COMBINING A PRIORI KNOWLEDGE AND SENSOR INFORMATION FOR UPDATING THE GLOBAL POSITION OF AN AUTONOMOUS VEHICLE

Zoran Zivkovic†, Albert Schoute‡ and Ferdinand van der Heijden†

†University of Twente, Laboratory for Measurement and Instrumentation
Dept. Electrical Engineering, P.O. Box 217, 7500 AE Enschede, The Netherlands
email: Z.Zivkovic@el.utwente.nl

‡ University of Twente, Distributed and Embedded Systems
Dept. Computer Science, P.O. Box 217, 7500 AE Enschede, The Netherlands

Abstract

The problem of updating the global position of an autonomous vehicle is considered. An iterative procedure is proposed to fit a map to a set of noisy point measurements. The procedure is inspired by a non-parametric procedure for probability density function mode searching. We show how this could be used to solve the global position update problem in an efficient and robust way. The simple procedure combines the a priori knowledge from the known map of the environment, the extracted information from calibrated camera images and the information from the vehicle wheels rotation measurements. Because it is very simple the procedure is appropriate to be performed in real-time.

1 Introduction

An autonomous vehicle should be able to perform a given tasks in a dynamic environment. The vehicle should be able to sense the state of the environment in order to be able to perform its actions.

The architecture of an experimental autonomous vehicle is described in the next section. For executing the various tasks the vehicle needs to navigate through the complex office environment. Obstacles should be detected and avoided, but it is also essential to know the vehicle location in the global office map. With known initial position and wheels rotation measurements the position at some later time can be reconstructed by odometry. The problem is that the errors are accumulated and the uncertainty grows with time. Additional information about the environment must be used to update the position. We assume that a known map of the static layout of the office environment is available. How to use this a priori knowledge and the information from a camera sensor in a simple, robust and efficient way is the main topic of this paper. A robust map matching technique is proposed in this paper. Then, we demonstrate how this could be used to solve the global autonomous vehicle position update problem. Finally, some preliminary experiments are shown.

2 Architecture

An experimental low-budget, PC-based, robot vehicle was developed as test-bed for autonomous system development. The scheme of the robot is shown in figure 1. The vehicle is equipped with a number of different sensors. A number of ultrasonic sensors are mounted on the sides of the robot. They are basically used for close object detection and obstacle avoidance. Tachometers give the information about the wheels rotation speeds. As an important information source a wide angle camera is mounted on the front side of the robot. A common scene as viewed by the robot's camera is shown in figure 4. The wide angle lenses introduce large distortions. The camera is calibrated using the simple procedure presented in (Zhang, 2000). More details about the vehicle architecture can be found in (Zivkovic, 1998).

The objective of the test-bed is to explore techniques by which the robot vehicle can navigate in an office environment using natural properties of the building. Although we allow the control program to exploit pre-knowledge of the static building features, we also have to cope with the actual situation in a robust and reactive manner. The control architecture for autonomous agents have been previously proposed by many researchers (Muller, 1997). The control architecture we used is *behaviour-based*. Detailed information about the framework that was developed is given in (Schoute, 2001).

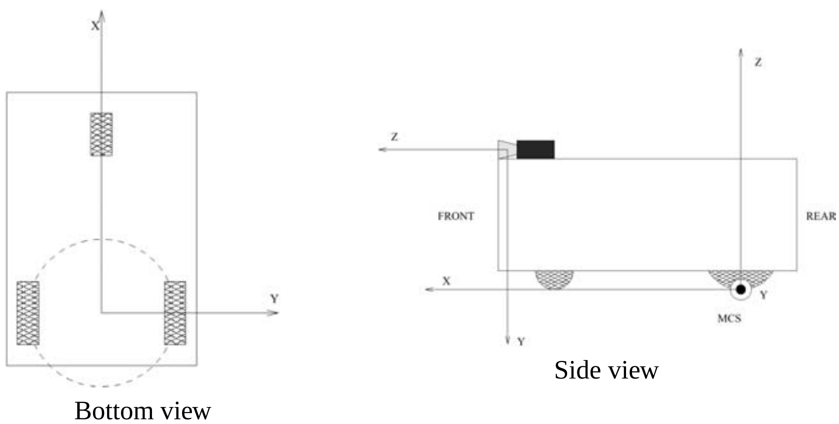


Figure 1: The robot scheme - with the vehicle coordinate frame (MCS) and the camera coordinate system (CCS)

