Detection and Tracking Experiment Design in Various Environments

Levente Tamas, Gheorghe Lazea

Technical University of Cluj-Napoca, Robotics Research Group

Abstract: This paper presents the details of several mobile robot experiment designs including indoor, outdoor and urban variants. The aim of the paper is to give insights to setting up these tracking experiments covering both the software and hardware components as well as the application details. Beside these aspects the references to the perception, tracking and estimation parts are also pointed out for a wheel based mobile robots, bicycles and auto vehicles in various scenarios.

Keywords: Mobile robots, experiment design, obstacle detection, tracking.

1. INTRODUCTION

Although there are a great number of published papers about the mobile robot related measurements they often present only the results of the theoretical and practical investigations without putting too much emphasis on the experiment details. However a well designed experiment setup is essential in order to have a good dataset on which further data processing work can be carried out [Ch. Laughier, 2007].

Several research works present the building steps for mobile robots, focusing mainly on the hardware components [Kozlowski, 2009], [Cepon, 2010]. On the other hand there is a great variety of publications dealing with the results of the performed experiments. The target domain for the mobile robots experiments in this case is related the ones dealing with environment perception, detection and tracking [Arras and Mozos, 2009], [Bar-Shalom and Li, 2001].

The main aim of this paper is to detail those parts of a mobile robot experiment design which include the sensor integration, communication and data preprocessing and tracking algorithms. These modules vary according to the type of the performed measurements. In this paper there are presented a variety of designed experiments starting from indoor mobile robot experiments, covering outdoor and urban scenarios.

The first steps in this design process should be architectural definition of the experiment based on the available hardware and software components. According to this idea the first part of the paper covers the necessary hardware and software components for such an experiment.

The second part deals with the algorithms that were used for the target application part of the experiments including the perception, detection and tracking algorithms. Finally, there are presented the details regarding the experiment designs in various environments starting from indoor scenarios to urban and outdoor variants.

⋆ The authors are members of the Robotics Research Group from the Technical University of Cluj-Napoca, and they can be contacted: {levente.tamas, gheorghe.lazea}@aut.utcluj.ro

2. HARDWARE COMPONENTS

This section summarizes the hardware design principles based on our available resources. Later on this can be used as a starting point for choosing the software components for the experiment. The hardware components in the proposed experiments are mainly sensors reflecting the state of the environments and the own the state of the moving vehicle. According to this idea first those sensors are presented which give information about the surrounding environment. Further on some custom devices are introduced used in the different measurements.

2.1 Environment Perception

There are many possibilities to acquire information from the surrounding environment. The measurement methods can be divided into three major categories based on applied sensor and sensing technology: vision with one or more cameras [Droeschel et al., 2009], active triangulation [Perceptron, 2003] and time-of-flight (TOF) measurements. One of the most precise TOF measurement systems is based on laser scanners.

Planar Laser Range Finder The Sick LMS200 has a depth resolution of 2 [cm] and an angular resolution of 0.25°, 0.5°, or 1° depending on the configuration. The scanning cone of the device can be set either to 100° or 180° depending on the actual needs, while the maximum range of readings is up to 80 [m]. The scanning time is around 15 [ms] and additional time is required to send the data to the PC at 9600, 19200, 38400 or 500000 [kb/s].

Stereo Camera A low cost option for 3D information acquisition is the use of stereo or multiple cameras. The reconstruction of the third dimension from multiple images can be expressed in several ways [Demirdjian and Darrell, 2002].

By simple geometrical deduction the depth information Z can be obtained in the following ways:

$$Z = \frac{fB}{x_1 + x_2}1 \tag{1}$$

where the absolute difference $d = |x_1 - x_2|$ represents the disparity, f is the focal length, B is the baseline, and X_1, X_2 represent the image coordinates in the right and left camera. The disparity can be defined as the difference between the coordinates of the same feature in the left and right image.

High Resolution Camera In the proposed experiments a high resolution Canon EOS 350 camera was used in order to get high quality images for image processing. The communication with the laptop was done by using the PTP (Picture Transfer Protocol) implemented in the GPhoto GNU licensed package [GPhoto, 2010]. By using this package it was possible to remotely record high resolution (12 MB) images every second.

