We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



9

Visual Navigation for Mobile Robots

Nils Axel Andersen, Jens Christian Andersen,
Enis Bayramoğlu and Ole Ravn
Technical University of Denmark, Department of Electrical Engineering
Denmark

1. Introduction

Looking at the number of living creatures using vision as their main sensor one should expect that vision also would be the first choice for mobile robots considering nice features as low price, low power, non contact and high potential information contents. Unfortunately it has proven much more difficult to extract the information from vision than expected and still no commercial robot relies on vision as its main sensor.
In spite of this several successful methods have been developed. This chapter presents a number of visual methods that has been experimentally verified: artificial visual landmarks, corridor following using vanishing point, and road following using terrain classification based on data fusion of laser scanner and vision.

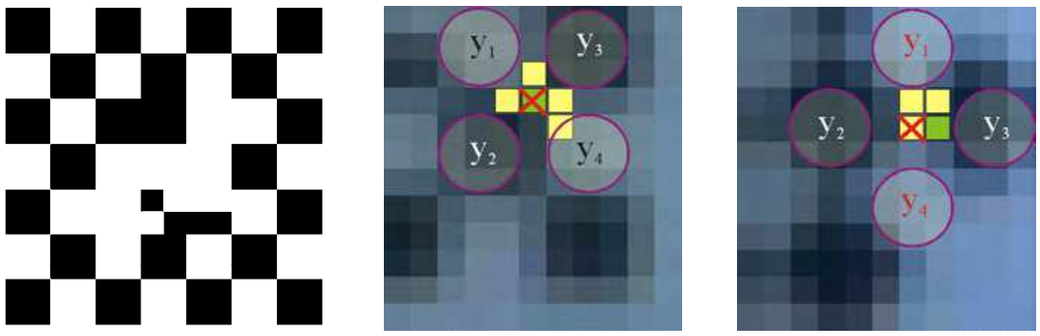
2. Artificial visual landmarks

In well structured environments (both indoor and outdoor) a mobile robot is able to navigate a reasonable distance based on odometry and inertial measurements. However to keep the navigation error bounded absolute position measurements are needed. These can be provided by visual landmarks. Landmark navigation is based on the assumption that the robot from recognizing a landmark can get a localization reference.
The landmark could be artificial and placed to be recognized by the robot, i.e. for indoor applications a method is to place unique landmarks on the ceiling and let a robot camera look for these landmarks, and further place the landmarks at so short intervals that the robot could navigate from one landmark to the next with sufficient accuracy to be able to find the next landmark.
The landmark itself could be at a known position, or just act as a unique reference position so that any ambiguity or accumulating errors could be resolved or reduced when recognizing the landmark. The initial position of the robot could be resolved by recognition of a unique artificial landmark. This landmark could refer to an entry in the robot database with knowledge of that specific area. One or more landmarks could be placed close to a recharging station that requires specifically accurate navigation.
Artificial visual landmarks have been studied by several researchers. Kabuka and Arenas (1987) study a standard pattern and present a thorough error analysis using simulation. Lin and Tummala (1997) describe a system based of simple geometrical patterns using a

Modified Elliptical Hough Transform for detecting the landmark and its properties. Bais and Sablating (2006) present a landmark based system used for soccer robots.

2.1 Design of an artificial landmark

The landmark must be easy to recognize and distinguish from other items normally occurring in the environment of the mobile robot. It should be detectable at different distances and at different angles. One of the easy and stable detectable shapes are checkerboard corners, they are scale invariant, and to some extent viewing angle invariant. The landmark should be simple to reproduce, and thus printable on a sheet of paper would be preferable. The final design was selected with a double checkerboard frame and a central area for a unique code as shown in Fig. 1.



(a) (b) (c)

Fig. 1. The landmark in (a) consists of a checkerboard frame and a central code. The camera view of the lower left part is shown in (b) and (c). The corner filter uses four 3 x 3 pixel areas to detect a corner at the centre position.

The centre code holds 9 square areas; each can be filled with a 4 bit code. Two of the top left squares are used as orientation marks to make it possible to decide the orientation of the landmark. This leaves 7 squares for codes. A few code combinations can be confused with the orientation mark and must thus be avoided, leaving some 24 usable bits except for 2^{16} unusable code combinations, or in total 16711680 code possibilities.
A smaller frame with just 4 squares in the centre would be sufficient in most cases, with one corner as orientation mark, and at maximum 3 bits used in the remaining a total of 9 bits or 512 codes would be available.
The landmark in Fig. 1(a) has two black bits set in each of the two least significant squares -- bottom right inside the frame -- corresponding to the value $9C_{hex}$ or 156 decimal.

2.2 Detection of a landmark

The landmark is detected in four steps: corner detection, frame detection, code detection and frame localization, each of these steps are described in the following.

2.2.1 Detection of corners

The corner detection is done by performing a comparison of four areas in the image expected to match two black areas and two white areas. The full image is first filtered using a 3 x 3pixel Gaussian kernel with σ =0.95. This yields the central pixel a weight *G(r,c)* of about as much as the sum of weights of the remaining 8 pixels. A set of corner pixels C_1 is found using equation **(1)**

$$
\begin{aligned}
&a = (r,c) \in C_1 \\
&if \quad (y_1 - y_2 > k_c \wedge y_1 - y_3 > k_c \wedge y_4 - y_2 > k_c \wedge y_4 - y_3 > k_c) \\
&where \\
&y_1 = G(r-2, c-2) \\
&y_2 = G(r+2, c-2) \\
&y_3 = G(r-2, c+2) \\
&y_4 = G(r+2, c+2) \\
&k_c = \tfrac{1}{3}(\max(y_1, y_2, y_3, y_4) - \min(y_1, y_2, y_3, y_4)) + k_{\min} \\
&w = y_1 + y_4 - y_2, y_3
\end{aligned}
\tag{1}
$$

An intensity difference is required from all bright pixels to all black pixels. This difference must be greater than a threshold k_c which proportional with the intensity difference from the brightest to the darkest pixel and includes a fixed minimum threshold k_{min}= 7 out of 256 intensity levels. This ensures that a landmark in both bright areas and darker areas is detectable.

This filter will detect a corner that is bright in the upper-left and lower-right part. The set of intensity-reversed corners C_2 is found in the same way by exchanging the calculation of the for pixel differences.

The landmark may however be observed at different orientations depending on the positioning of the landmark and the camera. By adding a guard band of one pixel between the 4 corner areas, the filter will be relatively insensitive to viewing angle rotation of the landmark relative to the camera, the corner detection sensitivity will be reduced as the angle increases, and at 45 degrees the sensitivity will be zero. To be able to detect landmarks at any rotation angle a second set of filter masks (rotated 45 degrees) is added as shown in Fig. 1 (c), using the same filter function as above giving two more sets of corner pixels C_3 and C_4

The corner pixels in {C_1, C_2 , C_3 and C_4 }are then enumerated individually using 8-connectivity into the total set of corner groups H_n. The centre of each corner group $h_n(H_n)$ is found as shown in equation **(2)**

$$
h_n(H_n) = \frac{1}{\sum_{i \in H_n} w_i} \sum_{i \in H_n} a_i w_i \tag{2}
$$

where the accuracy improvement by using the intensity weight w_i from equation 1 is neither quantified nor optimized. In Fig. 1 (b) and (c) the detected corner pixels are shown as bright colored squares, where dark green marks the pixel closest to found corner position.

2.2.2 Detection of frames

The corner positions have to match with the frame of the landmark, i.e. from a frame corner there should be two sets of 6 corners each describing a straight line following a frame edge, and further the corners should have almost the same separation.
Fig. 2 shows an example with three landmarks -- rotated at different angles. The original image is shown faintly in the background. The corner pixels are color coded, each color corresponds to one of the four filters.

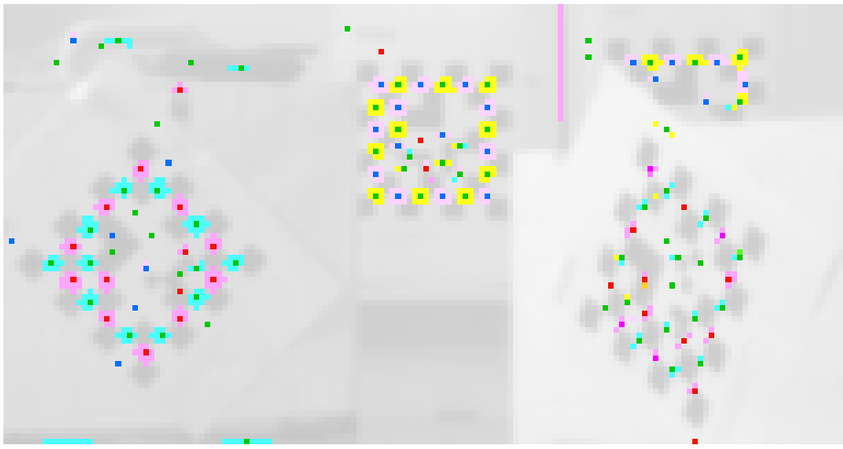


Fig. 2. The corner pixels from three landmarks are shown. The different colors correspond to the four used corner filters.