3 A single point matching

In this section we analyze the simple problem of estimating the point position given a set of noisy point position measurements.

3.1 Robust point matching

Lets assume that we are measuring a 2D position $\overrightarrow{x} = [\ x \quad y\ ]^T$ of a point with high noise level and a huge number of outliers as shown in the figure 2. Lets denote a set of the point position measurements with $\mathcal{Z} = \{\overrightarrow{z}_1, ..., \overrightarrow{z}_N\}$. The problem is to estimate the 2D position of the point $\widehat{\overrightarrow{x}}$. One might naively assume this could easily be done by finding the point that minimizes the distances from all other points:

$$\widehat{\overrightarrow{x}} = \arg\min_{\overrightarrow{x}} \sum_{n=1}^{N} \|\overrightarrow{x} - \overrightarrow{z}_n\|^2 \tag{1}$$

However, due to the huge number of outliers this would lead to poor results.

A solution, closely related to the well known robust statistics methods described in detail in (Huber, 1981) and (Hampel, Ronchetti, Rousseeuw and Stahel, 1986), is to use a weight function $k_W(x)$ that limits the influence of the far away points (possible outliers):

$$\widehat{\overrightarrow{x}} = \arg\min_{\overrightarrow{x}} \sum_{n=1}^{N} k_W\left(\left\|\frac{\overrightarrow{x} - \overrightarrow{z}_n}{h}\right\|^2\right) \tag{2}$$

where h is the parameter that controls the break point distance after which the points $\overrightarrow{z}_n$ are considered to be outliers. An example function $k_W(x)$ could be:

$$k_W(x) = \begin{cases} x & \text{if } x < 1 \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

There are various ways for solving (2). In the next section we reformulate the problem in order to get a simple and well behaved solution.

3.2 Sample mean shift

It is interesting for the future analysis to reformulate the problem by choosing an equivalent weight function:

$$k(x) = c(1 - k_W(x)) = \begin{cases} c(1-x) & \text{if } x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where c is an appropriately chosen constant described later. This turns the previously defined minimization problem (2) to a maximization problem:

$$\widehat{\overrightarrow{x}} = \arg\max_{\overrightarrow{x}} \sum_{n=1}^{N} k\left(\left\|\frac{\overrightarrow{x} - \overrightarrow{z}_n}{h}\right\|^2\right) \tag{5}$$

We could regard the point position measurements $\mathcal{Z} = \{\overrightarrow{z}_1, ..., \overrightarrow{z}_N\}$ as a random sample from some underlying probability density function. The value of the underlying probability density function at point $\overrightarrow{x}$ is denoted as $f_z(\overrightarrow{x})$. The Parzen kernel estimate of the probability density function at point $\overrightarrow{x}$ using the previously defined kernel (4) (also known as Epanechnikov kernel) looks like:

$$\widehat{f}_z(\overrightarrow{x}) = \frac{1}{Nh^d} \sum_{n=1}^{N} k\left(\left\|\frac{\overrightarrow{x} - \overrightarrow{z}_n}{h}\right\|^2\right) \tag{6}$$

where d is the dimension of $\vec{z}_n$ (and $\vec{x}$ also) and the constant c from (4) should be appropriately chosen so that $\hat{f}_z(\vec{x})$ integrates to one.

We observe that the probability density function approximation $\hat{f}_z(\vec{x})$ has the same form as (5). Therefore, the robust point matching problem is equivalent to the problem of finding the maximum of $\hat{f}_z(\vec{x})$ (*mode finding*).