2.2 State Reflection Devices

This class of sensor gives information about the state of the mobile vehicle including its position, orientation or relative displacement [Borenstein et al., 1997]. From a wide variety of commercial sensors only a few modules were used the rest being designed&integrated on dedicated PCB-s.

GPS-DGPS The GPS was used as a reference signal in the experiments during the navigation. Although it is widely available, it cannot be used as a single source of information for the localization as long as signal loss may appear in urban environments or the precision of the localization it is not satisfactory.

The GPS module involved in the experiments was built using a LEA4-H precision core with a custom PCB. The precision of the receiver with a dedicated antenna was around 5-10 [m] in urban environment. In order to reduce this error the DGPS was chosen as a low cost alternative. By using the NTRIP protocol [NTRIP, 2010] it was possible to do both the real time and the off line processing of the correction data to the GPS signal. In this way the absolute error was reduced in the range of 1-3 [m] depending on the GPS signal quality.

Digital Compass In our project we used this sensor to estimate the heading of the autonomous vehicle. The TCM2 it is a sensor module which is replacing several sensors within a system. In addition to compass heading the TCM2 also supplies the pitch, roll, magnetic field data and temperature information. The accuracy for the heading information is 0.5° RMS for pitch and roll and 1° RMS for yaw. This sensor module was connected to the computing unit through the serial port.

Custom Odometer The need for a low cost custom odometer arose for an outdoor experiment performed on a bicycle. The specifications for this sensor were a response time lower than 20[ms] and an absolute accuracy of 1[m]. With these requirements the use of an optical encoder from a mechanical mouse, as it can be seen on the left hand side of the Figure 1. By passing an obstacle between the transmitter and receiver from this device, an impulse was generated on the serial port. Thus with this approach easily it can be received the odometer information from a wheel on a normal serial port. The limitation of this solution relies on the general interrupt periods in which the serial mouse port is queried which in most of the systems is 18[ms].

Fig. 1. The bike with custom odometer

2.3 Dedicated Actuators

This part describes two dedicated actuators for the one small mobile robot platform and for one for a real car. Both solutions are low cost options.

Wireless Command Tool For the urban experiment design a Tag4M [Folea and Ghercioiu, 2010] wireless device was used in order to send displacement commands to the mobile robot as it can be seen in Figure 2. This measurement tag is small $6x5[cm]$, low power consumption device able to communicate in a 802.11b/g network implementing the full TCP/IP stack. It can be configured to send measured analogue and digital data at a user defined rate to a server. As measured data the output of 3 axes analogue ADXL330 accelerometer was considered.

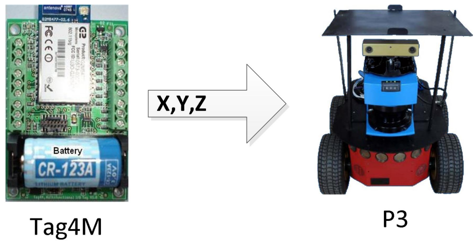

Fig. 2. Tag4M wireless tag control for the P3 mobile robot

In this way a wireless joystick like system was built which based on the inclination sent the command to the mobile robot. At the robot side the commands were received as UDP packets containing the raw values of the ADXL330 output every 250[ms].

Active Car Breaking System The active breaking system was designed according to the Figure 3. Its main purpose was to activate the breaking in the car in case that an obstacle was signalled by the main computer.

There are four main components in this design: the PC, the control unit, the actuator motor and the link to the breaking system. The detection of the obstacles by the PC and additional sensors will be presented in the next section. The control unit is based on an Atmeg128 microcontoller board and is communicating with the PC via the RS232 protocol. This control board commands the 12 [V] DC motor with a PWM signal, which activates the breaks. The link between the motor and the breaking system was constructed in such a way that the driver's break commands and the active breaking system act independently, thus enabling the both systems to work in the same time.

3. SOFTWARE COMPONENTS

Dealing with a wide variety of experiments it is important to have a well designed software architecture too. This can reduce substantially the development time by ensuring code reuse and robustness in the software modules. The first part presents the general concepts followed during the design phase. Further on, the involved components are highlighted and the details of the integration process.

3.1 Design Considerations

The main idea for design considerations was to obtain a balance between the maintainability of the packages and the flexibility for code reuse. The adapted solution in this case was the Domain Driven Design (DDD) which is well described in [Evans, 2003].

