Metrol. Meas. Syst. Vol. XVI (2009), No 3, pp. 359-369

PAN POLSKA AKADEMIA NAUK

METROLOGY AND MEASUREMENT SYSTEMS

Index 330930, ISSN 0860-8229

www.metrology.pg.gda.pl

M&MS

NON-LINEAR MULTIMODAL OBJECT TRACKING BASED ON 2D LIDAR DATA

Michael Thuy, Fernando Puente León

Institut für Industrielle Informationstechnik, Karlsruhe Institute of Technology, Hertzstrasse 16, D-76187 Karlsruhe, Germany
(✉ thuy@kit.de, +49 721 608 4517*, puente@kit.de)*

Abstract

The contribution introduces a novel approach for tracking objects based on two-dimensional lidar data. As a central tracking engine, we employ a particle-filter-based solution which is capable of modelling non-linear dynamic processes as well as non-Gaussian noise distributions for the underlying process and sensor as well. In contrast to other lidar-based tracking approaches, no newly detected objects have to be associated to already known objects in an explicit manner. Since our weighting function is multi-modal, the association is done by the filter itself.

Keywords: fusion, lidar, data association, tracking.

1. Introduction

Object detection is one of the key abilities of advanced driver assistance systems. A vast literature on this subject exists, and different sensors (radar, lidar, monocular and stereo cameras, *etc*.) have already been employed to approach a robust object detection. Generating object hypotheses with a monocular camera was shown in [1]. Facing the same problem with a stereo camera and an appropriate calibration algorithm was *e.g*. presented in [2].

Since acquiring pictures and all attached image processing often suffers from alternating ambient light conditions, the use of different sensor technologies can yield a drastical improvement of system robustness. Taking active sensors like radar or lidar (light detection and ranging) scanners breaks the limitation of the surrounding light condition due to their active sensing principle [3, 4, 5]. The sensor itself emits a narrowband laser pulse at a specified angle and measures the time the light takes to come back to the sensor. Besides the very low dependence of the environmental light condition, the sensor only delivers a distance measurement if the light hits an object. Consequently, every measured point represents a potential part of an object.

Our contribution presents an approach to object detection based on two 2-D lidar scanners. It does not only encompass an object perception, but additionally the objects are tracked over time as well. This proceeding requires an adequate dynamic model for the objects. Since the amount of data recorded by a 2-D lidar scanner is much less than the data from an image sensor, the signal processing can take place faster, which automatically reduces the reaction time of a technical system.

Common lidar-based tracking approaches utilize a linear or an extended Kalman filter to predict the new object position [6]. In the case of the linear filter, the underlying dynamic model allows an object to move forward independently of the movement to the left or right. Considering the physical constraints of an automobile, this model is not appropriate. In order to avoid these physical contradictions, one can utilize a non-linear process model like the bicycle model. Still, this model simplifies the real physical behaviour, but it prevents an

Article history: received on Jul. 27, 2009; accepted on Sep. 11, 2009; available online on Oct. 2, 2009.

unrealistic movement of the modelled automobiles. But since the extended Kalman filter tries to linearize the process dynamics in the current operating point, errors will occur when using this approach.

To minimize the errors due to the non-linear physical characteristics of the real object, we employ a particle filter based on Monte Carlo simulations for object tracking [7]. Thereby, it is possible to model non-linear process dynamics in a very easy and efficient way. The additional expense is longer computation time, which is associated with the propagation of a high amount of particles through the non-linear process model. However, the following sections show that it is possible to perform a real-time tracking with a sufficient number of particles.

Many approaches use a separate filter for each tracked object. In this case, the computational load becomes non-deterministic. Since a scene can contain an arbitrary number of objects, the tracking engine has to instantiate the corresponding amount of filters-provided the objects are correctly detected by the sensor. To constrain the computational load, one could restrict the number of tracked objects. An alternative would be to use a single filter that can handle several objects. This can be achieved by multi-modal filters, in which the probability density function describing the state variables objects estimated shows more than one maximum.