In many situations the most of the practically relevant information in a random sample would be obtained if we could estimate the mode of the underlying probability density function. In the previous analysis it was demonstrated that starting from a simple point matching problem and using some standard and natural heuristics to solve it, we arrive to the same conclusion. The probability density mode estimation is a well known subject in the probability and statistics area (Parzen, 1962). A simple iterative local mode search method is described in next (see also (Fukunaga and Hostetler, 1975) and (Tukey, 1976)).

The derivatives of the approximate probability density function can be easily computed. The first order derivative is (gradient vector):

$$\begin{aligned} \vec{g}(\vec{x}) &= \frac{\partial}{\partial \vec{x}} \hat{f}_z(\vec{x}) \\ &= \frac{2}{Nh^{d+2}} \sum_{n=1}^{N} (\vec{x} - \vec{z}_n) k' \left(\left\| \frac{\vec{x} - \vec{z}_n}{h} \right\|^2 \right) \end{aligned} \tag{7}$$

and the second order derivative (Hessian matrix):

$$\begin{aligned} H(\vec{x}) = \frac{\partial^2}{\partial \vec{x}^2} \hat{f}_z(\vec{x}) = \frac{2}{Nh^{d+2}} \Bigg\{ \sum_{n=1}^{N} I k' \left(\left\| \frac{\vec{x} - \vec{z}_n}{h} \right\|^2 \right) + \\ +2 \left(\frac{\vec{x} - \vec{z}_n}{h} \right) \left(\frac{\vec{x} - \vec{z}_n}{h} \right)^T k'' \left(\left\| \frac{\vec{x} - \vec{z}_n}{h} \right\|^2 \right) \Bigg\} \end{aligned} \tag{8}$$

where k' and k'' are the first and the second order derivatives of the kernel used in the probability density function estimation.

A Newton type iterative local maxima search can be written as:

$$\hat{\vec{x}}_{j+1} = \hat{\vec{x}}_j - \vec{g}(\hat{\vec{x}}_j) H(\hat{\vec{x}}_j)^{-1} \tag{9}$$

where $\hat{\vec{x}}_j$ is the estimated position of the maximum at j-th iteration.

The behaviour of the Newton type iterations depends on the underlying function. There are many extensions that can guarantee the convergence. In our special case keeping only the first term in (8) we get the well known mean shift algorithm:

$$\hat{\vec{x}}_{j+1} = \frac{\sum_{n=1}^{N} \vec{z}_n k' \left(\left\| \frac{\hat{\vec{x}}_j - \vec{z}_n}{h} \right\|^2 \right)}{\sum_{n=1}^{N} k' \left(\left\| \frac{\hat{\vec{x}}_j - \vec{z}_n}{h} \right\|^2 \right)} \tag{10}$$

It can be shown that the mean shift iterations (10) are globally convergent if the kernel function k is convex and monotonic decreasing (see (Comaniciu, Ramesh and Meer, 2000) and (Comaniciu and Meer, 2002)).

An example generated using uniformly distributed outliers is shown in figure 2. Starting from the mean of the data the mean shift converges to the mode of the distribution.