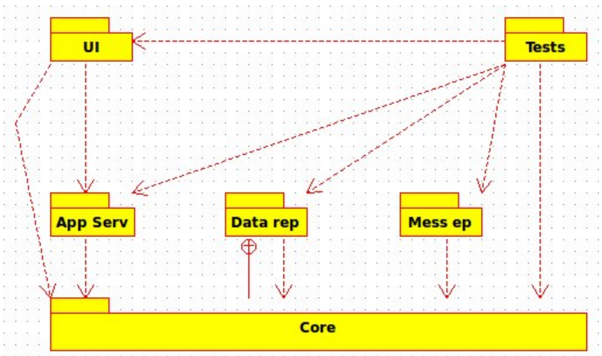

Fig. 3. Active breaking system overview

The domain layer contains the core of the package which for the detection&tracking applications might be the detection algorithm itself (e.g. Particle Filters). It implements the interfaces to the application layer, data repositories and message endpoints. The application service layers main role is the delegation and execution of the tasks. It is a middle layer between the core and the user interface without knowing about the data representation and the communication within the package. The communication with external data sources is done via the data repository layer, while the message endpoints specify the communication internally and externally to the package.

All layers have their test units defined. The adopted solution for unit tests is based on the test doubles, i.e. stub objects, which act as the other packages during the test. This may be useful for larger projects to separate the testing for layers [Meszaros, 2007].

3.2 Involved API Components

Several components were developed for the experiments on different platforms (Windows or Linux), thus it was important to have cross platform libraries which could be merged into a single application. Also different developing languages like C++, Python or Matlab M-code needed to be used in the same application. The target language was C++ on a Linux platform. The cross compilation of the M-code could be easily done by the Matlab Compiler both under Windows and Linux. The Robotics Operation System (ROS) [WillowGarage, 2010] seemed to be a good choice for integrating the different hardware and software modules.

4. LASER BASED DETECTION AND TRACKING

In this section the laser classifier is presented with the segmentation, feature extraction and classification components. Basically, the laser measures bearing-range information about the surrounding objects with a relative good accuracy (in the performed experiments $1cm$ accuracy at a $10m$ range).

4.1 Scan Segmentation

The scan segmentation belongs to the primary modules of the laser architecture among with the data acquisition and pre-filtering modules. The segmentation is the process of splitting a scan into several coherent clusters, i.e. point clouds. The choice of segmentation method is rather arbitrary and dependent on other design choices as the alignment and covariance estimation strategies [Borges and Aldon, 2004]. The current strategy is the one based on the simple assumption of Euclidean distances between segments adopted from [Mozos et al., 2007].

The laser range scan information is a set of beams of the form $Z = \{b_1, ..., b_L\}$. Each element b_j of this set is a pair of (θ_j, ρ_j), where θ_j is the angle of the laser beam relative to the robot and ρ_j is the distance from the reflecting surface.

4.2 Feature Extraction

This module extracts the relevant information from the segmented data and ensures robustness in the algorithm. The extracted information is used later on in the classifier module and can also be used for visualisation purposes too. The feature vector components may be chosen upon the required information [Mozos et al., 2007]. The basic set of feature which was used in the experiments contained the following e_1, e_2 and e_3 entries:

(1) e_1: object centroid;
(2) e_2: normalized Eucledian distances given by:

$$f2 = \sqrt{\Delta X^2 + \Delta Y^2} \quad (2)$$

(3) e_3: the standard deviation of the point from the r centroid computed for n points:

$$f3 = \sqrt{\frac{1}{n-1} \sum ||r_n - \bar{x}||} \quad (3)$$

These components are essential to the classifier.

4.3 GMM Object Description

A Gaussian mixture model (GMM) is a weighted combination of Gaussian probability density functions (pdf). These densities are used to capture the particularities of an object. In a GMM model the probability distribution of a x random variable is defined as a sum of M weighted Gaussian probability density functions:

$$p(x|\Theta) = \sum_{m=1}^{M} \alpha_m p(x|\theta_m) \quad (4)$$

where $\theta_1, ..., \theta_M$ are the parameter of the Gaussian distributions and $\alpha_1, ..., \alpha_M$ is a weighted vector such that

$\sum_{m=1}^{M} \alpha_m = 1$. A set of parameters for a mixture model is given by $\Theta = (\alpha; \theta_1, ..., \theta_M)$ where each parameter $\theta_m = (\mu_m, \Sigma_m)$ represents the mean and the covariance of the model with Gaussian pdf.