Based on vision and lidar data, [8, 9, 10] already showed a solution capable of tracking several objects with only a particle filter. Our contribution enhances this solution by building a new weight function. This weight function incorporates all the extracted data coming from the sensor. As a consequence, the clustering process concerning the assignment between particle and measurement is done by the filter itself. By adjusting the coefficients within the weight function, the sensor-specific characteristics can be modelled as well.

At the end of the paper, results are provided that demonstrate the achieved tracking outcomes. Having a scanning cycle of 13.3 ms, the processing can still be performed in real-time.

The paper is organized as follows: Section 2 illustrates the basic sensor setup and the data path to the detection stage. Section 3 introduces the statistical fundamentals for our contribution. The next section describes the chosen process model as well as the new weight function. Thereafter, Section 5 characterizes the feature extraction stage. After having explained the preprocessing, Section 6 describes the filter. Section 7 presents experimental tracking results. The last section closes the paper with a short conclusion.

2. Fusion of Raw Lidar Data and Inertial Measurement Unit

2.1. Sensor setup and data processing

In our object tracking approach, we use two separate one-layer laser scanners (2-D lidars). The two sensors are synchronized by hardware and sample object distances with an angular resolution of a half degree and an overall angular range of 180 degrees. In order to get clear shapes of the surrounding objects, the first scanner is placed at the front bumper of the car. The second scanner is fixed at the rear bumper. For an ego-movement without a pitch angle, the scanning plane is adjusted parallel to the street. The maximum scanning distance is claimed to be 80 m, but it is in fact mainly restricted through the actual object size.

Additionally, in our contribution we use the data coming from an inertial measurement unit (IMU) combined with a satellite-corrected differential global positioning system (DGPS). Consequently, the system delivers the exact ego-position and, additionally, the current ego-heading. By calibrating the two lidar sensors with respect to the IMU's coordinate system, a

first calculation of the ego-position in a fixed world coordinate system as well as of the absolute position of the detected object is enabled.

As noticed in [11] and [12], the lidar scanners are connected to the central computer and served by a corresponding hardware driver. The data is then read out of the real-time database by the lidar tracking, which is responsible for the tasks of object detection, association, and tracking.

2.2. Spatial and temporal registration of 3-D laser points

As already mentioned, the two lidar scanners cooperatively sense their environment, *i.e.* they deliver information which is – related to their coverage area – independent and can thus be fused complementarily to get evidence about the surrounding. In the present case, combining the data of both sensors requires to register them spatially into a common coordinate system.

The spatial registration involves the process of combining the hypotheses derived from both the front and rear sensor. Since they are calibrated one with respect to the other, the data measured by one of the sensors has to be shifted and rotated according to the calibration parameters. Then, the scanning points are described by their position within a coordinate system in which the car's reference point represents the origin. But still, the description of the environment has a relative character regarded from the fixed world's point of view.

2.3. Fixed world observation model due to coordinate transformation

Transforming the above-mentioned relative environment description $(x_{\text{rel}}, y_{\text{rel}})$ into an absolute one requires to incorporate the absolute ego-position $(x_{\text{ego}}, y_{\text{ego}})$ and the current heading α with respect to the fixed coordinate system, which is possible thanks to the data available from the IMU. To obtain a continuous Cartesian coordinate system, we transform the ego-position from its native description through longitude and latitude into UTM coordinates. Starting from that ego-position we can, by shifting and rotating, register the scan points within a global system.

Assuming the rotation matrix **R** based on the ego-heading α, one can now calculate the absolute positions of the relative coordinates $(x_{\text{rel}}, y_{\text{rel}})$:

$$\begin{pmatrix} x_{\text{fix}} \\ y_{\text{fix}} \end{pmatrix} = \begin{pmatrix} x_{\text{ego}} \\ y_{\text{ego}} \end{pmatrix} + \mathbf{R}(\alpha) \cdot \begin{pmatrix} x_{\text{rel}} \\ y_{\text{rel}} \end{pmatrix}. \tag{1}$$

For the sake of clearness, in the following we will denote the coordinates $(x_{\text{fix}}, y_{\text{fix}})$ without an index. If coordinates refer to relative coordinates, they will be marked with the index “rel”.