From all corner positions the up to eight closest neighbors are found within a maximum distance – of $\frac{1}{5}$ image height -- allowing a landmark to fill the whole image.
A frame edge corner set f_j is therefore described as six corners on an approximately straight line fulfilling the requirements in equation (3)

$$f_j = \left\langle \begin{array}{l} \{h_i\} \\ i \in [1,2,...,6] \\ h_i \in h_n \end{array} \middle| \begin{array}{l} v_2 = h_2 - h_1 \\ \hat{h}_2 = h_2 \\ \hat{h}_{i+1} = \hat{h}_i + v_i \\ v_{i+1} = v_i + k_g (h_i - \hat{h}_i) \\ |h_i - \hat{h}_i| < k_{\lim} |v_i| \\ i \in [2,5] \end{array} \right\rangle \quad (3)$$

where $k_{lim} = 0.37$ and $k_g = 0.5$.

A frame corner should be the end point of at least two frame edges. In this way a full frame F is a set of four frame edges as described in (4) with the corners in each edge ordered as described in frame conditions

$$F = \left\langle \{f_1, f_2, f_3, f_4\} \left| \begin{array}{l} F_{n,m} = h_m \in f \\ F_{1,1} = F_{2,1} \\ F_{2,6} = F_{3,1} \\ F_{1,6} = F_{4,1} \\ v_n = F_{n,6} - F_{n,1} \\ \left|v_1 - v_3\right| =< k_p \left|v_1\right| \\ \left|v_2 - v_4\right| =< k_p \left|v_2\right| \end{array} \right. \right\rangle \tag{4}$$

where the frame is described counter-clockwise, so that f_1 is the topmost edge and f_2 is the leftmost edge. The edges of a frame should in pairs be approximately parallel and of equal lengths, i.e. $f1$ parallel to f_3 and f_2 parallel to f_4 with the limits k_p = 0.5.
The six corners h_1 to h_6 are fitted to a straight line, and the crossing of this line with the line from one of the adjacent edges is used as the true frame corner. The frame can then alternatively be described by these four frame corners, describing the frame counter-clockwise with h'1 as the topmost corner

$$F = \left\{h_1', h_2', h_3', h_4'\right\} \tag{5}$$

2.2.3 Detection of code

The code in the landmark requires detection of black and white areas in the centre of the frame, and each of the code bits cover an area of only a quarter of the blocks in the frame. The intensity level that separates a black bit from a white bit must further be determined.
The size of the code bits are selected so that the probability of detection for the frame and the code vanishes at about the same distance. At the distance where the frame is detected with a 95% probability, the code is correctly detected with a probability of about 95% (of the instances where the frame is detected).A frame grid is constructed by dividing the distance between two adjacent corners on every frame edge in two. The corners are projected to the fitted line edge, but the distance between the corners are not equalized.
The correct distance between the corners may change over the frame if the landmark is seen in perspective, but this effect is mostly significant if part of the landmark is very close to the camera, and being close to the camera the code detection is usually easy, as a high number of pixels are available for each code bit.
At longer distances all cells in the grid will be very close to the same size, and here the grid accuracy is more important for code recognition. A minor improvement in code detection may therefore be obtainable if the grid spacing along the edges was made equal.

Fig. 3. shows an example of some detected landmarks. The code grid is painted in green and includes the inner part of the frame blocks as well as the code area itself.
All pixels inside the green area are evaluated as belonging to one of the cells, and the average intensity of the pixels inside the cell is used to estimate its value.

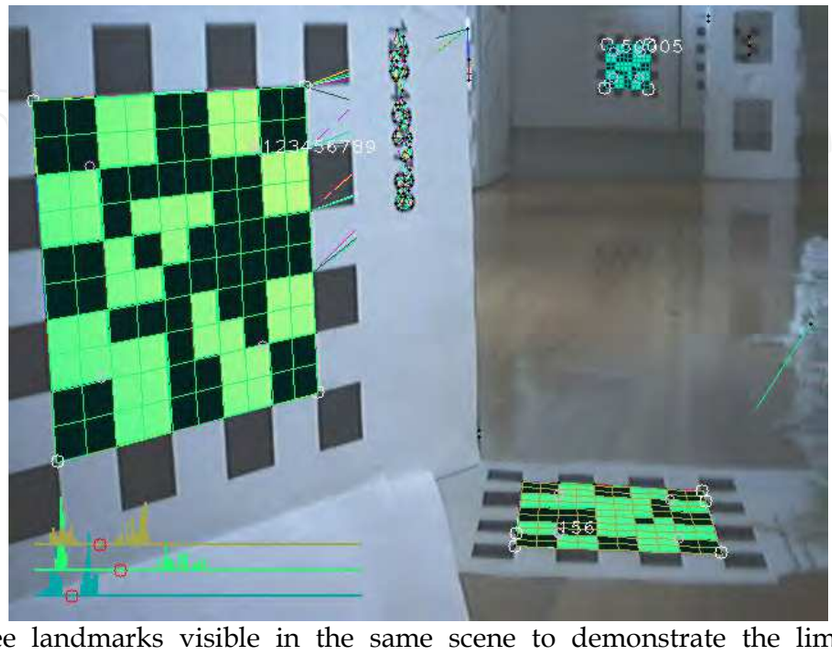


Fig. 3. Three landmarks visible in the same scene to demonstrate the limitations. All landmarks have a frame width of 17.5cm. The near is at 0.46m and the far at 3.2m, the one on the floor is tilted 79^0. The image resolution is 640 x 480 pixels.

Half of the cells covering the frame blocks are always white, the other half black. The average intensity for these cells is used as a threshold value when classifying bits as black or white in the code area.
The histograms in the bottom left corner of Fig. 3 show the distribution of intensity values for each of the three detected landmarks. Left is black, right is white on the histogram line.
The histogram color should match the grid color painted on top of the landmark.
The two all black code areas for the orientation mark are located, and the code values in the remaining code area are ordered accordingly. The two FF_{hex} squares mark the top left corner of the code area, and in this orientation the four bits in each block are coded as follows:

1	2
3	4

The ordering of the code blocks from the most significant to the least significant is located as follows:

*	*	1
2	3	4
5	6	7

The code in the large landmark in Fig. 3 is therefore $75BCD15_{hex}$, or in decimal 123456789.

2.2.4 Position estimation

The landmark position relative to the camera can be estimated when the size of the landmark is known and the camera geometry is assumed to be available. The landmark corner positions are known with a relatively high accuracy, as these are averaged from the line fitting of the frame edge. The perspective described by the positioning of the landmark corners should therefore allow a reasonably accurate estimate of the orientation too. The landmark position and orientation is estimated using a least square parameter estimation method. From the frame extraction algorithm above the position in the image of the frame corners are known $h'_i = (h_r,h_c)$. The position of the corners on the landmark surface is known from the landmark design as four coordinate pairs. The coordinates on the landmark are selected as being seen in the same way as a robot, that is x being forward (in front of the landmark), z being up, and when looking in the direction of x (from behind the landmark) y is to the left. When looking at the landmark on a wall, then z is up and y is right.

The centre of the landmark is taken as the reference position, i.e. the top right frame corner has the frame coordinate $B = [0,b_y,b_z]^T$ with positive values for both b_y and b_z.

A landmark may be at any position $g_t = [x,y,z]^T$ relative to the robot and rotated following the normal convention: first turned κ around the vertical z-axis with positive being counter-clockwise, then tilted Φ around the y-axis with positive being a down tilt and finally roll Ω around the x-axis with positive being a roll to the right.