The Gaussian mixture parameters for each object of interest was determined using the expectation-maximization (EM) algorithm [McLachlan and Krishnan, 1997]. For each set of feature vectors ($\Omega^N = \Omega_1, ..., \Omega_N$) the EM algorithm computes M Gaussian parameter vectors that maximizes the joint likelihood of the Gaussian density [Paalanen et al., 2005] :

$$p(\Omega^N|q_i, \Theta^i) = \prod_{j=1}^{M} p(\Omega_j|q_i, \Theta^i_m) \tag{5}$$

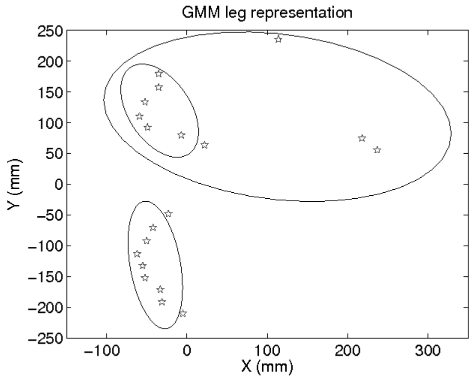

Fig. 4. Possible leg forms in the laser scan

A typical leg form described with GMM is presented on the Figure 4.

4.4 Bayesian Classifier

After a Gaussian mixture pdf for classified object is available a Ω_k feature-vector is considered in order to classify which category (q_i) fits the current observation. Based on a Bayesian decision framework the log-likelihood of the fitness is computed.

Computing the log-likelihood has the advantage of reduced computational effort by avoiding the computation of the exponential in the pdf and by turning the product (5) into sums. Furthermore, as the log-likelihood is a monotonically increasing function allows it can be used the former directly to classify the objects.

By considering the features equip-probable, the logarithm of the posterior probability $log(P(\Theta^i|\Omega))$ for all categories is proportional to the sum of the log-likelihood of the logarithm of the prior probability:

$$log(P(\Theta^i|\Omega)) \approx log(p(\Omega|\Theta^i)) + log(P(\Theta^i)) \tag{6}$$

It is more convenient to use Bayes' law to estimate the posterior probability as it uses only the likelihoods and the prior probability. The former pdf is computed at each scan, which will become in the next scan the last estimated posterior. Therefore the prior probability is updated dynamically as:

$$P(\Theta^i_k) = P(\Theta^i|\Omega_{k-1}) \tag{7}$$

4.5 Extended Kalman Filter for Tracking

A large number of mobile robots use position estimation based on the Kalman filters. Originally the theoretical backgrounds were formulated by Rudolf Kalman in 1960 and later on several extensions were developed [Borenstein et al., 1997]. The Kalman filter is an optimal recursive data processing algorithm for linear systems corrupted by noise.

The Extended Kalman filter (EKF) [Maybeck, 1979] uses a model to describe a discrete-time state transition. The filtering algorithm can be described in two steps: prediction and update. The *prediction step* is done at time instant $k-1$, before the information from the measurement is available and it is based on the previous state estimate $\mathbf{x}^+_{k-1}$. The *update step* is performed after the measurement from the time step k is available, and includes this information as a correction for the predicted state.

4.6 The Motion Models for Humans

Two motion models were adopted for people tracking. For both models the measured state variables were the positions in the Cartesian coordinates (x_k, y_k).

Position-velocity-heading (PVH) Model − used to estimate the human motion with constant velocity model. In our experiments this model was extended with the orientation ϕ_k and velocity v_k according to [Bellotto and Hu, 2005] as follows:

$$\begin{cases} x_k = x_{k-1} + \delta_k v_{k-1} cos\phi_{k-1} \\ y_k = y_{k-1} + \delta_k v_{k-1} sin\phi_{k-1} \\ \phi_k = \phi_{k-1} + n^{\phi}_{k-1} \\ v_k = v_{k-1} + n^{v}_{k-1} \end{cases} \tag{8}$$

with δ_k being the sampling time, n^{ϕ}_{k-1} and n^{v}_{k-1} the zero-mean Gaussian noises with $\sigma_\phi = \frac{\pi}{16}$ and $\sigma_v = 0.05ms^{-1}$.