The great benefit of describing the environment within global coordinates is obvious: all the tracked object movements are independent of the own ego-movement. This issue is emphasized in the filter section. Another item to be considered in connection with the IMU is its update rate. Since the latter is one order of magnitude above the scanning rate, no position interpolation is necessary. Otherwise, one would have to interpolate the ego-movement, for instance, with a separate filter.

3. Tracking Theory of Multimodal Object Tracking

3.1. General formulation of the posterior density distribution

The Monte Carlo simulation can be seen as a particular solution to the general prediction and tracking problem of the current state vector $\mathbf{x}_t$ based on all available observations $\mathbf{Z}_t$. Therefore, one has to determine the posterior probability density function, which estimates the distribution of the current state vector based on all measurements:

$$p(\mathbf{X}_t|\mathbf{Z}_t), \tag{2}$$

where $\mathbf{X}_t$ represents the state variable vectors $\mathbf{x}_t = \left[x_1^t,\ x_2^t,\ x_3^t, \mathrm{K}\ , x_n^t\right]^{\mathrm{T}}$ for all points in time until the current time *t*:

$$\mathbf{X}_t = \{\mathbf{x}_0, \mathbf{x}_1, \mathrm{K}\ , \mathbf{x}_t\}. \tag{3}$$

Analogously, $\mathbf{Z}_t$ denotes the observation vector $\mathbf{z}_t = \left[z_1^t,\ z_2^t,\ z_3^t, \mathrm{K}\ , z_n^t\right]^{\mathrm{T}}$ until the point in time *t*:

$$\mathbf{Z}_t = \{\mathbf{z}_1, \mathbf{z}_2, \mathrm{K}\ , \mathbf{z}_t\}. \tag{4}$$

In the following, the likelihood function is described by:

$$p(\mathbf{z}_t|\mathbf{x}_t). \tag{5}$$

Assuming an underlying Markov process and the likelihood distribution as $p(\mathbf{z}_t|\mathbf{x}_t)$, we can now apply Bayes' rule to obtain the posterior density:

$$p(\mathbf{x}_t|\mathbf{Z}_t) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)\cdot p(\mathbf{x}_t|\mathbf{Z}_{t-1})}{p(\mathbf{z}_t|\mathbf{Z}_{t-1})}\ . \tag{6}$$

For the updating phase – *i.e.* the estimation of the current state based on all past observations – one can determine the prediction density distribution as follows:

$$p(\mathbf{x}_t|\mathbf{Z}_{t-1}) = \int_{-\infty}^{t} p(\mathbf{x}_t|\mathbf{x}_{t-1})\cdot p(\mathbf{x}_{t-1}|\mathbf{Z}_{t-1})\mathrm{d}\mathbf{x}_{t-1}. \tag{7}$$

From a theoretical point of view, the problem is solved. However, a look at Eq. (7) reveals a complex multidimensional integral which incorporates an indefinite number of observations $\mathbf{Z}_{t-1}$ until the time $t-1$. Consequently, one has to seek for a feasible numerical approach to calculate the posterior density distribution $p(\mathbf{x}_t|\mathbf{Z}_{t-1})$.

3.2. The Monte Carlo approach

The basic idea about the sequential Monte Carlo simulation is to represent the posterior distribution through a limited amount of particles. Each particle can be described through a corresponding state vector with its appropriate weight. Thus, the posterior distribution is sampled by the particles.

Using *N* particles with their state vectors $\{\mathbf{X}_t^i, i = 0, \mathrm{K}\ , N\}$ one can infer the corresponding weights as follows:

$$w_t^i \propto \frac{p(\mathbf{X}_t^i|\mathbf{Z}_t)}{q(\mathbf{X}_t^i|\mathbf{Z}_t)}, \tag{8}$$

where $q(.)$ represents the importance density. But still we have to incorporate all the observations $\mathbf{Z}_t$ till time t.