When a point on the landmark surface $B = (0,b_y,b_z)$ is being seen at the 3D position $A = [a_x, a_y , a_z , 1]^T$ in robot coordinates, then A and B are related with the landmarks orientation and position (x,y,z,Ω, Φ,κ) (also in robot coordinates) as in (6)

$$\begin{bmatrix} 0 \\ b_y w \\ b_z w \\ w \end{bmatrix} = R_\Omega R_\Phi R_\kappa T A \tag{6}$$

where $R_\Omega, R_\Phi, R_\kappa$ are rotation matrices in homogeneous coordinates and T is a translation matrix as shown below:

$$R_\Omega = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\Omega) & \sin(\Omega) & 0 \\ 0 & -\sin(\Omega) & \cos(\Omega) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

$$R_\Phi = \begin{bmatrix} \cos(\Phi) & 0 & -\sin(\Phi) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\Phi) & 0 & \cos(\Phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$R_\kappa = \begin{bmatrix} \cos(\kappa) & \sin(\kappa) & 0 & 0 \\ -\sin(\kappa) & \cos(\kappa) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

$$T = \begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & -z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

The conversion between image coordinates and the 3D position A of the point on the landmark are -- except for lens distortion -- as defined in (11)

$$\begin{bmatrix} h_r w \\ h_c w \\ w \end{bmatrix} = \begin{bmatrix} -1 & 0 & h_x \\ 0 & 1 & h_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ \frac{1}{c} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \\ 1 \end{bmatrix} \tag{11}$$

$$I = bPA$$

where $I = [h_r w, h_c w, w]^T$ holds the row h_r and column h_c of the corresponding pixel position of $A = [a_x, a_y, a_z, 1]^T$ in the image.
The b-matrix offsets the position to get positive row and column values by adding the (optical) image centre (h_x, h_y)and changing the direction of the row axis to get down as positive. P-matrix adds the perspective by scaling the row and column values by 1/c into w proportional to the distance from the camera a_x. The a_y direction corresponds to columns in the image with changed sign, and the height a_z corresponds to image rows.

When the camera is positioned at the centre of the robot coordinates, then the two equations (11) and (6) can be combined as shown in (12)

$$\begin{bmatrix} h_r w \\ h_c w \\ w \end{bmatrix} = bPT^{-1} R_\kappa^T R_\Phi^T R_\Omega^T \begin{bmatrix} 0 \\ b_y \\ b_z \\ 1 \end{bmatrix} \tag{12}$$

The right side of this equation can be evaluated to three functions of the unknown $v=[x,y,z,\Omega, \Phi,\kappa]^T$ and the known position (b_y, b_z) as

$$\begin{bmatrix} h_r w \\ h_c w \\ w \end{bmatrix} = \begin{bmatrix} f_r(x, y, z, \Omega, \Phi, \kappa, b_y, b_z) \\ f_c(x, y, z, \Omega, \Phi, \kappa, b_y, b_z) \\ f_w(x, y, z, \Omega, \Phi, \kappa, b_y, b_z) \end{bmatrix} \tag{13}$$

The last w equation can be inserted into the first two as in (14) where the six unknowns are replaced by the vector v

$$\begin{bmatrix} h_r f_w(v, b_y, b_z) - f_r(v, b_y, b_z) \\ h_c f_w(v, b_y, b_z) - f_c(v, b_y, b_z) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{14}$$

To solve for the six unknowns at least six equations are needed, so the four corners of the landmark frame yield eight equations by substituting b_y, b_z , h_r and h_c in (14) with the values from the remaining three corners. The problem may then be solvable. The eight functions on the left side of (14) should now all evaluate to zero with the correct value of the six unknowns

$$F = 0 \tag{15}$$

As the functions are nonlinear the six unknown parameters are estimated using Newton's iteration method. With an initial guess of the ^v the equations will (probably) not be zero, but assuming that the errors are small and the equations are approximately linear at the guessed position, the error can be compensated for by a linear adjustment Δv as shown in (16)

$$F(\hat{v}) + J(\hat{v}) + \Delta v = 0 \tag{16}$$

where $F(\hat{v})$ is the value of the eight equations evaluated with the guessed set of parameters $\hat{v}$ and $J(\hat{v})$ is the Jacobian F with respect to the unknowns in v taken at $\hat{v}$, finally Δv is the adjustment to the guess needed to get the required zero result.

A better guess of v would therefore be $\hat{v}_2$ as shown in (17)

$$\hat{v}_2 = \hat{v} + \Delta v \tag{17}$$

The estimated adjustment Δv is found by solving (16) as:

$$\Delta v = -(J^T J)^{-1} J^T F \tag{18}$$

Equations (16), (17) and (18) are then repeated, setting $\hat{v} = \hat{v}_2$ for the next iteration, until the estimated parameters have converged sufficiently. The pixel position in the image is adjusted for radial lens error prior to insertion into the functions in F

The iteration is terminated when the parameter change Δv_n in iteration n is significantly small according to the stop criteria in (19)

$$stop\,criteria: \Delta v_n = \left[\hat{x}, \hat{y}, \hat{z}, \hat{\Omega}, \hat{\Phi}, \hat{\kappa}\right] \left| \begin{array}{l} \left|\hat{x}, \hat{y}, \hat{z}\right| < P_{\inf} \wedge \\ \left|\hat{\Omega}, \hat{\Phi}, \hat{\kappa}\right| < R_{\inf} \end{array} \right. \tag{19}$$

When looking at a landmark that is tilted slightly forward or backward it may be difficult to see the difference, this could indicate local optimum that could trap the parameter estimation.

Fig. 4 shows the pixel error as a function of a combination of turn (κ) and tilt (Φ). This shows the correct value for these parameters ($\Phi=5^0$) and ($\kappa=22^0$) but also a local minimum at about ($\Phi=-13^0$) and ($\kappa=25^0$)

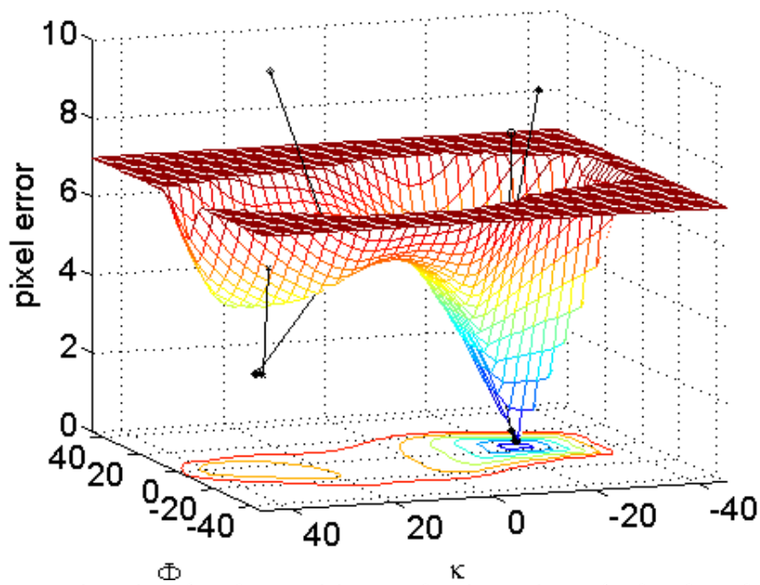


Fig. 4. Parameter estimation for the position and orientation of a landmark has (often) local minimum. The pixel error is shown as a function of turn of turn (κ) and tilt (Φ) of the landmark (limited to a maximum error of seven pixels).

The rotation (Ω) of the landmark will have four equally accurate solutions, as there is no discrimination of the four frame corners. But the position of the code index is known and is used to get the correct Ω -value. The position has a local minimum at the same distance behind the camera, but this is easily avoided by selecting an initial guess in front of the camera (a positive x-value). To avoid the (κ,Φ) local minimum, four initial positions in the four quadrants in the κ Φ–coordinate system are tested, and after a few iterations the parameter set with the least pixel error is continued to get a final estimate. The iteration error progress is shown in Fig. 4 as four black lines, of which two ends in the local minimum at with a minimum pixel error of 2.5 pixels, compared to 0.12 pixels at the global minimum in $\Phi=5.0^0$ and $\kappa=-22.8^0$.

2.3 Experimental results with artificial visual landmarks

Three experiments have been carried out to evaluate the performance of the landmark system in navigation of mobile robots. The first test investigates the repeatability of the position estimates by taking several readings from the same position with different viewing angle. The second test deals with the robustness of the landmark code reading function performing hundreds of code readings from different positions and the third experiment uses the landmarks to navigate a mobile robot between different positions showing that the drift of odometry may be compensated by landmark position readings. In all three experiments the used camera is a Philips USB-web camera with a focal length of 1050 pixels. The used image resolution is 640 X 320 pixels.