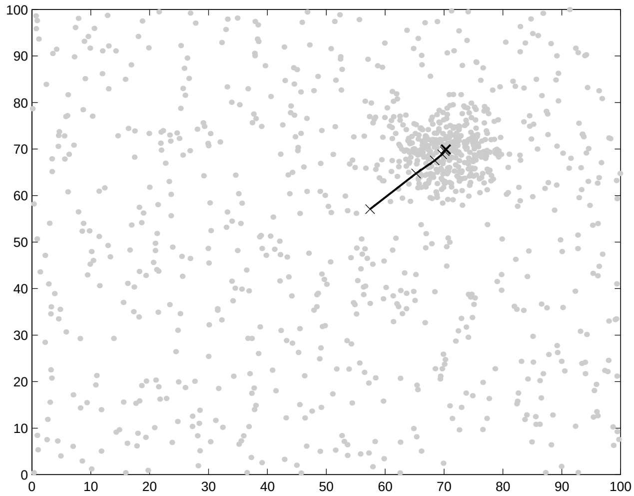


Figure 2: Mean shift mode search with 70% outliers

4 Map matching

We assume that we have a global map of the environment. It consists of M positions of the corners $\mathcal{M} = \{\vec{x}_1, ..., \vec{x}_M\}$ in the global 2D coordinate system (GCS). Lets assume that we measured the positions $\mathcal{Z} = \{\vec{z}_1, ..., \vec{z}_N\}$ of some of the corners with lots of noise and outliers and in another coordinate system GCS' that is translated by $\vec{t}$ and rotated for an angle α with respect to the GCS. The problem of map matching is to estimate the translation and rotation $\widehat{\vec{y}} = [\ \widehat{\vec{t}}^T \quad \widehat{\alpha}\ ]^T$ between the two coordinate systems from the given noisy measurements. A natural way of formulating the problem is similar to the previously described point matching problem. Each map point $\vec{x}_m$ is matched to the nearby points as described by the weight function k. This can be written as:

$$\widehat{\vec{y}} = \arg\max_{\vec{y}} \sum_{m=1}^{M} \sum_{n=1}^{N} k\left(\left\| \frac{p(\vec{x}_m, \vec{y}) - \vec{z}_n}{h} \right\|^2 \right) \tag{11}$$

where

$$p(\vec{x}_m, \vec{y}) = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \vec{x}_m + \vec{t} \tag{12}$$

is the transformation between the two coordinate systems.

Following the same logic as in the previous point matching problem we get the following iterative procedure:

$$\vec{y}_{j+1} = \vec{y}_j + d\vec{y}_j \tag{13}$$

$$d\vec{y}_j = \left\{ \sum_{m=1}^{M} \sum_{n=1}^{N} p'(\vec{x}_m, \vec{y}_j)^T (p(\vec{x}_m, \vec{y}_j) - \vec{z}_n) k' \left(\left\| \frac{p(\vec{x}_m, \vec{y}_j) - \vec{z}_n}{h} \right\|^2 \right) \right\} \cdot \left\{ \sum_{m=1}^{M} \sum_{n=1}^{N} p'(\vec{x}_m, \vec{y}_j)^T k' \left(\left\| \frac{p(\vec{x}_m, \vec{y}_j) - \vec{z}_n}{h} \right\|^2 \right) p'(\vec{x}_m, \vec{y}_j) \right\}^{-1} \quad (14)$$

where

$$p'(\vec{x}_m, \vec{y}) = \frac{\partial}{\partial \vec{y}} p(\vec{x}_m, \vec{y}) = \begin{bmatrix} 1 & 0 & [\, -\sin(\alpha) \quad \cos(\alpha) \,]\vec{x}_m \\ 0 & 1 & [\, -\cos(\alpha) \quad -\sin(\alpha) \,]\vec{x}_m \end{bmatrix} \quad (15)$$

is the derivative of the function $p(\vec{x}_m, \vec{y})$ (a 2×3 matrix). The final value of the iterative process $\vec{y}_j$ we use as the estimated $\widehat{\vec{y}}$.

An example on simulated data is shown in figure 3. The outliers are uniform distributed and the point measurements normal distributed. There are 70% outliers. Three points from the map are observed and can be seen in the figure as three distinct point clusters. Starting from some nearby pose the algorithm converges to the true pose after few iterations. The initial and the final pose are presented in the figure. For the real data example see figure 5.

