the other sections of this paper, we aim at providing useful features for a semantic interpretation of the environment of the robot. We only consider the segments which have been extracted by the algorithm of Section 3. We will show how this choice benefits the robustness and speed of the recursive estimation process.

5.2 Preliminary observations

A 3-D segment should be observed from very different angles to be accurately reconstructed. Two approaches can be used to achieve this with a monocular camera. The first one, active vision, pans and tilts the camera to keep some interesting features in the field of view. The second approach, which we chose, consists of fitting the camera with a wide-angle lens. A 3-D segment is usually first detected when far ahead of the robot, but as the robot drives closer, the 2-D projection moves to one side of the focus of expansion. A short focal length will enable the 3-D segment to remain in the field of view long enough for an accurate reconstruction. The drawback to using a wide-angle lens is geometric distortion. The first processing step is therefore to correct this distortion.

The 3-D segments which are parallel to the direction of motion cannot be properly reconstructed without considering their endpoints. The experience of all researchers using real images indicates that endpoints are unreliable, because 2-D segments may be broken or extended (due to specular reflection, for example). Therefore, we will not use endpoints to estimate the parameters of infinite 3-D lines. There is no justification for wasting computation to estimate features that will not help the interpretation of the scene. In practice, we will not attempt to reconstruct the 3-D segments parallel to the direction of motion. Those segments are sent to the semantic interpreter with an estimate of orientation but not of position, unless the robot changes its course. This contrasts with other approaches that do send an estimate of the position and orientation of those segments, both with a huge uncertainty.

Figure 7 presents a block diagram of the recursive estimation of 3-D segments. As in Section 4, we combine translations from odometry to rotations from vanishing points to estimate the motion of the robot. Because the rotations are obtained from each

2-D image with an excellent accuracy, we do not introduce any uncertainty on their parameters. Translations, however, rely partly on odometry. Therefore, we model the location of the robot in the world coordinate system by a 3-D vector t_k and its covariance matrix $E(t_k t_k^T)$ under the Gaussian noise assumption.

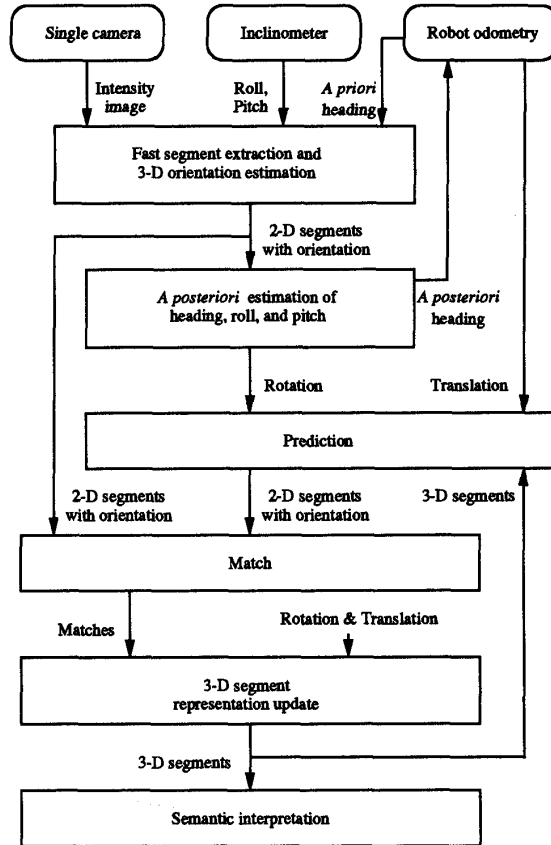


Figure 7: Recursive estimation of 3-D segments

5.3 Representation of lines

Since the algorithm of Section 3 already provides the most likely 3-D line orientation, we only estimate two positional parameters for each 3-D line. Let us define a coordinate system L_n for each predefined 3-D orientation n as follows: L_n is a rotation of W so that the 3-D orientation n is parallel to the z axis of L_n . The $T_{L_n, W}$ matrices are all known by definition. The state vector of a tracked 3-D line is then made of the first two components of the coordinates in L_n of any point of the line.