2.3.1 Relative estimation accuracy

In this experiment the repeatability of the position measurement is tested. The camera is placed 2.2 m from the landmark and the landmark is placed with four different viewing angles. At each viewing angle 100 measurements are taken. As seen in Table 1 the estimation accuracy of a landmark position is dependent on the viewing angle of the landmark.

Viewing angle κ	Position $\sigma_{\lvert x,y \rvert}$	Orientation σ_{Ω} (roll)	Orientation σ_{Φ} (tilt)	Orientation σ_{κ} (turn)	Block pixels	N samples
0^0	1.7 mm	0.04^0	1.55^0	0.61^0	11.9	100
10^0	1.3 mm	0.06^0	0.72^0	0.27^0	11.7	100
30^0	2.2 mm	0.12^0	0.21^0	0.12^0	10.3	100
60^0	2.5 mm	0.10^0	0.11^0	0.06^0	5.9	24

Table 1. Relative estimation accuracy of 100 position requests of a landmark at 2.2 m at different turn angles of the landmark.

The position estimation error in (x, y) is about 0.2 cm and is partially correlated with an estimation error in the landmark orientation; typically a small tilt combined with a slight turn makes the landmark seem slightly smaller and thus further away.
When the turn angle is zero (landmark is facing the camera) the relative estimation error in roll σ_{Ω} is uncorrelated with the other errors and thus small, at larger turn angles the roll error increases and the error value is now correlated with the other estimated parameters.
The obtainable absolute position accuracy is dependent on the mounting accuracy of the camera, the focal length of the lens and the accuracy of the estimated lens (radial) errors. With the used USB camera an absolute position accuracy of less than 5 cm and an angle accuracy of less than 5^0 is obtained within the camera coverage area.
When a landmark is viewed with a rotation of 22.5^0 -- just in between the two sets of corner filters ($C_{1,2}$ and $C_{3,4}$) -- the sensitivity is slightly reduced. This reduces the distance at which the landmark can be detected.
The number of pixels needed for each of the squares in the frame to be able to detect the landmark is shown in Table 2 as 'block pixels'.

Orientation of grid	pd= 0.5 Pixels	meter	pd= 0.95 Pixels	meter
0.00^0	3.8	3.4	3.9	3.3
22.5^0	4.6	2.8	4.8	2.7
45.0^0	4.2	3.1	4.3	3.0

Table 2. Number of pixels needed for each frame block to detect landmarks at different rotation angles relative to camera. The distance in meters corresponds to a focal length of 525 pixels (image size of 320 x 240 pixels)

When the probability of detection (pd) is about 0.95 the landmark code is evaluated correctly with a probability of about 0.95 too (for the detected landmarks). Stable landmark detection requires that each of the blocks in the landmark frame should be covered by at

least five pixels. When the landmark is not at the distance with the optimal focus the detection distance will decrease further.

2.3.2 Landmark code reader test

To test the landmark code reader performance an experiment with a small mobile robot (see Fig. 5) has been done.



Fig. 5. Mobile robot used for experiments.

Two landmarks have been put on the wall beside two office doors. The distance between the two landmarks is approximately 8 meters. A black tape stripe is put in the middle of the corridor and the mobile robot is programmed to run between the two landmarks following the black tape stripe. At each landmark the robot turns 90 degrees and faces towards the landmark at a distance of approximately 1.5 meters. The code of the landmark is read by the robot and compared to the expected code. In one experiment the mobile robot goes back and fro 100 times which is about the maximum allowed by the battery capacity. In each experiment the number of reading errors is registered. The experiment has been carried out more than ten times indicating a robust system as more than 2000 errorless readings are made.

2.3.3 Landmark navigation test

In this experiment the same mobile robot (Fig. 5) is program to drive between two points using odometry. One point is placed 1 meter in front of a landmark the other is placed at a distance of 3 m from the landmark. When the robot is at the one-meter point facing the landmark the odometry is corrected using the measured position of the landmark. This means that the landmark measurement is used to compensate for the drift of odometry coordinates. Each time the robot is at the one-meter point its position is measured. In the experiment the robot drives between the two points 100 times. The measurements show that the one-meter position of the robot stays within a circle with radius of 10 cm which means that the use of landmark position measurements is able to compensate for drift in odometry coordinates if the distance between landmarks is sufficiently small. The exact maximum distance depends on the odometry accuracy of the given robot.

3. Corridor following

Office buildings and hospitals are often dominated by long corridors so being able to drive along a corridor solves a great part of the navigation problem in these buildings. A method that uses a Hough transform with a novel discretization method to extract lines along the corridor and find the vanishing point from these is presented (Bayramoğlu. et al.,2009) Fusion of odometry data and vanishing point estimates using extended Kalman filter methods have lead to a robust visual navigation method for corridors. Experiments have shown that the robot is able to go along the corridor with lateral errors less than 3-4 cm and orientation errors less than 1-2 degrees.

3.1 Visual Pose Estimation

The low-level processing of the images consists of the detection of edge pixels and the extraction of lines from those edge pixels. The resulting lines are then classified to find a parallel set that constitutes the lines along the corners. The corresponding vanishing point, i.e., the point where the corner lines meet, is used for the classification. The classified lines are finally matched to the known width and height of the corridor to estimate the orientation and the lateral position.

3.1.1 Low-level Processing

Two feature detectors are used in consequence to prepare the data for higher level processing. First, a Canny edge detector (Canny, 1986) is used. Canny edge detector is a non-linear filter that marks pixels with a high intensity change, combined with other criteria, as edge pixels. The result is an edge image with the detected edge pixels colored white on a black background.

Lines are then extracted from the edge image using a segmented Hough transform method. The procedure starts by segmenting the image into 10x10 sub-images to increase the speed of the following steps. Line segments are extracted from these sub-images using a modified version of the Hough transform (Duda and Hart, 1972). The idea of the Hough transform is to evaluate every possible line through the image by the number of edge pixels along the line. The lines with highest support are admitted. These line segments are then traced through the image to be combined with other collinear line segments. Fig. 6. illustrates these steps.



(a) (b) (c)

Fig. 6. Steps of the low level image processing. (a) The original image taken with the robot's camera. (b) Edge image obtained from the Canny edge detector (c) Extracted lines superimposed on the original image.

3.1.2 Vanishing Point Extraction

Lines, that are parallel in 3D space, converge to a point (possibly at infinity) when their perspective projection is taken on an image. This point is called the vanishing point of that set of lines, and equivalently of their direction.

The vanishing point is useful in two ways; first, it can be used to eliminate lines that are not along its corresponding direction, since such lines are unlikely to pass through it on the image. Second, its image coordinates only depend on the camera orientation with respect to its corresponding direction; therefore, it gives a simple expression for the orientation.

The vanishing point of the corridor direction is expected to be found near the intersection of many lines. In order to find it, an intersection point for every combination of two image lines is calculated as a candidate. If there are *N* corridor lines among *M* lines in the image, there will be a cluster of *N(N+1)/2* candidates around the vanishing point as opposed to *M(M+1)/2* total candidates. This cluster is isolated from the vast number of faulty candidates by iteratively removing the furthest one from the overall center of gravity. This procedure makes use of the density of the desired cluster to discard the numerous but scattered faulty candidates. After removing most of the lines, the center of gravity of the remaining few candidates gives a good estimate of the vanishing point. Refer to Fig. 7 for the illustration of these steps.



Fig. 7. The illustration of the vanishing point extraction. The extracted lines are shown in red, the vanishing point candidates are shown in blue and the green cross at the center is the detected vanishing point.

3.1.3 Estimation of the Orientation

The image coordinates of the vanishing point is a function of, first, its corresponding direction in the scene, and second, the camera orientation. If the direction is given in the real corridor frame by the vector $\vec{v}$, then we can call its representation in the image frame $\vec{v}^{i}(\theta,\alpha,\beta)$ and it is a function of the orientation parameters θ,α,β. The image coordinates of the vanishing point are then given by;

$$x_{vp} = \frac{v_x^i}{v_z^i}, y_{vp} = \frac{v_y^i}{v_z^i} \tag{20}$$

In order to derive an expression for the orientation parameters, they are defined as follows; θ is the orientation of the mobile robot, it is the angular deviation of the front of the mobile robot from the direction of the corridor measured counter-clockwise. In the assumed setup the camera is able to rotate up or down. α is the angle of deviation of the camera from the horizon and it increases as the camera looks down. β is included for completeness and it is the camera orientation in the camera z axis, and is always equal to 0. With these definitions for the parameters the following expressions are obtained for θ, α :

$$\begin{aligned} \alpha &= \arctan\left(\frac{y_{vp} - c_y}{f'}\right) \\ \theta &= -\arctan\left(\frac{\left(x_{vp} - c_x\right)\cos\alpha}{f'}\right) \end{aligned} \tag{21}$$

Here, c_x and c_y are the image coordinates of the image center, usually half the image resolution in each direction. f' is the camera focal length in pixels.