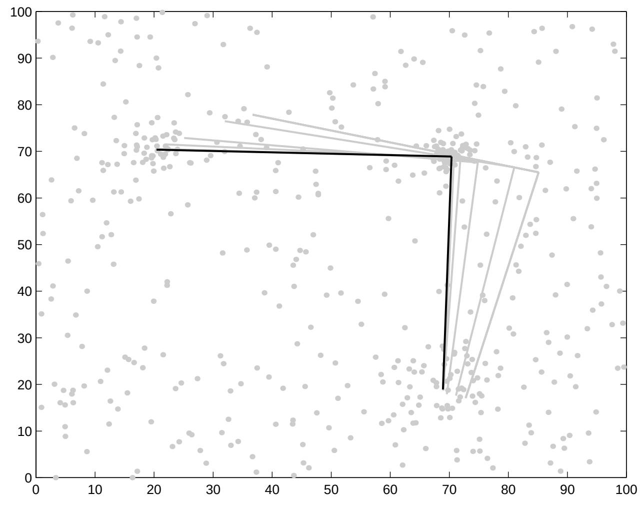


Figure 3: Matching three map corners

5 Global position update

In the previous sections we proposed a method for the robust map fitting. The problem of updating the global autonomous vehicle position is the main goal of this paper. We describe here how this problem can be transformed to the previous map fitting problem.

Lets denote the current vehicle position $\overrightarrow{t}(i)$ and orientation $\alpha(i)$ at time i in the global coordinate system GCS as $\overrightarrow{y}(i) = [\ \overrightarrow{t}^T(i) \quad \alpha(i)\ ]^T$. If only the wheel rotation measurements are used the current estimate $\widehat{\overrightarrow{y}}(I)$ at time I could have accumulated a large error:

$$\Delta\overrightarrow{y}(I) = \widehat{\overrightarrow{y}}(I) - \overrightarrow{y}(I) \tag{16}$$

We assume also that integrating the wheel rotation measurements gives us still a reasonable estimate of the vehicle pose within a certain small time window $T \ll I$:

$$\widehat{\overrightarrow{y}}(i) - \overrightarrow{y}(i) \approx \Delta\overrightarrow{y}(I) \text{ for } i = I - T, ..., I \tag{17}$$

First, we describe how a single global position update could be done:

1. Lets say that after some time $I - T$ we think that the accumulated error $\Delta\overrightarrow{y}$ is getting too big. At this point we can start detecting some distinctive points in the camera images that we expect to correspond mostly to some points from our known environment map $\mathcal{M} = \{\overrightarrow{x}_1, ..., \overrightarrow{x}_M\}$. We propose to use for example the corner-like points that, we hope, correspond mostly with the corners of the office environment and lie in the floor level (a simple way of extracting such points is described in the next section). A detected image point under the assumption that it lies in the known floor level (calibrated camera), can be transformed to its 2D floor position with respect to the vehicle. Using the current estimate $\widehat{\overrightarrow{y}}(i)$ we can transform the point to its global map position (in GCS). Performing this for all the detected points of interest for all the images during some short time interval $i = I - T, ..., I$ gives us a set of 2D points $\mathcal{Z} = \{\overrightarrow{z}_1, ..., \overrightarrow{z}_N\}$.

2. Because the accumulated error $\Delta\overrightarrow{y}$ is present in the $\widehat{\overrightarrow{y}}(i)$ during $i = I-T, ..., I$ (see the assumption (17)), the 2D points $\mathcal{Z}$ are actually the points in a translated and rotated coordinate system GCS'. We recognize here the previously analyzed map matching problem. We can use the previously proposed iterative procedure to fit the map $\mathcal{M}$ to the data $\mathcal{Z}$ and estimate $\Delta\overrightarrow{y}$. Note that at least two corner points measurements should form distinguishable clusters in order to be able to fit the map.

3. Update the current vehicle position estimate $\widehat{\overrightarrow{y}}(I)$.

In a practical implementation this global position update could be done constantly by adding the new measurements from the new images to $\mathcal{Z}$ and throwing away the measurements from $\mathcal{Z}$ that are older than T. The time interval T depends on how good the wheel rotation measurements are (the assumption (17) must be valid). It is also possible to use the older measurements but giving them some small weight factor to reduce their influence. There are various possibilities but that is beyond the scope of this paper.