Because we are not using stereovision, a problem arises when a 2-D segment is acquired for the first time. Our approach is to model it by a state vector with arbitrary depth and a very large uncertainty in depth. After repeated observations from different angles, the uncertainty ellipsoid associated with the state vector will shrink around the best estimate of the position.

The measurement $y_{k,i}$ of a 2-D line in frame k given its associated 3-D orientation is defined as follows: if the line is "rather horizontal," it can be described in the image plane by $v = \alpha u + y_{k,i}$ with some parameter α . If the line is "rather vertical," $y_{k,i}$ is such that $u = \alpha v + y_{k,i}$. Deciding whether a line is rather horizontal or vertical is based on the estimated 3-D orientation, so that singularities in the representation are avoided yet consistency across frames is maintained. The explicit state and measurement equations for the Kalman filter are not derived here for lack of space.

5.4 Implementation and results

The implementation benefits from hypothesizing the 3-D orientation of lines before estimating their 3-D position in the following ways:

- Matching is attempted only between segments of the same orientation in 3-D. In practice, other implementations of 3-D segment recursive estimators use arbitrary 2-D orientation "buckets" to achieve the same purpose. By matching segments of given 3-D orientations, we have a more formal approach to the problem of limiting possible matches.
- No computation is wasted on segments which cannot be reconstructed because they are parallel in 3-D to the direction of motion.
- The problem of singularities in the representation (see [6] for example) is avoided: although there is still a need for different line parameterizations, the appropriate one needs to be determined only once for each segment being tracked.
- Traditional approaches have difficulties dealing with moving objects and a moving camera. Large moving objects will corrupt the 3-D map of the environment and yield incorrect motion estimates if the egomotion is estimated from the image. By tuning the line extractor to certain orientations of interest, it is unlikely that an unexpected moving object could durably display misleading features. Therefore, the identification of the visible objects

of interest will not be significantly hampered by large moving objects. This is extremely important in real-world situations.

Several approaches are possible for tracking segments across images. If a conventional token-tracker is used [5], matching is necessary only between segments with the same 3-D orientation. The results we are presenting here have been obtained with manual token-tracking. We are currently working on a new type of token-tracker that relies on topological and semantic information to eliminate incorrect matches.

Figure 8 shows the vertical segments in a top view of a corridor, as estimated after 1, 4, 7, and 10 images in a sequence. The robot moved nine meters during the sequence, or about a quarter of the length of the corridor (the motion is from left to right in the figure). The segments visible for the first time are given an arbitrary depth and a very large uncertainty in depth. The width of the corridor at the location of the robot was estimated with an error varying from 0.4% to 4.7%.

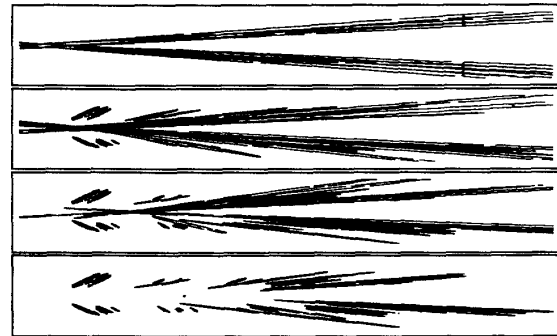


Figure 8: Top view of a corridor after 1, 4, 7, and 10 images (uncertainty of vertical segments)

5.5 High-level semantic interpretation

The semantic interpretation of the 3-D segment maps should be consistent with the interpretation of single images since the 3-D information is not present or accurate for some segments. The search for segments associated with objects of interest depends primarily on 3-D segment orientation and 2-D image topology. The position of 3-D segments is used in a later stage to verify object hypotheses and to estimate their range. The reflection of segments in the floor, often a problem with 2-D interpretation, is handled by clipping the 3-D segments below the floor plane.