3.1.4 Line Matching

Those image lines that are found to pass very close to the vanishing point are labeled to be along the direction of the corridor. The labelled lines need to be assigned to either of the corners, (or the line at the center of the floor for the particular corridor used in the experiments). The location of a classified line with respect to the vanishing point restricts which corner it could belong to. If the line in the image is to the upper left of the vanishing point, for instance, it can only correspond to the upper left corner if it is a correct line. The center line on the floor creates a confusion for the lower lines, each lower line is matched also to the center line to resolve this. At this point, pairs of matched image lines and real lines are obtained.

3.1.5 Estimation of the Lateral Position

Assume that the image lines are expressed in the image coordinates with the Cartesian line equation given in Eq. (22). a, b and c are the parameters defining the line and they are calculated during line extraction. Each image line – real line pair gives a constraint for the camera lateral position as given in Eq. (23).

$$ax + by = c \tag{22}$$

$$\begin{aligned} &a\left(\left(-f'\cos\theta - c_x\cos\alpha\sin\theta\right)y_d + c_x\sin\alpha z_d\right) + \\ &b\left(\left(-f'\sin\alpha\sin\theta - c_y\cos\alpha\sin\theta\right)y_d + \left(-f'\cos\alpha + c_y\sin\alpha\right)z_d\right) = c\left(-\cos\alpha\sin\theta y_d + \sin\alpha z_d\right) \end{aligned} \tag{23}$$

Here, $y_d = \left(y_{camera} - y_{line}\right)$ is the lateral distance between the real line and the camera and $z_d = \left(z_{camera} - z_{line}\right)$ is the height difference between the camera and the real line. y and z directions are defined as the axes of a right-handed coordinate frame when x points along the corridor and z points up.

The only unknown in Eq. ((23)) is the camera lateral position, therefore each matched line pair returns an estimate for it. A minority of these estimates are incorrect as the line matching step occasionally matches wrong pairs. As in the vanishing point estimation, a dense cluster of estimates are expected around the correct value. The same method of iterative furthest point removal is followed to find the correct value. To increase the robustness further, while calculating the center, the estimates are weighted according to their likelihoods based on the prior estimate.

3.2 Fusion with Dead Reckoning

The pure visual pose estimation method described so far returns a value for the orientation and the lateral position in an absolute frame. However, a single instance of such a measurement contains a considerable amount of error, especially in position (10-15 cm). The sampling rate is also low (5 fps) due to the required processing time. These problems are alleviated by fusing the visual measurements with dead reckoning, which has a high sampling rate and very high accuracy for short distances.

Probabilistic error models for both dead reckoning and visual pose estimation are required, in order to apply Bayesian fusion. The error model chosen for the dead reckoning is described by Kleeman, 2003. It is a distance driven error model where the sources of error are the uncertainty on the effective wheel separation and distances traveled by each wheel. The amount of uncertainty is assumed to be proportional to the distance traveled for a particular sample. A simple uncorrelated Gaussian white noise is assumed for the visual measurements.

An extended Kalman filter(EKF) is used to perform the fusion. The time update step of the EKF is the familiar dead reckoning pose update with the mentioned distance driven error model. The update is performed for every wheel encoder sample until a visual measurement arrives. The measurement update step of the EKF is applied when it arrives. The assumed measurement model is given as follows:

$$\begin{bmatrix} \theta_v(k) \\ y_v(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta(k) \\ x(k) \\ y(k) \end{bmatrix} + \vec{v}(k) \tag{24}$$

Where $\vec{v}(k)$ is an uncorrelated Gaussian white noise with a covariance matrix calculated empirically.

3.3 Observed Performance

The performance of the method is evaluated by comparing its measurements with the actual robot pose. The visual pose estimation is calculated to be accurate within 1-2 degrees of error in the orientation. Since it is hard to measure the robot orientation to this accuracy, the performance is evaluated based on the error in the lateral position.

Single frame visual estimation is evaluated for performance first. Fig. 8 contains four interesting cases. In Fig. 8 (a), part of the corridor is obscured by a person and a door, but the estimation is not effected at all with an error of 2.5cm. Fig. 8 (b) displays a case where only the left wall is visible, but the method still succeeds with an error of 0.2cm. Fig. 8 (c) shows an extreme case. Even though the end of the corridor is not visible, the algorithm performs well with an error of 0.9cm. Fig. 8 (d) shows a weakness of the method. The image has no particular difficulty, but the measurement has 11.8cm error. The final case occurs rarely but it suggests the use of a higher standard deviation for the assumed measured error.

The second step is the evaluation of the performance after fusion with dead reckoning. The navigation task is moving backwards and forwards at the center of the corridor. Fig. 9 contains three sets of data plotted together. The red curve is the overall pose estimation after sensor fusion. The green dots are the visual estimations alone. Finally, the blue curve is a collection of absolute measurements taken with a ruler. The error is observed to remain below 3cm in this experiment.



(a) Only two corners are detected



(b) The view is partially blocked



(c) Moving towards the wall



(d) This case has high error

Fig. 8. Images with special properties illustrating the strengths and the weaknesses of the pure visual estimation. (a), (b) and (c) illustrate difficult cases successfully measured while (d) show a case with a convenient image with a high measurement error.

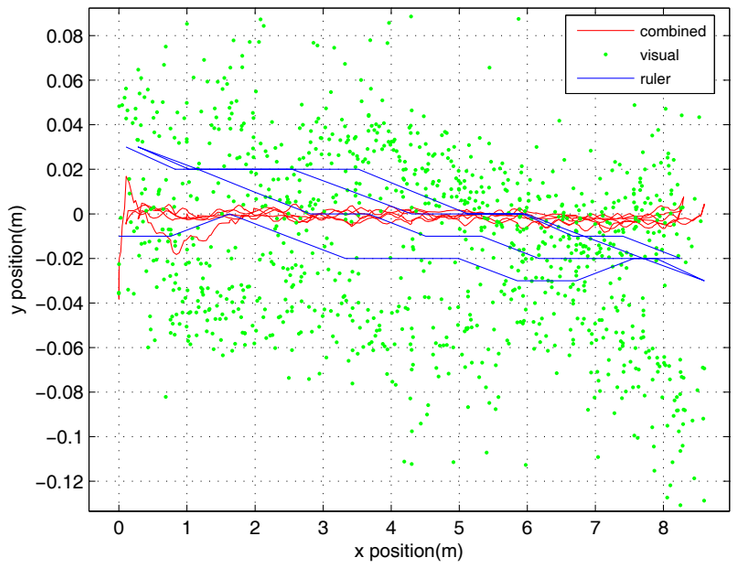


Fig. 9. Position data from various sources for system performance illustration.

4. Laser and vision based road following

Many semi structured environments with gravel paths and asphalt roads exist e.g. public parks. A method for navigating in such environments is presented. A slightly tilted laser scanner is used for classification of the area in front of the vehicle into traversable and non-traversable segments and to detect relevant obstacles within the coverage area. The laser is supplemented with a vision sensor capable of finding the outline of the traversable road beyond the laser scanner range (Fig. 10). The detected features – traversable segments, obstacles and road outline- are then fused into a feature map directly used for path decisions. The method has been experimentally verified by several 3 km runs in a nature park having both gravel roads and asphalt roads.