Further proceeding finally leads to an iterative description of the weights:

$$w_t^i \propto w_{t-1}^i \cdot \frac{p\left(\mathbf{z}_t \middle| \mathbf{x}_t^i\right) p\left(\mathbf{x}_t^i \middle| \mathbf{x}_{t-1}^i\right)}{q\left(\mathbf{x}_t^i \middle| \mathbf{x}_{t-1,}^i \mathbf{z}_t\right)}. \tag{9}$$

Now the choice of the importance density becomes the most crucial task. Assuming that the importance density function to represent the process model as well as the change of the state vector does not depend on the observations:

$$q\left(\mathbf{x}_t \middle| \mathbf{x}_{t-1}, \mathbf{z}_t\right) = q\left(\mathbf{x}_t \middle| \mathbf{x}_{t-1}\right), \tag{10}$$

one can easily determine the formula for the weights w_t^i of the i-th particle:

$$w_t^i \propto p\left(\mathbf{z}_t \middle| \mathbf{x}_t^i\right). \tag{11}$$

Eq. (11) means that the corresponding particle weight w_t^i can be calculated with the help of the likelihood function $p\left(\mathbf{z}_t \middle| \mathbf{x}_t^i\right)$. As all the calculations can now be done sequentially, the filter is named Sequential Importance Sampling filter.

After all, an initial state distribution – represented by a certain amount of particles with their corresponding weights – can now be propagated in time for one time step t utilizing the process model resp. the process function. The observation then leads to the corresponding particle weight w_t^i. But what now can happen is a depletion of the particles, such that many particles feature weights that tend to zero. This leads to an unbalanced sampled posterior distribution. A resampling strategy prevents this development. Therefore, after the measurement update, the particles with a low weight are omitted, whereas particles with high weights are duplicated.

3.3. Final tracking filter

After the theoretical introduction in the last paragraph, all the ingredients for the filter design are discussed. Based on the knowledge of the underlying process model $q\left(\mathbf{x}_t \middle| \mathbf{x}_{t-1}\right)$ and the characteristic likelihood function $p\left(\mathbf{z}_t \middle| \mathbf{x}_t\right)$ for the sensor, one can compose the particle filter.

In contrary to a Kalman or extended Kalman filter, it is now possible to model all noise sources as well as non-Gaussian. The multimodal character of the particle filter occurs due to the non-linear process model as well as a possible multimodal likelihood function.

4. Non-Linear Observation and Process Model

4.1. The bicycle process model

After the fundamentals have been introduced, we can now present the underlying process function. First, we have to define our state vector **x**:

$$\mathbf{x}[t] = (x[t],\ y[t],\ v[t],\ a[t],\ \beta[t])^{\mathrm{T}}, \tag{12}$$

where $(x,y)^{\mathrm{T}}$ represents the object reference point, v the object velocity, a the object acceleration, and β the yaw angle. The values are illustrated in Fig. 1.

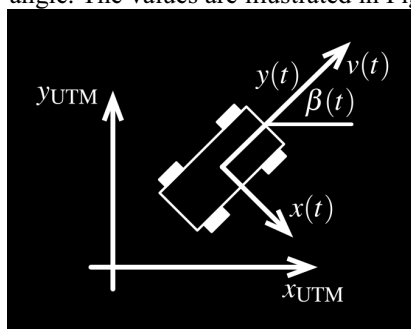

Fig. 1. State values representing the process model.

With the definition of these state variables, we can now formulate our non-linear, time-discrete process model:

$$\mathbf{x}[t+1]=\mathbf{x}[t]+\begin{pmatrix} v[t]\cdot\cos(\beta[t]) \\ v[t]\cdot\sin(\beta[t]) \\ a[t] \\ 0 \\ 0 \end{pmatrix}\Delta T+\begin{pmatrix} 0.5\cdot v[t]\cdot\cos(\beta[t]) \\ 0.5\cdot v[t]\cdot\sin(\beta[t]) \\ a[t] \\ 0 \\ 0 \end{pmatrix}\Delta T^2=\mathbf{x}[t]+\mathbf{F}(\mathbf{x}[t]), \tag{13}$$

which characterizes the change of the state variables within one time step ΔT.