Position-velocity-acceleration (PVA) Model − or referred as the $\alpha - \beta - \gamma$ filter [Bar-Shalom and Li, 1993] is the model of a Newtonian system represented in 2D coordinate system. Along a single axes the motion equations are given as follows:

$$x_k = \begin{bmatrix} 1 & \delta_k & \delta_k^2/2 \\ 0 & 1 & \delta_k \\ 0 & 0 & 1 \end{bmatrix} x_{k-1} + \begin{bmatrix} \delta_k^2/2 \\ \delta_k \\ 1 \end{bmatrix} n_{k-1} \tag{9}$$

The same equations are valid for the y_k coordinates. When using this model special care must be taken for computing the model noise, which is a function of the sampling rate δ_k. Additional information on filter tuning can be found in [Durrant-Whyte, 2006b].

In both cases the legs position are measured as bearing-range information with relative to the robot's position (x^R_k, y^R_k, ϕ^R_k) as follows:

$$\begin{cases} b_k = tan^{-1}\left(\dfrac{y_k - y_l}{x_k - x_l}\right) - \phi^R_k + n^b_k \\ r_k = \sqrt{(x_k - x_l)^2 + (y_k - y_l)^2} + n^r_k \end{cases} \tag{10}$$

where (x_l, y_l) are the offset of the laser device with respect to the robot. The noises n^b_k and n^r_k are device specific measurement Gaussian noises, considered for the experimental part $\sigma_b = \frac{\pi}{32}$ and $\sigma_r = 0.05m$.

5. TARGET EXPERIMENTS

5.1 Indoor Human Tracking with Laser

The aim of this experiment was to compare motion models for human position tracking in indoor environment. To compare the two models, the EKF was used to estimate the position of the detected person with respect to the robot position. The same dataset was tested checking the computational effort and the standard deviation of the innovation along the x and y axes during the experiments. The results are summarised in Table 1.

Table 1. Comparison of the PVH-PVA models

Criteria	PVH Model	PVA Model
Runtime (s)	6.2	7.9
$X_{Std}(cm)$	171	112
$Y_{Std}(cm)$	78	23

As it can be seen in Table 1, the PVH model runs faster, but it gives larger standard deviation along the axis compared to PVA. This should be expected as in the case of PVA there are 6 states compared to PVH with only 4 state variables.

5.2 Bycicle Experiment

The aim of the two sensors fusion was to obtain a continuous, bounded-error position estimation for the vehicle. While in the case of the GPS the error usually is not drifted in time - thus is reliable on longer distances - on short distances (less than 3 [m]) it cannot be used for position estimation due to the limited accuracy of the sensor. It can also be the case that the data from this type of sensor is not available (e.g.: in tunnels, or within high buildings).

On the other hand, the dead reckoning (DR) sensors produce unbounded errors in the position estimation, unless they are regularly reset from another type of sensor reading. Thus the external update form the GPS would increase the position estimation performance both on short and long distances.

The Process Model For Estimation The model used to describe the bicycle motion under the constant velocity assumption in our experiment was extended beside the orientation ϕ_k and velocity v_k with the radius of the wheel R according to [Durrant-Whyte, 2006a] as follows:

$$\begin{cases} x_k = x_{k-1} + \delta_k v_{k-1} R cos\phi_{k-1} \\ y_k = y_{k-1} + \delta_k v_{k-1} R sin\phi_{k-1} \\ \phi_k = \phi_{k-1} + n^{\phi}_{k-1} \\ v_k = v_{k-1} + n^{v}_{k-1} \\ R_k = R_{k-1} + n^{R}_{k-1} \end{cases} \tag{11}$$

with δ_k being the sampling time, n^{ϕ}_{k-1} and n^{v}_{k-1} the zero-mean Gaussian noises with $\sigma_\phi = \frac{\pi}{64}$, $\sigma_v = 0.15ms^{-1}$ and $\sigma_R = 0.04m$. The wheel information is considered to be constant, but with small perturbations during the time (e.g. changing the load on the bicycle). Further information regarding the modelling of the error for such a case can be found in [Durrant-Whyte, 2006a]. The odometer information is given as a function of the wheel radius. As this is not directly measurable, it can only be estimated using an external sensor reading.