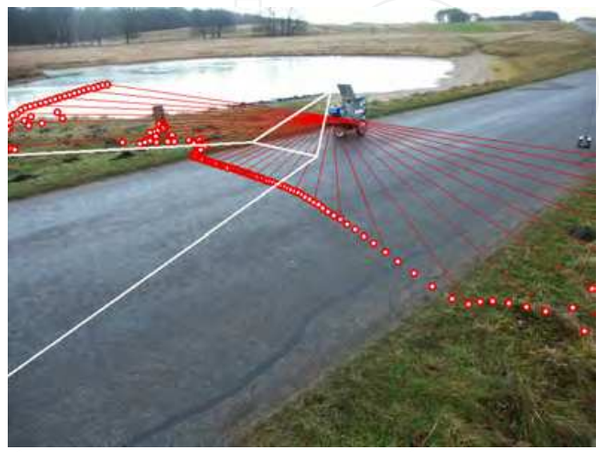


Fig. 10. The robot with laser scanner measurements and camera coverage

4.1 Related work

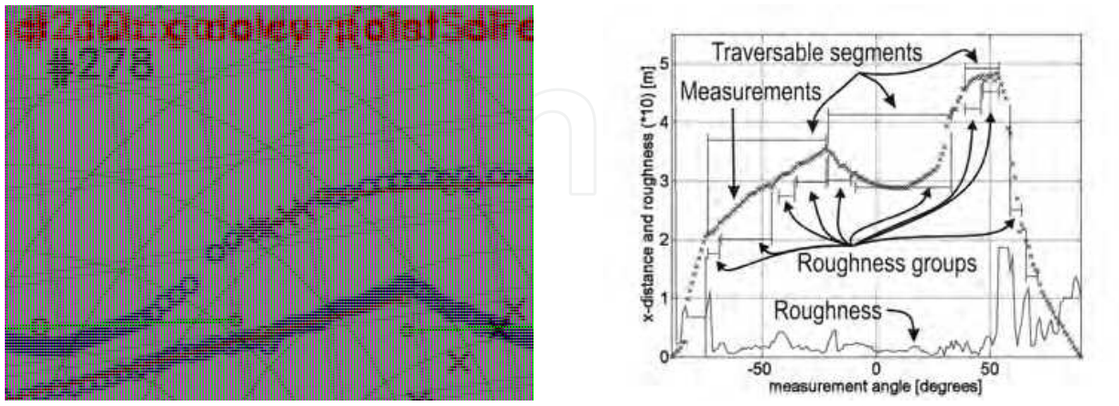
Current work in the area tends to focus on using 3D laser scanners or a combination of 3D laser scanners and vision. Using 3D laser scanner solutions has been proposed by Vandapel et al. (2004) by transforming point clouds into linear features, surfaces, and scatter. These were classified by using a Bayesian filter based on a manually classified training set.
Identification of navigable terrain using a 3D laser scanner by checking if all height measurements in the vicinity of a range reading had less than a few centimeters deviation is described in Montemerlo & Thrun (2004)
An algorithm that distinguished compressible grass (which is traversable) from obstacles such as rocks using spatial coherence techniques with an omni-directional single line laser is described in (Macedo et. al.,2000).
A method for detection and tracking the vertical edges of the curbstones bordering the road, using a 2D laser scanner, described in Wijesoma et. Al. (2004) is a way of indirect road detection.
Detection of borders or obstacles using laser scanners is often used both indoors and in populated outdoor environments, and is the favored method when the purpose includes map building, as in Guivant et al. (2001) and Klöör et al. (1993).
Detection of nontraversasble terrain shapes like steps using laser scanner for planetary exploration is described in Henriksen & Krotkov (1997)

The DARPA Grand Challenge 2004 race demonstrated the difficulties in employing road following and obstacle avoidance for autonomous vehicles (Urmson et. al, 2004).
This situation seems to be improved in the 2005 version of the race, where five autonomous vehicles completed the 212~km planned route. The winning team from Stanford perceived the environment through four laser range finders, a radar system, and a monocular vision system. Other teams, like the gray team Trepagnier et al. (2005) also use laser scanners as the main sensor for traversability sensing supplemented by (stereo) vision.
The solution of the winning team in 2005 is described in (Thrun et al., 2006); a 2D laser scanner detects traversable road based on the vertical distance between measurements, this solution is combined with vision and radar for longer range detections.

4.2 Terrain classification from laser scanner

A slightly tilted laser obtains scans in front of the robot (Fig. 10**).** The assumption is that the terrain seen by the laser scanner can be divided into three different classes $C = \{C_t\ (traversable), C_n (not\ traversable), C_\varnothing (invalid\ data)\}$ and that this can be done by mapping function $M_{CF} : F \rightarrow C$. Here F is a set of features extracted from single laserscans:
$F = \{H_h\ raw height, F_\sigma\ roughness, F_z\ stepsize, F_c\ curvature, F_t\ slope, F_w\ width\}$
The roughness of data in a 2D laser scan is defined as the square root of the local vatiance of the distance to reflections along the scan. A roughness value is calculated as deviation from a fitted line for measurements converging approx. a wheel base distance (0.45 m), to emphasize terrain variation with a spatial period shorter than this distance. The roughness feature function F_σ divides the measurements into groups based on this roughness value, these groups are then combined and filtered based on the remaining feature functions. Each of these functions increases the probability that the measurements are correctly classified. The method is described in detail in (Andersen et al.,2006b)

An example of the obtained classification is shown in Fig. 11 where a narrow gravelled road is crossed by a horse track. The road and the horse track are first divided into a number of roughness groups as shown in Fig. 2b, these are then filtered down to three traversable segments, one for the road (in the middle) and one each side from the horse track.



(a) (b)

Fig. 11. Data from a graveled road crossed by a horse track

The laser scanner measurements are shown in (a) as circles (traversable) and crosses (not traversable), the rough grass on the road edges before the horse track is just visible left and right of the robot. The road is the area with the high profile (about 15 cm higher at the center). On both side are relative flat areas from the horse track. The segmentation into roughness groups and traversable segments are shown in (b).

4.3 Road outline from vision

As seen on Fig. 10 the laserscan overlaps the camera image.The method (Andersen et al.,2006a) estimates the outline of the road by analyzing the image, based on a seed area in the image classified as traversable by the laserscanner. The main features describing the road are its homogeneity. But there may be variation in the visual expression due to e.g. shadows, sunlight, specular reflections, surface granularity, flaws, partially wet or dry surface and minor obstacles like leaves.

The road detection is therefore based on two features: the chromaticity $\mathbf{C}$ and the intensity gradient ∇I . The chromaticity is colour stripped from intensity as shown in Eq. (25) based on a RGB image.

$$\mathbf{c} = \begin{bmatrix} c_{\text{red}} \\ c_{\text{green}} \end{bmatrix} = \begin{bmatrix} r/(r+g+b) \\ g/(r+g+b) \end{bmatrix} \tag{25}$$

Each pixel $H_{i,j}$ is classified into class R = {road, not road} based on these features. The R_{road} classification is defined as

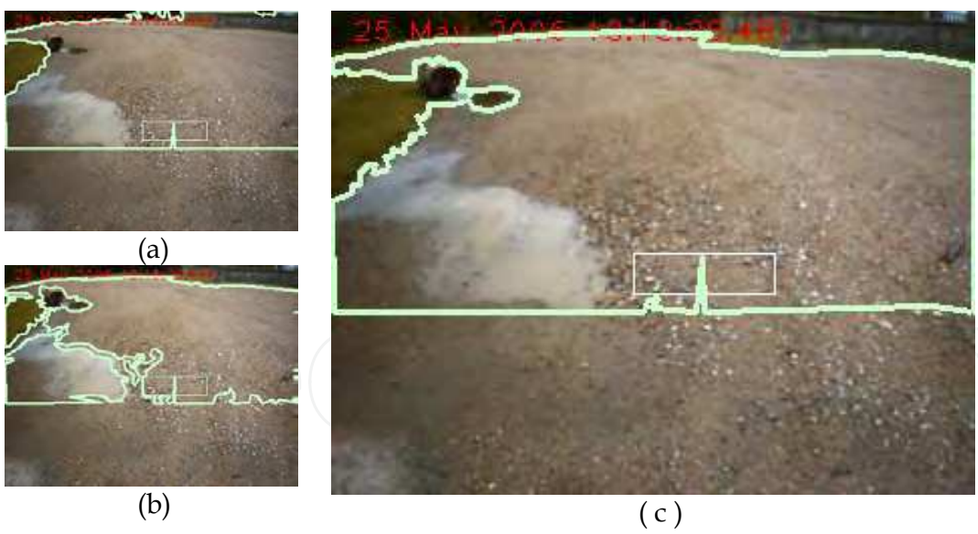
$$\mathsf{R}_{\text{road}}(H_{i,j}) = \left\{ H_{i,j} \left| \begin{matrix} P_c(\mathbf{C}(H_{i,j})) + \\ P_e(\nabla I(H_{i,j})) \\ > K_{\text{limit}} \end{matrix} \right. \right\} \tag{26}$$

where $P_c(\cdot)$ Eq. (27) is a probability function the based on the Mahalanobi distance of the chromaticity relative to the seed area. $P_e(\cdot)$ Eq. (28) is based on the intensity gradient, calculated using a Sobel operator. The Sobel kernel size is selected as appropriate for the position in the image, i.e. the lower in the image the larger the kernel (3x3 at the top and 5x5 pixels at the bottom for the used 320 X 240 image resolution).

$$P_C(i,j) = \left(1 + w_C(i)(\mathbf{c}_{i,j} - \bar{\mathbf{c}}w)^T \mathbf{Q}^{-1}(\mathbf{c}_{i,j} - \bar{\mathbf{c}}w)\right)^{-1} \tag{27}$$

$$P_e(i,j) = \left(1 + w_e\left[\left|\frac{\partial I(i,j)}{\partial i}\right|^2 + \left|\frac{\partial I(i,j)}{\partial j}\right|^2\right]\right)^{-1} \tag{28}$$

$\mathbf{Q}$ is the chromaticity covariance for the seed area. The $w_c(i)$ and w_e are weight factors.
An example of the capabilities of the filter functions is shown in Fig. 12. Only the pixels at the road contour are evaluated, i.e. from the seed area pixels are tested towards the image edge or road border, the road border is then followed back to the seed area.