Assuming this process model as the actual object movement characteristics, no object can move to the left or to the right without going forward as well. This real physical behaviour prevents a side-slipping movement of tracked objects. This often occurs with linear object models and non-accurate object detection, when one tracked object is split into two objects.

In order to handle process model errors correctly, we have to expand the process model with the help of noise sources. Consequently, equation (13) becomes:

$$\mathbf{x}[t+1]=\mathbf{x}[t]+\mathbf{F}(\mathbf{x}[t])+\mathbf{n}[t], \tag{14}$$

where **n** incorporates the specific noise source. As we model the noise terms to be independent unbiased Gaussians, their characteristics are described by their variance vector **r**.

4.2. The observation model

To complete all ingredients for the filter, we now have to face the observation model. This means – analogously to the definition of state variables – the determining of the values one can extract form the measurement. For the sake of robustness, we only analyze a reference point $(x_{\mathrm{obj}},y_{\mathrm{obj}})$ for each detected object. Analyzing the yaw angle leads to big errors for long distances.

As already mentioned in equation (11), one has to find the characteristic likelihood function for the defined observation values. For lidar sensors, often Gaussian noise models are used, which leads to the desired two-dimensional likelihood function that only depends on the extracted coordinates x_{obj} and y_{obj} as observation variables:

$$w_t^i \propto p\left(\mathbf{z}_t \middle| \mathbf{x}_t^i\right) = A \cdot \exp\left(-\frac{\left(x_{\text{obj}} - x^i\right)^2}{2\sigma_x^2} - \frac{\left(y_{\text{obj}} - y^i\right)^2}{2\sigma_y^2} \right). \tag{15}$$

One has to emphasize once again that the only data needed by the filter is the absolute object position. All the other state variables are inferred from the process model. Thus, only x_{obj} and y_{obj} are needed by the likelihood function. Still, the observation model does not handle more than one object.

Since the feature extraction stage yields the measured object positions $\left(x_{\text{obj,k}}, y_{\text{obj,k}}\right)$, one can now expand the likelihood function to a multimodal behaviour. To this end, we regard each detected object position as a hypothesis that represents a potential object in space. Consequently, one can formulate the resulting weight function as follows:

$$w_t^i \propto p\left(\mathbf{z}_t \middle| \mathbf{x}_t^i\right) = \sum_k \left(A_k \cdot \exp\left(-\frac{\left(x_{\text{obj,k}} - x^i\right)^2}{2\sigma_{x,k}^2} - \frac{\left(y_{\text{obj,k}} - y^i\right)^2}{2\sigma_{y,k}^2} \right) \right), \tag{16}$$

where A_k denotes the measurement specific constant which adjusts the influence of the corresponding object within the whole sum.

In our contribution, we did not model the sensor variance as a value depending on the measurand, since the influence is negligible. The variance itself is determined by experiment.

5. Feature Extraction Stage

The main task of the feature extraction stage is the determination of the measurement values needed by the filter, *i.e.* the generation of an object list containing the object coordinates. Since all the tracking is done in a fixed-world coordinate system, the object position must be converted to an UTM description.

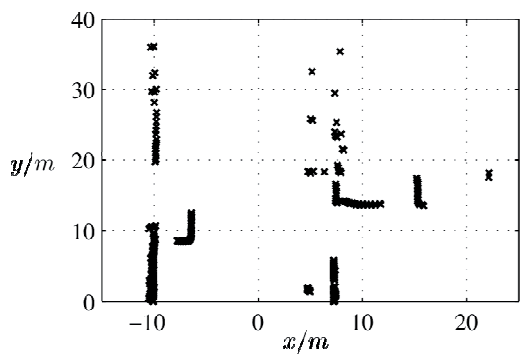

Fig. 2. Raw data of a lidar scan.