The challenge in the observation model was the combination of the asynchronous readings. As the rate of the DR readings were with an order of magnitude higher than the ones from the GPS, a sequential [Bellotto and Hu, 2005] approach was adopted.

According to the sequential update, each time a GPS data is available, a position estimate update is also performed. Between two GPS measurements, the position estimation based on the DR data is done.

Beside the sequential update of the estimate from the GPS data, the covariance of the estimate is reduced to the values of the GPS measurement covariance matrix. A similar approach is presented in the case of systems based on the landmark observations. The covariance reduction in this case is rather straightforward: the uncertainty in the position after the GPS data is fed into the filter is given by covariance of the GPS data.

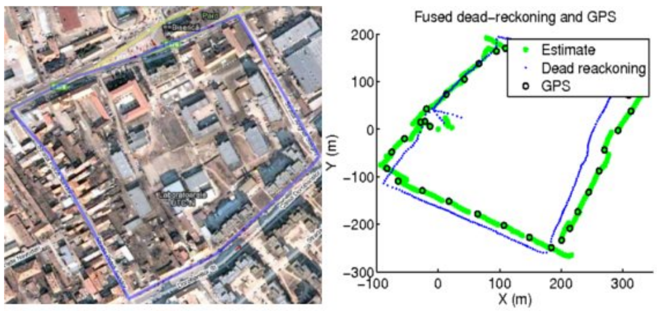

Fig. 5. The original map (a); fused odometer and gps data (b)

The fused DR and GPS data is shown in Figure 5. The GPS data and the DR readings were fused with the Extended Kalman Filter [Bar-Shalom and Li, 2001]. As it can be seen, the DR data after a certain point has heading errors. These can easily occur in case that a large electromagnetic field is encountered in the near proximity of the compass. This was also in the case of the current experiment, when a trolley passed by the compass.

Between two consecutive GPS readings, the DR can perform acceptable. Even if the data would be missing for a short period (e.g. tunnel without GPS availability), the DR still could give a reasonable position estimation.

5.3 Active Car Breaking Systems

The car with the sensors and the active breaking system is shown on the Figure 6. On the left hand side the Bumblebee2 stereo camera and the LMS200 laser range finder (LRF) is shown. These sensors were used for the pedestrian recognition and they were mounted on a metal support [Tamas et al., 2009]. On the right hand side of the figure the breaking system is presented. This contains the control unit (CU), the actuating motor and the break cable connected in parallel to the main breaking system.

The data acquisition and processing of the information from the stereo camera and the LRF was done within ROS with detection libraries developed in Matlab and

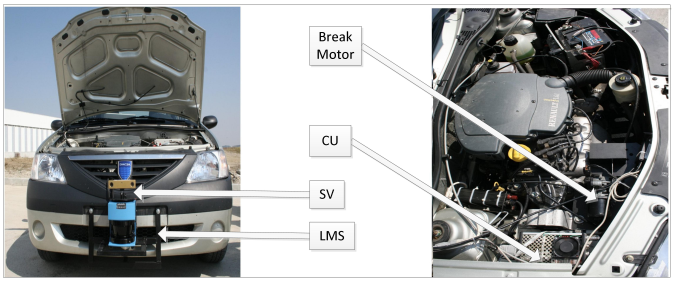

Fig. 6. The car with the active breaking system

cross compiled to the Linux platform. The detection with the LRF was based on the Gaussian Mixture Model representation of the predefined shapes [Premebida and Nunes, 2006]. The active breaking system was able to slow down the car in safe conditions in case of the pedestrian crossing the front of the vehicle. The test scenario was performed on normal road surface with a cruising speed of 40 $[km/h]$ and with the pedestrian crossing in front of the car at a 5 [m] distance from the car. These specifications can be further more extended with faster communication cards between the sensors and the main computer as well as by using a breaking system with lower response time.

6. CONCLUSION

This paper covered a series of mobile vehicle experiments starting from indoor 4 wheeled mobile robot, outdoor bicycle or car navigation related experiments. The setups for these applications were presented separately, highlighting the important setup&configuration aspects for each of them.