The algorithm presents a simple way of combining the information from the camera, the information from the sensors measuring the wheels rotation and the knowledge contained in the known environment map. By looking at the equation (14) we see that if there are no points near some of the points from the environment map $\mathcal{M}$ this terms just disappear. So, if the environment map was huge and we observed only a few visible corners, this would have no influence for the map matching procedure. Therefore no complex visibility analysis is needed. For reducing computational time we could just throw away some points from the map that are too far from the current vehicle position. Note also that at some distance from the vehicle the transformation from image to the floor position is too sensitive to the image position errors and these points should be avoided anyway or their influence should be limited by a weight factor .

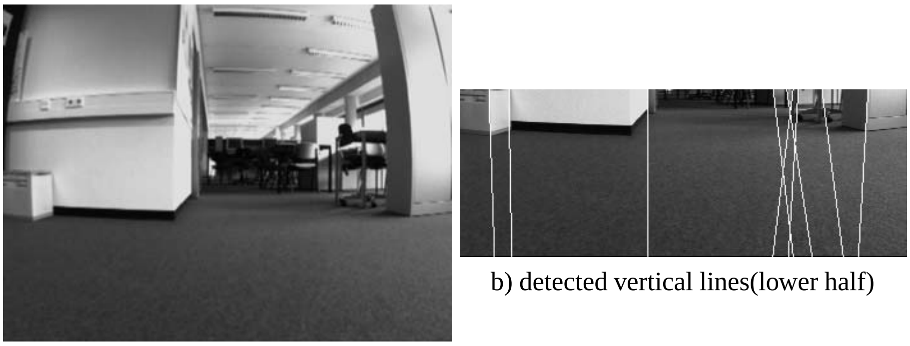
6 Experiments

We describe here some of our preliminary experiments. First we describe a simple way to detect corners that are likely to correspond to the environment map corners and lie in the floor level. Then, the results from one of the experiments are presented.

6.1 Point measurements

An office environment map consists of walls and doors. The corners and the doors present vertical structures in the images taken by the robot. These structures also end up on the floor level. We could try to detect the corners and the doors by detecting all almost vertical lines (edge detection + Hough transform) (see figure 4 b)). Finding the last long connected parts of the vertical lines (edges) we get the points that possibly lie in the floor level and hopefully correspond to the corners and doors from our global map. The camera is calibrated and using the well known *inverse perspective mapping* we can reconstruct 2D position of the points in the floor level from a single image (that is all under the assumption that the points are in the known floor level). It is obvious that this point detection method will lead to many false detections (outliers) and errors. However, the robust techniques we presented here should be able to resolve this problems.

Using these type of point measurements actually means that our map $\mathcal{M}$ consists of all the corners from our environment. Any other object with fixed placement that matches up the description (vertical edges that end up at the floor level) should also be included: cupboard edges, legs of the tables,...(see figure 5). The chairs and but also the tables that are often moved should be probably excluded.



b) detected vertical lines(lower half)