A lidar scan from the front lidar can look as shown in Fig. 2. As denoted in this figure, detecting objects like cars out of the lidar raw points can be solved by analysing the distance of consecutive laser points. Since the laser rays are – even for long distances up to 80 m – sufficiently close each to another, an object silhouette yields to lidar points which are very close to each other as well. If other objects appear, it is very likely that the distance between two laser hits shows a distance leap.

Considering the distance between two laser points belonging to the same object, one can easily imagine that the distance grows with increasing measurement distance. This is due to the equally-spaced angular resolution of the scanner.

Calculating the distance-dependent spacing ΔD between two consecutive and independent lidar points r_i and r_{i+1} leads to the following formula:

$$\Delta D(r_{i+1}, r_i) = \sqrt{r_i^2 + r_{i+1}^2 - 2\, r_i\, r_{i+1} \cos(\Delta\alpha)}\,, \tag{17}$$

where the angle $\Delta\alpha$ denotes the constant scanning angle of the lidar.

The threshold which should be exceeded by two subsequent lidar points in order to separate them into two objects due to the distance leap can be described by:

$$d_{\text{th}}(r_{i+1}, r_i) = d_{\text{th},0} + d_{\text{th},1}(\Delta\alpha, r_{i+1}, r_i)\,. \tag{18}$$

This formula can be separated into two additive terms. The first term $d_{\text{th},0}$ represents a constant bias. The second summand is responsible for the distance dependence and can be calculated according to the next equation:

$$d_{\text{th},1}(\Delta\alpha, r_{i+1}, r_i) = \tan(\Delta\alpha) \cdot \min\{r_{i+1}, r_i\}\,. \tag{19}$$

If the distance between two consecutive laser points is less than d_{th}, the points are associated with the same object. Otherwise, a point of a new object has been detected. Consequently, the segmentation result is an obstacle list, containing all detected obstacles including their corresponding scanning points. At this point, no lidar point has been omitted, and no a priori shape information has been incorporated.

In order to generate for each object a reference point, we calculate its center of gravity, which results in the object position $\left(x_{\text{obj}}, y_{\text{obj}}\right)$.

6. Filter Composition

The following paragraph describes the resulting filter, after all its parts have been introduced. The flow for tracking objects is formulated sequentially:

1. All the particles from time step $t - 1$ are propagated through the process model.
2. The data coming from the two lidars at time step *t* are registered in the common fixed world coordinate system.
3. The data segmentation stage segments the raw lidar data of the current frame. The result is a list of objects characterized through their absolute coordinates $\left(x_{\text{obj}}, y_{\text{obj}}\right)$.
4. Based on the object features, the multi-modal weight function according to equation (16) is constructed.
5. Recalculation of the weights of each particle is perfomed.
6. Finally, the estimated object positions – represented by clusters of particles – are extracted out of the state space. This can be performed by dynamic k-means clustering, which permits an arbitrary number of clusters. The clustering mechanism automatically increases the number of clusters in the case of high discrepancy of particles from their corresponding center of gravity. However, as we are tracking automobiles, the clusters basically do not interfere, and non-representative points can be omitted to achieve a higher efficiency.

7. Results

This section presents the results. The corresponding processing is performed in real time. This means, the presented calculation can be completed within the sensor's sample time. Therefore, the code has been optimized for parallel computing on a multicore processor architecture.

Fig. 3 shows the chosen initial spatial state distribution of the particle filter. To this end, 1000 particles were spread uniformly around the ego-position.

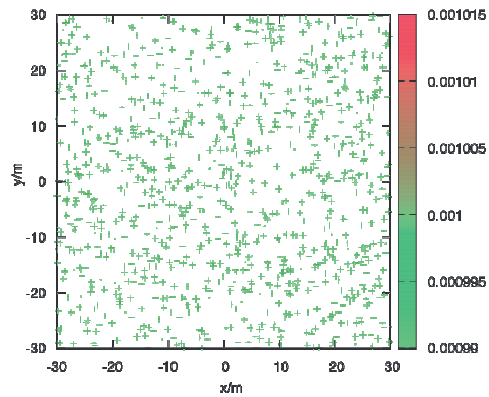

Fig. 3. Initial spatial state distribution.