For the future it is intended to be used advanced detection&tracking algorithms in the traffic implemented on high speed FPGA boards in order to be suitable both for aerial and urban scenes. We are also intending to implement an algorithm which combines 3D with image data for improved segmentation.

ACKNOWLEDGEMENTS

This work was done within the PRODOC project at the Technical University of Cluj-Napoca, supported by European Social Fund, POSDRU/6/1.5/S/5 as well as 4D-POSTDOC postdoctoral project POSDRU/89/1.5/S/52603.

REFERENCES

O. Arras and . Mozos. People detection and tracking workshop. IEEE International Conference on Robotics and Automation, 2009.

Y. Bar-Shalom and X. Li. *Estimation and Tracking-Principles, Techniques and Software.* Artech House, Norwood, MA, 1993.

Y. Bar-Shalom and X. Li. *Estimation with Applications to Tracking and Navigation.* Artech House, Norwood, MA, 2001.

Nicola Bellotto and Huosheng Hu. Multisensor integration for human-robot interaction. *IEEE Journal of Intelligent Cybernetic Systems*, 1, 2005.

J. Borenstein, H. R. Everett, L. Feng, and D. Wehe. Mobile robot positioning sensors and techniques. *J. Robot. Syst*, 14:231–249, November 1997.

G. A Borges and M. J Aldon. Line extraction in 2d range images for mobile robotics. *Journal of Intelligent & Robotic Systems*, 40:267297, 2004.

P. Cepon. Experimental mobile robotic platform. *International Journal of Mechanics and Control*, 11:59–65, 2010.

R. Chatila Ch. Laughier. *Autonomous Navigation in Dynamic Environments.* Springer, 2007.

D. Demirdjian and T. Darrell. Multiple-hypothesis disparity maps and image velocity for 3-d motion estimation. *International Journal of Computer Vision*, pages 219–228, 2002.

F. Droeschel, D. Wisspeintner, T. May, S. Surmann, and H. Maurelli. A 3d laser scanner system for autonomous vehicle navigation. Muchen,, 2009. ICAR.

H. Durrant-Whyte. *Introduction to Estimation and the Kalman Filter.* ACFR, 2006a.

H. Durrant-Whyte. *Multi Sensor Data Fusion.* Australian Center for Field Robotics, 2006b.

E. Evans. *Domain-Driven Design: Tackling Complexity in the Heart of Software.* Addison-Wesley Professional, 2003.

S. Folea and M. Ghercioiu. Tag4m, a wi-fi rfid active tag optimized for sensor measurements. 2010.

GPhoto, 2010. URL `http://www.gphoto.org/`.

S. Kozlowski. Control system for designed mobile robot - design, implementation and test. *Robot Motion and Control*, pages 401–410, 2009.

P.S. Maybeck. *Stochastic Models, Estimaton and Control*, volume 1. Academic Press, 1979.

G McLachlan and T Krishnan. *The EM Algorithm and Extensions.* John Wiley and Sons, New York,, 1997.

G. Meszaros. *xUnit Test Patterns: Refactoring Test Code.* Addison-Wesley, 2007.

O Mozos, K Arras, and W Burgard. Using boosted features for detection of people in 2d range scans. In *In Proc. of the IEEE Intl. Conf. on Robotics and Automation*, 2007.

NTRIP, 2010. URL `http://www.ntrip.org/`.

P Paalanen, J. K Kamarainen, J Ilonen, and H Klviinen. Feature representation and discrimination based on gaussian mixture model probability densities - practices and algorithms. Technical report, 2005. URL `www.it.lut.fi/project/gmmbayes`.

Perceptron. Scanworks 3d product brochure, 2003. URL `www.perceptron.com`.

C. Premebida and U. Nunes. A multi-target tracking and gmm-classifier for intelligent vehicles. Toronto,, 2006. 9th International IEEE Conference on Intelligent Transportation Systems.

L. Tamas, Gh. Lazea, I. Szoke, M. Popa, and A. Majdik. Laser based localization techniques for indoor mobile robots. In *IEEE International Conference on Learning and Adaptive Behavior in Robotic Systems*, 2009.

WillowGarage, 2010. URL `http://www.ros.org/wiki/`.