(a) (b) (c)

Fig. 12. Road outline extraction based on chromaticity (a), on gradient detection (b) and combined (c). In the top left corner there is a stone fence, this is not distinguished from the gravel road surface using the chromaticity filter in (a). The gradient filter (b) makes a border to the pit (bottom left). The combined filter (c) outlines the traversable area as desired. The seed area classified by the laser scanner is shown as a (thin) rectangle. The part of the image below the seed area is not analyzed.

4.4 Fusion of laser and vision data

A feature map representation is adapted for sensor fusion and navigation planning. The detected features are the traversable segments from the laser scanner covering ranges up to about 2.5 m in front of the robot, the vision based road outline from about 2 m and forward, and the obstacles detected from the laser scanner data.

Each traversable segment s_j^k extracted by the laserscanner in the most recent scan k is correlated with traversable segments from previous scans s^{k-i}, forming a set of traversable corridors B as shown in Eq. (29) correlation exists if the segments overlap with more than a robot width.

$$B_i = \{S_{a0}^k, S_{a1}^{k-1}, S_{a2}^{k-2}, \cdots, S_{aN}^{k-N}\} \tag{29}$$

where S_a^{k-i} is the a th traversable segment found in scan $k-i$.

This corridor of traversable segments gets extended beyond the range of the laserscanner using the road outline from the vision sensor. Intersection lines S^{k+v} (perpendicular to the current robot heading) at increasing intervals are used to extend the laser scanner corridors, as shown in Eq. (30)

$$B_i = \{S_{b1}^{v1}, S_{b2}^{v2}, \cdots, S_{bM}^{vM}, S_{a0}^k, S_{a1}^{k-1}, S_{a2}^{k-2}, \cdots, S_{aN}^{k-N}\} \tag{30}$$

where S_b^{vn} is the b th intersection segment of intersection line n inside the estimated road utline. See example in Fig. 15

A number of such corridors may exist, e.g. left and right of obstacles, left and right in road forks or as a result of erroneous classification from the laser scanner or form the vision. A navigation route is planned along each of these corrodors considering the obstacles, current navigation objectives and the robot dynamics. The best route is qualified using a number of parameters including corridor statistics. If the vision is unable to estimate a usable road outline then the laser scanner data is used only.

4.5 Experimental results

The method is tested primarily on several runs of a 3 km route in a national park. The navigation is guided by a script specifying how to follow the roads and for how long. At the junctions the script guides the robot in an approximate direction until the next road is found. GPS is used sparsely to determine when a road section is about to end. Fig. 13 shows the road width detection from two road sections, a homogene asphalt road (a) and a 4 m wide gravelled road (b). The weather conditions were overcast with mild showers. The road width is estimated based on the available data EQ.(30) at time of manoeuvre decision. The vision based road width estimate is in the plane of the robot base, and as the road is mostly convex curved, the road width in the projected plane is narrower than in reality.

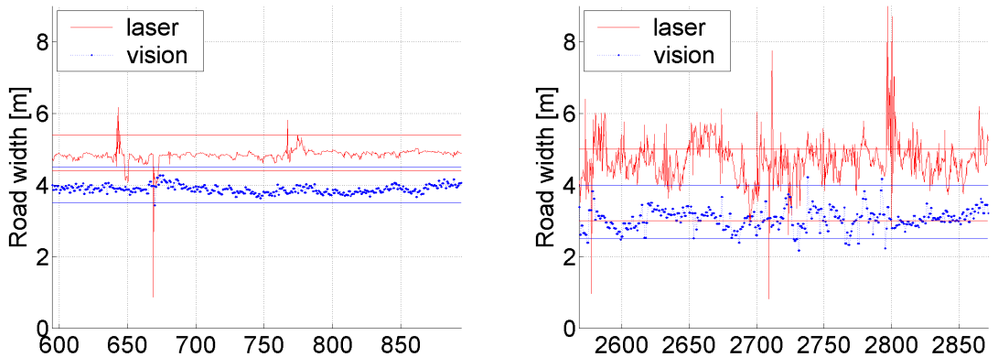


Fig. 13. Road width estimation based on laser scanner (upper (red) curve) and vision based (dotted blue). A section from a 4.9 m wide asphalt road (a), and a $\approx$4 m wide gravelled road (b). The vision based road width is estimated in the plane of the robot base, and as the road is (convex) curved, the estimated road width is narrower. The asphalt road (a) is easy for both the laser scanner and the vision sensor. The gravelled road (b) is detected with a higher uncertainty, especially by the laser scanner.

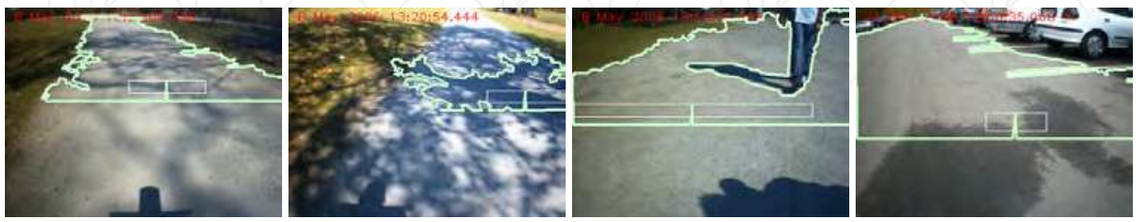
The road width estimate and the road width stability can be taken as a performance measure of the vision and laser scanner sensors. The road width estimates are summarized in Table 3 for the laser scanner and vision sensor respectively. The laser scanner based estimate shows the correct road width in most cases, with a tendency to include the road shoulder at times, especially for the gravelled road where the roughness difference between the road and the road shoulder is limited

	true	laser based				vision based				Fig.
Road segment	width [m]	mean	σ	failed	N	mean	σ	failed	N	ref
Asphalt gray/wet	4.9	4.8	0.19	0%	1000	3.9	0.11	0%	291	Fig. 13a
Graveled gray/wet	4	4.7	0.62	0%	1000	3.1	0.30	1%	276	Fig. 13b
Asphalt gray/wet	3--4	3.5	0.63	0%	890	2.8	0.36	2%	224	
Asphalt sun/shade	3--4	3.3	0.46	0%	482	2.8	0.53	16%	79	

Table 3. Road width estimate summary from the data shown in Fig. 4. On the asphalt roads the laser scanner based estimate are with good precision, on the gravelled road the flat road shoulder widens the average road width estimate and makes the width estimate uncertain (higher σ). The last two rows are from the same road segment but in different weather conditions. N is the number of measurements.