Fig. 4 shows the state distribution after object tracking over 140 time-steps. One can clearly identify the resulting clusters, which represent different objects. The sidebar on the right side describes the specific particle weight. The clustering process successively separates the three clusters.

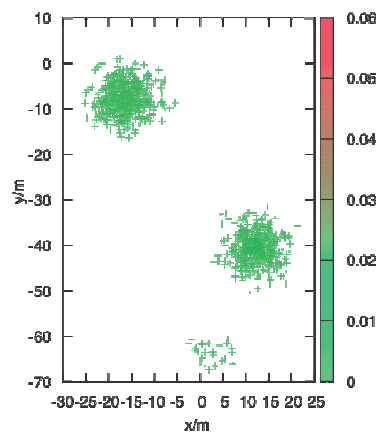

Fig. 4. State distribution after sequential tracking.

Fig. 5 shows on the right hand side the spatial distribution of objects detected especially at the side of the street. One the left hand side one can see the original, already registered lidar scan. Since it was mainly the central barrier and grass which was wrongly detected as an object, the spatial elongation is rather high.

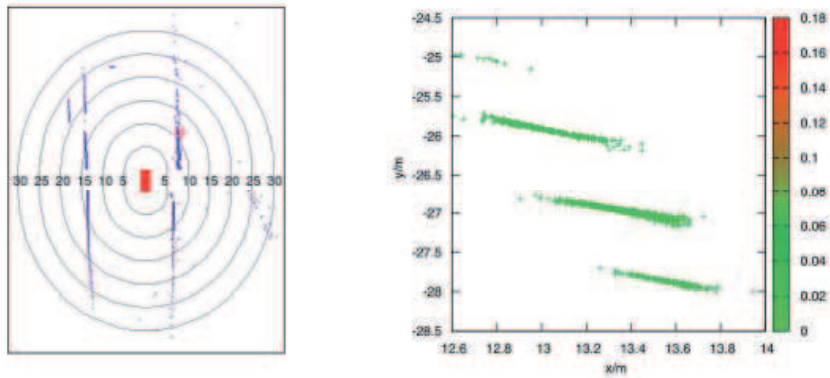

Fig. 5. State distribution of a street surrounded by natural objects.

Finally, Fig. 6 shows the error for one multi-modal tracked object. The chart clearly shows the converging phase followed by a phase of smaller error.

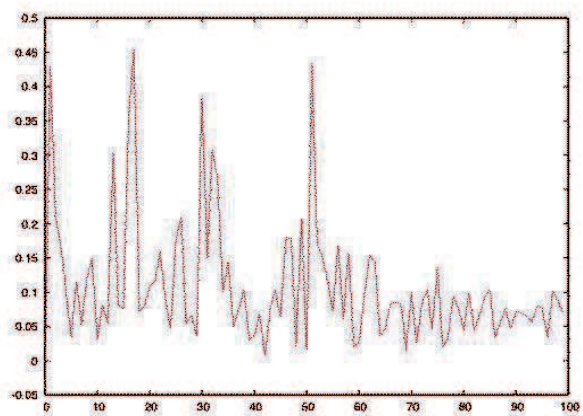

Fig. 6. Error in meters plotted over iterations of tracked object.

8. Conclusion and outlook

In this contribution, we introduced a novel approach of multi-modal object tracking.

Starting from the sensor setup, we introduced our temporal and spatial registration of the lidar data. To achieve a global description of the current scene, we transform all the data into a fixed world coordinate system. In addition, this ensures that the assumed object model has only to cover the dynamic behavior of the tracked object. The ego-movement does not affect the object measurement in any way.

Thereafter, the probabilistic fundamentals for object tracking were characterized. Based upon this, the Monte Carlo approach for solving this problem was presented.