Vertical segments in the top view				
Door no.	Δx (in mm)	Δy (in mm)	Error in Δx	Angle of the frame
1	917	-54	-9.7%	-3.4 deg
2	1083	4	6.6%	0.2 deg
3	1013	-18	-0.3%	-1.0 deg
4	996	25	-2.0%	1.4 deg
5	769	-62	-24.3%	-4.6 deg
6	1114	16	9.6%	0.8 deg
7	421	-72	-58.6%	9.7 deg
8	292	99	-71.3%	18.7 deg

Table 2: Comparing estimated door frame edges to their model

The semantic interpretation of particular objects is application-dependent. To illustrate how the recursive estimation of oriented segments can be used by an interpreter, we will describe the detection of the door frames in the corridor of Figure 8. The door frames are aligned with the axes of the building, therefore segments can be selected based on their 3-D orientation and 2-D topology. To verify a door frame hypothesis, the two selected vertical segments are checked for their distance and the 3-D location of their endpoints. The segments should also belong to a 3-D plane formed by two predefined 3-D orientations. Since the positions of the 3-D segments are uncertain, the Mahalanobis distance is used to adapt the thresholds.

As an illustration, Table 2 analyzes the vertical edges of the door frames shown in the last top view of Figure 8. Ideally, the two vertical segments should be 1016 mm apart and the angle of the door frame in the scene should be zero. The last two doors are about 9 meters ahead of the robot and the 3-D position of their segments is still very uncertain. The number of images in which the third and fourth door frames were visible explains the high precision of their reconstruction.

6 Conclusion

We have presented a new approach to the interpretation of monocular line images for the navigation of a mobile robot. We described algorithms for interpreting single images and for recursively updating interpretations with monocular sequences of images. The quality and usefulness of the results of each processing stage justified the practical constraint placed on the problem: the *a priori* knowledge of a few prominent 3-D segment orientations. This constraint is reason-

able since even humans get disoriented when placed in a tilted environment, despite having our own "inclinometers" in our inner ears. By reasoning in terms of what features the high-level interpretation stage will actually use, we were able to design lower-level algorithms for better and faster processing. The ultimate benefits of this approach were demonstrated on real images in real situations, using conventional computing hardware, and with image acquisitions every few seconds.

References

- [1] N. Ayache. *Artificial Vision for Mobile Robots*. The MIT Press, 1991.
- [2] S. T. Barnard. Interpreting perspective images. *Artificial Intelligence*, 21(4):435-462, November 1983.
- [3] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127-139, March 1990.
- [4] J. Crowley and P. Stelmaszyk. Measurement and integration of 3-D structures by tracking edge lines. In *Proc. First European Conf. on Computer Vision*, pages 269-280, Antibes, France, April 1990. Springer-Verlag.
- [5] R. Deriche and O. Faugeras. Tracking line segments. In *Proc. First European Conf. on Computer Vision*, pages 259-268, Antibes, France, April 1990. Springer-Verlag.
- [6] J. L. Jezouin and N. Ayache. Computing 3D structure from a monocular sequence of images with known motion. Technical report, Matra-LTIS, BP 235, 78052 Guyancourt, France, 1991.
- [7] X. Lebègue and J. K. Aggarwal. Detecting 3-D parallel lines for perceptual organization. In *Proc. Second European Conf. on Computer Vision*, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.
- [8] X. Lebègue and J. K. Aggarwal. Semantically significant line segments for a mobile robot... ROBOTEX. In *Video Proc. IEEE Int. Conf. Robotics and Automation*, Nice, France, May 1992.
- [9] M. Thonnat. Semantic interpretation of 3-D stereo data: finding the main structures. In *Proc. Eighth Int. Conf. on Pattern Recognition*, pages 1051-1054, Paris, France, October 1986.
- [10] T. Tsubouchi and S. Yuta. Map assisted vision system of mobile robots for reckoning in a building environment. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1978-1984, Raleigh, March 1987.
- [11] T. Viéville and O. Faugeras. Feed-forward recovery of motion and structure from a sequence of 2D-line matches. In *Proc. Third Int. Conf. on Computer Vision*, pages 517-520, Osaka, Japan, December 1990.