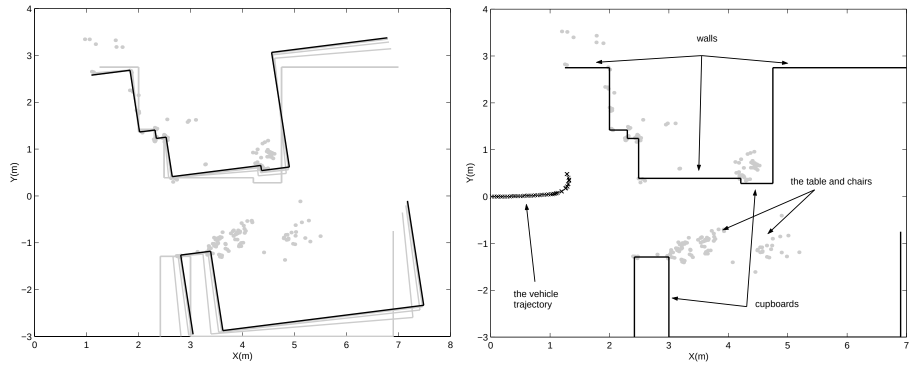
a) original scene

Figure 4: The observed scene

6.2 Global position update experiment

An example from our preliminary experiments is presented in figure 5. The vehicle was moving and the points of interest are detected as described in the previous section. This was done for 50 images. Time difference between two consecutive images was $200ms$ ($T = 50 * 0.2s = 10s$). All the detected points are transformed to the global map coordinate system according to the wheel measurement sensors (GCS'). The manually created map was than fitted to the points as presented in figure 5a. Using the estimated rotation and translation error $\Delta \overrightarrow{y}$ we present the updated vehicle trajectory $i = I - T, ..., I$ in the global map coordinate system (GCS) in figure 5b. We also present here all the detected points but transformed to the GCS.

The first image that was processed is presented in the figure 4. The vehicle was at the origin and looking in the x-axis direction. Note that the chair and the table in the right part of the image (behind the cupboard) were not included in the map. They produced many point measurements (see figure 5b) but that had no influence for the map fitting. We can also see that there are many errors for the far away points. It is also noticeable that often the lowest part of a vertical line is detected on some wrong higher level and the points are then projected as they were further than they actually are. However, the map fitting procedure is not very sensitive to this systematic errors.



a) matching to the real data (GCS') b) corrected (GCS)

Figure 5: Global vehicle position update

7 Conclusions

An iterative procedure is proposed to fit a map to a set of noisy measurements. We have shown how this could be used to solve the global autonomous vehicle position update problem. The procedure is very simple and robust. There are still many possibilities for improvement. The main point in the further research could be to test the procedure with other types of image measurements. In principle anything that could correspond to something in the map and can be detected, could be used. Still we need to be able to reconstruct the position of the detected object in 2D floor plane in order to use it directly in the current map fitting procedure. Relaxing this requirement to form a procedure that uses the detected objects image position only could be an interesting point to investigate. Further, the proposed map fitting procedure is a local search method. For practical use also some global search methods should be developed to be able to start without initialization and to recover from possible errors. Finally, the environment map was manually constructed. Developing techniques for automatic map generation is also important.

References

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis, *IEEE Trans. Pattern Analysis Machine Intelligence*.

Comaniciu, D., Ramesh, V. and Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift, *In Proc. IEEE Conf. on Comput. Vis. and Patt. Recog.* pp. 142–149.

Fukunaga, K. and Hostetler, L. (1975). The estimation of the gradient of a density function with applications in pattern recognition, *IEEE Trans. Information Theory* **21**: 32–40.

Hampel, F., Ronchetti, E., Rousseeuw, P. and Stahel, W. (1986). *Robust Statistics: The Approach Based on Influence Functions*, Wiley.

Huber, P. (1981). *Robust Statistics*, wiley.

Muller, J. (1997). Control architectures for autonomous and interacting agents: a survey, *Lecture Notes in Artificial Intelligence* **1209**: 79–98.

Parzen, E. (1962). On estimation of a probability density function and mode, *Annals of Mathematical Statistics* **33**(3): 1065–1076.

Schoute, A. (2001). Behaviour-based control framework for an autonomous mobile robot, *In Proceedings 3rd Workshop European Scientific and Industrial Collaboration, Enschede, Netherlands* pp. 237–246.

Tukey, J. (1976). Some recent developements in data analysis, *Presented at 150th Meeting of the Institute of Mathematical Statistics*.

Zhang, Z. (2000). A flexible new technique for camera calibration, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(11): 1330–1334.

Zivkovic, Z. (1998). Vision system for an autonomous vehicle, *IAESTE training technical report, University of Twente, 1998*.