The next paragraph described our assumed process model. For the observation model, we introduced the novel multi-modal weight function, which combines all the detected objects.

The feature extraction stage completed all the ingredients needed for the tracking. Thereafter, the results section showed the performance of our approach. To this end, we gave an overview over the initial spatial distribution. Starting from that point, the objects representing clusters are shaped by the filter. In addition, we gave a distribution for a scenario

in which only typical off-street objects were detected, which can be identified thanks to their large spatial dimensions.

Acknowledgment

The authors gratefully acknowledge support of this work by the Deutsche Forschungsgemeinschaft (German Research Foundation) within the Transregional Collaborative Research Centre 28 “Cognitive Automobiles”.

References

[1] C. Hoffman, T. Dang, C. Stiller: “Vehicle detection fusing 2D visual features”. *IEEE Intelligent Vehicles Symposium*, June 2004, pp. 280-285.

[2] T. Dang: “An iterative parameter estimation method for observation models with nonlinear constraints”. *Metrol Meas Syst*, vol. 15, no. 4, 2008, pp. 421-432.

[3] S. Kammel, J. Ziegler, B. Pitzer, M. Werling, T. Gindele, D. Jagzent, J. Schröder, M. Thuy, M. Goebl, F. von Hundelshausen, O. Pink, C. Frese, C. Stiller: “Team Annieway’s autonomous system for the DARPA Urban Challenge 2007”. *International Journal of Field Robotics Research*, vol. 25, no. 9, 2008, pp. 615-639.

[4] M. Thuy, A. Saber Tehrani, F. Puente León: “Bayessche Fusion von Stereobildfolgen und Lidardaten”. *Bildverarbeitung in der Mess- und Automatisierungstechnik*. VDI-Berichte, VDI Verlag, Düsseldorf, 2007, pp. 67-78.

[5] S. Wender K. Dietmayer: “A feature level fusion approach for object classification”. *IEEE Intelligent Vehicles Symposium,* June 2007, pp. 1132-1137.

[6] A. Mendes, L. Bento, U. Nunes: “Multi-target detection and tracking with a laser scanner”. *IEEE Intelligent Vehicles Symposium,* June 2004, pp. 796-801.

[7] M. Thuy, F. Puente León: “Non-linear, shape independent object tracking based on 2D lidar data”. *IEEE Intelligent Vehicles Symposium,* June 2009, pp. 532-537.

[8] M. Marron, J. Garcia, M. Sotelo, D. Fernandez, D. Pizarro: “xpfcp”: An extended particle filter for tracking multiple and dynamic objects in complex environments”. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug. 2005*,* pp. 2474-2479.

[9] E.B. Koller-Meier, F. Ade: “Tracking multiple objects using the condensation algorithm”. *Robotics and Autonomous Systems*, vol. 31, 2001, pp. 93-105.

[10] M. Marron, J. Garcia, M. Sotelo, M. Cabello, D. Pizarro, F. Huerta, J. Cerro: “Comparing a Kalman filter and a particle filter in a multiple objects tracking application”. *IEEE International Symposium on Intelligent Signal Processing*, Oct. 2007, pp. 1-6.

[11] M. Goebl, M. Althoff, M. Buss, G. Farber, F. Hecker, B. Heissing, S. Kraus, R. Nagel, F. Puente León, F. Rattei, M. Russ, M. Schweitzer, M. Thuy, C. Wang, H. Wuensche: “Design and capabilities of the munich cognitive automobile”. *IEEE Intelligent Vehicles Symposium,* June 2008, pp. 1101-1107.

[12] M. Thuy, M. Althoff, M. Buss, K. Diepold, J. Eberspächer, G. Färber, M. Goebl, B. Heißing, S. Kraus, R. Nagel, Y. Naous, F. Obermeier, F. Puente León, F. Rattei, C. Wang, M. Schweitzer, H.-J. Wünsche: “Cognitive Automobiles – new concepts and ideas of the transregional collaborative research centre TR-28”. *Aktive Sicherheit durch Fahrerassistenz,* Garching, Germany, April 2008.