The vision part shows a narrower road width estimate, as expected. Additionally the vision tends to estimate a to narrow road in case of shadows. The last two rows in Table 3 are from

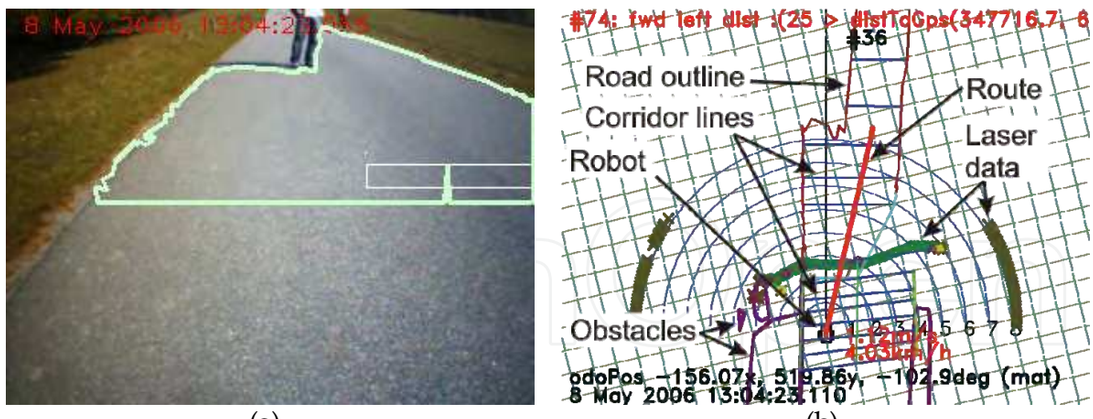
a road segment that is partially below large trees, and here the road outline estimates failed in 16% of the measurements on the day with sunshine, compared to just 2% in gray and wet weather condition.
The vision based road outline detector does not cope well with focused shadow lines as shown in Fig. 14b and c, nor with painted road markings as in Fig. 14d. Unfocused shadows as in Fig. 14a are handled reasonably well.Wet and dry parts of the road are much less of a problem for the road outline detector as shown in Fig. 14d.



(a) (b) (c) (d)

Fig. 14. Shadows and road markings at the limits of the vision sensor capabilities. Unfocused shadows like in (a) are handled reasonably well, but if the shadows are more focused as in (b) the result is of little or no use. Hard shadows as in (c) and road markings as the white markings in a parking lot in (d) are handled as obstacles.

When the road outline is limited by obstacles as in the situation shown in Fig. 15a, the route possibilities in the feature map (Fig. 15b) will be limited correspondingly, and the result is an obstacle avoidance route initiated at an early stage. The pedestrian can follow the robot intentions as soon as the obstacle avoidance route is initiated, and thus limit potential conflict situations.



(a) (b)

Fig. 15. The pedestrian (and his shadow) is reducing the road outline (a) and an obstacle avoidance route is planned (as shown in (b)) long before the obstacle is seen by the laser scanner.

5. Conclusion

Three methods for support of visual navigation have been presented: Artificial visual landmarks, corridor following using vanishing point, and road following using terrain classification based on data fusion of laser scanner and vision. All the methods have been verified experimentally so that their usefulness in real systems is demonstrated. It has been shown that a combination of artificial landmarks and odometry will be able to limit the navigation error to a given level if the landmarks are placed with a sufficiently small distance. Using both landmarks and corridor following the same navigation accuracy may be obtained with much fewer landmarks thus enhancing the usability of the system.
The results for terrain classification show that fusing data from laser and vision gives a good foundation for path and road-following for outdoor robots. This may be used for service robots that are operating in e.g. public parks and gardens.

6. References

Andersen, J. C..; Blas M. R.; Ravn, O.; Andersen, N. A.; Blanke, M.(2006a). Traversable terrain classification for autonomous robots using single 2D laser scans. *Integrated Computer-aided engineering,* Vol.13, No.3,pp. 223-232, ISSN 1069-2509

Andersen J. C.(2006b). *Mobile Robot Navigation. ,PhD. Thesis* , Technical University of Denmark, ISBN 87-91184-64-9, Kgs. Lyngby Denmark

Andersen, J. C..; Andersen, N. A.; Ravn O.(2008). Vision Assisted Laser Navigation for Autonomous Robot *Experimental Robotics,* Star 39, Khatib,O.; Kumar, V. & Rus D. (Eds.), pp. 111-120,Springer-Verlag Berlin Heidelberg

Bais, A & Sablatnig R. (2006) Landmark based global self-localization of mobile soccer robots, In: COMPUTER VISION-ACCV 2006,PT II, Lecture Notes In Computer Sc ience, , pp. 842-851, Springer-Verlag Berlin, ISBN 3-540-31244-7, Berlin Germany

Bayramoğlu, E.; Andersen,N.A.; Poulsen, N.K.; Andersen, J. C.; Ravn, O. (2009). Mobile Robot Navigation in a Corridor Using Visual Odometry. Procedings of 14th Int. Conf. In Advanced Robotics, id. 58, June 22-26, 2009, Munich, Germany

Canny, J. (1986) A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679--98, 1986.

Duda ,R.O. & Hart, P.E.(1972). Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11--15, 1972.

Guivant, J. E., Masson, F. R. & Nebot, E. M. (2001), Optimization of the simultaneous localization and map-building algorithm for real-time implementation, *IEEE Trans. on Robotics and Automation*, 17 (3),pp. 242—257, The University of Sidney, Australia.

Henriksen, L. & Krotkov, E. (1997), Natural terrain hazard detection with a laser rangefinder, *IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 968--973.

Kabuka, M.& Arenas, A. (1987). Position verification of a mobile robot using standard pattern. *Journal of Robotics and Automation,* Vol.3, No.6,pp. 505-516, ISSN 1042296x

Kleeman, L.(2003). Advanced sonar and odometry error modeling for simultaneous localisation and map building. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1:699--704, 2003.

Klöör, P.L., Lundquist, P., Ohlsson, P., Nygårds, J. & Wernersson, A. (1993), Change detection in natural scenes using laser range measurements from a mobile robot, *Proceedings of 1st IFACInternational Workshop on Intelligent Autonomous Vehicles*, IFAC, University of Southampton, pp. 71--76.

Lin, C.C. & Tummala, R. L.. (1997). Mobile robot navigation using artificial landmarks. *Journal of Robotic Systems,* Vol.14, No.2,pp. 93-106, ISSN 0741-2223

Macedo,J.; Matthies, L.; & Manduchi, R.(2000). Ladar.based discriminations of grass from obstacles for autonomous navigation. *Experimental Robotics VII,* Proceedings ISER 2000, Waikiki, Hawaii, Springer ,pp. 111-120

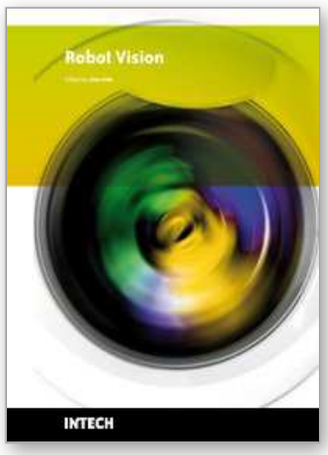
Trepagnier, P. G., Kinney, P. M., Nagel, J. E., Doner, M. T. & Pearce (2005), *Team gray techical paper*, Technical report, Gray & Company Inc.

Urmson, C., Anhalt, J., Clark, M., Galatali, T., Gonzalez, J. P.,Gutierrez, A., Harbaugh, S., Johnson-Roberson, M., Kato, H., Koon, P. L., Peterson, K., Smith, B.K., Spiker, S., Tryzelaar, E. & Whittaker, W. R.L. (2004), High speed navigation of unrehearsed terrain: Red team technology for grand challenge 2004*, Technical Report CMU-RI-TR-04-37*, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Vandapel, N., Huber, D., Kapuria, A. & Hebert, M. (2004), Natural terrain classification using 3-d ladar data, *Robotics and Automation Proceedings. ICRA '04. 2004 IEEE International Conference on*', IEEE, pp. 5117--5122.

Wijesoma, W., Kodagoda, K. & Balasuriya, A. (2004), `Road-boundary detection and tracking using ladar sensing*, Robotics and Automation, IEEE Transactions on*, pp. 456-464.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C.,Palatucci, M., Pratt, V. & Pascal,S. (2006), Stanley: The robot that won the DARPA grand challenge, *http://cs.stanford.edu/group/roadrunner*



Robot Vision

Edited by Ales Ude

ISBN 978-953-307-077-3
Hard cover, 614 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

The purpose of robot vision is to enable robots to perceive the external world in order to perform a large range of tasks such as navigation, visual servoing for object tracking and manipulation, object recognition and categorization, surveillance, and higher-level decision-making. Among different perceptual modalities, vision is arguably the most important one. It is therefore an essential building block of a cognitive robot. This book presents a snapshot of the wide variety of work in robot vision that is currently going on in different parts of the world.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Nils Axel Andersen, Jens Christian Andersen, Enis Bayramoglu and Ole Ravn (2010). Visual Navigation for Mobile Robots, Robot Vision, Ales Ude (Ed.), ISBN: 978-953-307-077-3, InTech, Available from: http://www.intechopen.com/books/robot-vision/visual-navigation-for-mobile-robots

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen