Non-Linear, Shape Independent Object Tracking based on 2D Lidar Data

Michael Thuy, Fernando Puente León
Universität Karlsruhe (TH)
Institut für Industrielle Informationstechnik
Hertzstraße 16, D-76187 Karlsruhe
Germany
E-mail: {thuy, puente}@iiit.uni-karlsruhe.de

***Abstract*—The paper presents a new lidar-based approach to object tracking. To this end, range data are recorded by two vehicle-born lidar scanners and registered in a common coordinate system. In contrary to common approaches, particle filters are employed to track the objects. This ensures no linearization of the underlying non-linear process model and, thus, a decreasing estimation error. For the object association, a new method is proposed that considers the knowledge about the object shape as well. Based on a statistical formulation, this ensures a robust object assignment even in ambiguous traffic scenes.**

I. Introduction

Object detection is one of the key abilities of modern driver assistance systems. A vast literature on this subject exists, and different sensors (radars, lidars, monocular and stereo cameras, etc.) have already been employed to approach a robust object detection. Generation of object hypotheses with a monocular camera was e.g. shown in [1]. The same problem has been faced with stereo cameras; see e.g. [2], [3].

Since image acquisition and the subsequent image processing often suffers from varying ambient light conditions [4], the concurrent use of complementary sensing principles can yield a drastical improvement of system robustness. Using active sensors like lidar scanners breaks the limitation of the surrounding light condition through the integrated light source [5], [6]. The sensor itself sends out at a specified angle a laser impulse and measures the time interval until the reflected light is registered. Besides the very low dependence of the environmental light condition, the sensors only deliver a distance measurement if the light hits an object. Consequently, every given distance measurement is a potential part of an object and must be considered.

Our contribution presents a novel approach to object detection and tracking based on two 2D lidar scanners. This proceeding assumes an adequate dynamic model behind the object. As we mainly focus on object detection and tracking of surrounding vehicles, the physically underlaying process model is characterized by a non-linear behaviour. Many published articles like [7] therefore use an linear or extended Kalman filter, which—in case of the extended version—linearizes the process model within the current operating point. Different to that approach, we use Monte-Carlo-based particle filters for the tracked objects. Since such filters are able to represent non-linear systems, the resulting estimation error drops [8]. In spite of the higher computation time, we show that the necessary calculations can still be done online.

Besides an adequate tracking, an assignment of known objects to newly detected objects has to be performed. As shown in [9], this task can be faced by a center-of-gravity or a reference-point approach. Then, the assignment is mainly based on the minimum distance between the object's predicted and the newly measured reference points. However, especially in dense traffic scenarios, this method gets ambiguous, as this method requires a distinct minimum distance. Our method solves this problem by a correlation-based algorithmwhich yields a measure of shape conformity. Since the shape—assuming a sufficient sampling rate—of an object does not significantly change frame by frame, comparing a learned and stored shape to newly detected shapes results in a robust object association.

The paper is organized as follows: Section II introduces the sensor setup mounted on the test vehicle. It infers a global data registration based on a fixed world coordinate system. Section III introduces the chosen Monte Carlo approach which leads to the particle filter. The next section IV introduces the underlying process model. The following section V builds up the filter. After section VI, which describes the segmentation process, the association is described. Section VIII presents association and tracking results with the underlying particle filtering. Section IX concludes this paper with a short summary of our results.

II. Sensor Setup and Global Data Registration

For the further development and understanding, it is important to characterize the employed sensor setup. For the measurements, we use two lidar scanners. The first one is mounted on the front bumper, the second is mounted at the rear bumper. If the car is standing still, their respective scanning planes are parallel to the ground. Figure 1 shows a bird's eyes view of the sensor setup. Both lidar scanners sample their environment with an angular resolution of a half degree.

The next important item is the registration of the lidar data, i.e. the alignment of the measured lidar points with regards to time and space. Since the two scanners are synchronized by hardware, the delivered data is already synchronized in time.

978-1-4244-3504-3/09/$25.00 ©2009 IEEE

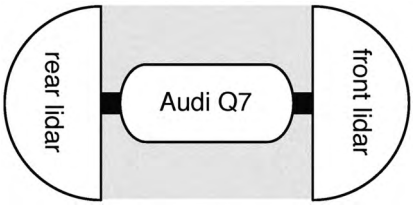

Fig. 1. Sensor setup.

As it is possible to calibrate the two sensors in respect to the ego-car's reference point, from the first the lidar data is describing the surrounding relative to the ego-car. Tracking the other objects relative to the ego-car drastically degrades the filter convergence. This is due to the fact that the assumed object process model must then also cover the ego-movement. In regard of a non-linear proces model, it can even lead to non-converging filter and, thus, to a loss of the object track.

For that reason, we incorporate an inertial measurement unit (IMU) combined with a high-precision differential global positioning system (DGPS). This unit delivers the current position described in latitude, longitude and altitude as well the current heading. Since the sampling rate of the IMU is one dimension higher than the actual scanning rate, the delivered spatial resolution is sufficient without any interpolation. Including this information, we can perform all the tracking in a fixed world coordinate system. Therefore, we transform the given ego-position into UTM coordinates and, from that point on, transform all the relative positions into absolute UTM coordinates.

Let $(x_{\text{ego}}, y_{\text{ego}})$ denote the absolute ego-position and $(x_{\text{scan},i}, y_{\text{scan},i})$ the position of the i-th scan point. Then, we can calculate the absolute coordinates of the lidar points as follows:

$$\begin{pmatrix} x_{\text{fix},i} \\ y_{\text{fix},i} \end{pmatrix} = \begin{pmatrix} x_{\text{ego}} \\ y_{\text{ego}} \end{pmatrix} + \mathbf{R} \cdot \begin{pmatrix} x_{\text{scan},i} \\ y_{\text{scan},i} \end{pmatrix}, \tag{1}$$

where $\mathbf{R}$ denotes the rotation matrix due to the ego-heading. So, all the the coordinates given to the filter are absolute coordinates. Consequently, the tracked object velocity also describes the absolute speed and can be used for the classification of a moving or standing object.

III. Basics on Monte Carlo Simulation

A. General formulation of the a-posterior density distribution

The Monte Carlo simulation can be seen as a particular solution of the general predicting and tracking problem, i.e. the estimation of the current state vector based on all available observations. 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 vectors $\mathbf{x} = [x_1, x_2, \ldots, x_n]^{\mathrm{T}}$ for all points in time until the curent time t:

$$\mathbf{X}_t = \{\mathbf{x_0}, \mathbf{x_1}, \ldots, \mathbf{x_t}\}\,. \tag{3}$$

Analogously, $\mathbf{Z}_t$ means the observation vector $\mathbf{z} = [z_1, z_2, \ldots, z_n]^{\mathrm{T}}$ for all time points until the current time t:

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

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

$$p(\mathbf{x}_t|\mathbf{Z}_t) = \frac{p(\mathbf{z}_t|\mathbf{x}_t) \cdot p(\mathbf{x}_t|\mathbf{Z}_{t-1})}{p(\mathbf{z}_t|\mathbf{Z}_{t-1})}\,. \tag{5}$$

For the update, one can determine the prediction density distribution as follows:

$$p(\mathbf{x}_t|\mathbf{Z}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) \cdot p(\mathbf{x}_{t-1}|\mathbf{Z}_{t-1}) \cdot \mathrm{d}\mathbf{x}_{t-1}\,. \tag{6}$$

Now, from the theoretical point of view, the problem is solved. But a look at equation (6) reveals a complex multi-dimensional 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.

B. Monte Carlo approach

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

Using N particles with their state vectors $\{\mathbf{x}^i_{0:t}, i = 0, ..., N\}$ one can infer the corresponding weights as follows:

$$w^i_t \sim \frac{p(\mathbf{x}^i_{0:t}|\mathbf{z}_{1:t})}{q(\mathbf{x}^i_{0:t}|\mathbf{z}_{1:t})}\,, \tag{7}$$

whereas $q(.)$ represents the importance density. But still we have to incorporate all the observations $\mathbf{z}_{1:t}$ till time t.

A further proceeding finally yields to an iterative description of the weight:

$$w^i_t \sim w^i_{t-1} \cdot \frac{p(\mathbf{z}_t|\mathbf{x}^i_t)\, p(\mathbf{x}^i_t|\mathbf{x}^i_{t-1})}{q(\mathbf{x}_t|\mathbf{x}^i_{t-1}, \mathbf{z}_t)}\,. \tag{8}$$

Now the choice of the importance density gets the most crucial task. Assuming

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_{1:t}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})\,, \tag{9}$$

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

$$w^i_t \sim p(\mathbf{z}_t|\mathbf{x}^i_t)\,. \tag{10}$$

Equation (10) means that the corresponding particle weight w^i_t can be calculated with the help of the likelihood function $p(\mathbf{z}_t|\mathbf{x}^i_t)$. As all the calculations can now be done sequentially, the filter is named Sequential Importance Sampling filter.

So after all, an initial state distribution—represented through 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 up to the corresponding particle weight

w_t^i. But what now can happen is a depletion of the particles: many particles tend to weights nearly zero. This leads to an unbalanced sampled posterior distribution. A resampling strategy prevents this development. To this end, after the measurement update, the particles with low weight are omitted, whereas particles with high weights are duplicated.

IV. Dynamic Obstacle Model

After having introduced the fundamentals, the underlaying process function will be presented. First, we have to define our state vector $\mathbf{x}$:

$$\mathbf{x} = (x, y, v, a, \beta)^{\mathrm{T}} \,, \tag{11}$$

where (x, y) represents the object reference point, v the object velocity, a its acceleration and β the yaw angle.

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

$$\begin{aligned} \mathbf{x}[t+1] = \quad & \mathbf{x}[t] \quad + \begin{pmatrix} v[t] \cdot \cos(\beta[t]) \\ v[t] \cdot \sin(\beta[t]) \\ a[t] \\ 0 \\ 0 \end{pmatrix} \Delta t + \\ & + \quad \begin{pmatrix} 0{,}5 \cdot a[t] \cdot \cos(\beta[t]) \\ 0{,}5 \cdot a[t] \cdot \sin(\beta[t]) \\ 0 \\ 0 \\ 0 \end{pmatrix} \Delta t^2 \,, \end{aligned} \tag{12}$$

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

Assuming this process model as the object movement characteristic, no object can go left or 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 a non-accurate object detection when a tracked object is split into two objects.

V. Filter Design

As now all the basic conditions have been cleared, we can proceed with designing the whole filter for a tracked object.

A. 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 determination 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 (10), one has to find the characteristic likelihood function for the defined observation values. For a lidar sensor, one often uses a Gaussian noise model. Continuing this assumption leads to the desired two-dimensional likelihood function, which only depends on the extracted x_{obj} and y_{obj} coordinates as model state variables:

$$\begin{aligned} w_t^i \quad &\sim \quad p(\mathbf{z}_t | \mathbf{x}_t^i) = \\ &= \quad A \cdot \exp\left(-\frac{(x_{\mathrm{obj}} - \hat{x})^2}{2\sigma_x^2} - \frac{(y_{\mathrm{obj}} - \hat{y})^2}{2\sigma_y^2} \right) . \end{aligned} \tag{13}$$

It should be emphasized that the only information needed by the filter is the absolute object position. All the other state variables are inferred from the process model. In our contribution, we did not model the sensor variance as a measurement-dependent value, since its influence is negligible. The variance itself is determined experimentally.

B. Noise-afflicted process model

As the process model according to equation (12) does not handle any process noise, we would assume an accurate process model that does not differ from the real physical behaviour of the objects. Due to model errors, we have to extend it by adding additional noise sources. In this case, we model the noise characteristics as a Gaussian noise pattern as well. To this end, we expand the equation to:

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

where the noise sources $\mathbf{n}$ are described by their corresponding variances σ^2.

C. Filter composition

Now, as all important basics for the particle filter are introduced, we can assemble the filter method. In order to track objects, one has to predict the defined object state vector at the future time $t+1$ from time point t. This is needed to associate an already known object from the past observations to a detected object in the new measurement. As already known from the Kalman tracking, one can separate this task into two stages: the prediction stage and the update stage.

As we now have spread the particles to cover the state space as a combination of a certain state vector $\mathbf{x}_i$ and a corresponding weight w_i, we propagate them in time through running the process model. This means taking the particle's state vector $\mathbf{x}_i$ and applying it to the process model according to equation (14). As for each specific state noise is added, the particles tend to scatter.

For the upcoming assigning process, one has now to find the estimated object position out of the new calculated particles. As it can be seen in the literature, this task can be solved by two different strategies. The first calculates the mean value of the posterior density sampled by the particles:

$$\hat{x} = \sum_i w_i \cdot x_i \quad \text{and} \quad \hat{y} = \sum_i w_i \cdot y_i \,. \tag{15}$$

The other method clusters the particles in the state space. But since we use the particle filter in an unimodal manner, we determine the estimated position by equation (15). Nevertheless, as we always add Gaussian noise, the resulting posterior distribution is Gaussian as well. So, assuming ideal conditions, both methods yield the same estimate.

As already mentioned in paragraph III-B, the particles tend to deplete. As a resampling mechanism, we analyzed two strategies. The first tries to omit particles with low weight based on a drawing without replacement experiment by duplicating the ones with higher weight. The second approach simply orders the particles by their weight and omits the lower half. But in both cases, after resampling, the weight is normalized to $1/N$, where N is the number of used particles.

Assuming a correct association between a newly detected object—which is described in detail in the next paragraph—and an already tracked object, it is then possible to update the filter with the new object coordinate. This is done by recalculating the weights w_i according to equation (13).

VI. Lidar Data Segmentation

The main task of the feature extraction stage is to determine the measurement values desired by the filter, i.e. to generate 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 a UTM description.

Getting objects like cars out of the lidar raw points can be solved by analyzing the distance of consecutive laser points. Since the laser rays are—even for long distances up to 80 m—really close to each other, an object silhouette yields to lidar points which are very close to each other as well. If other objects occur, 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 up with an 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 $r_{\mathrm{i}+1}$ and r_{i} yields the following formula:

$$\Delta D(r_{\mathrm{i}+1}, r_{\mathrm{i}}) = \sqrt{r_{\mathrm{i}}^2 + r_{\mathrm{i}+1}^2 - 2 r_{\mathrm{i}} r_{\mathrm{i}+1} \cos(\Delta\alpha)}\,, \qquad (16)$$

where the angle $\Delta\alpha$ denotes the constant and equally-spaced 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:

$$d_{\mathrm{th}}(r_{\mathrm{i}+1}, r_{\mathrm{i}}) = d_{\mathrm{th},0} + d_{\mathrm{th},1}(\Delta\alpha, r_{\mathrm{i}+1}, r_{\mathrm{i}})\,. \qquad (17)$$

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

$$d_{\mathrm{th},1}(\Delta\alpha, r_{\mathrm{i}+1}, r_{\mathrm{i}}) = \tan(\Delta\alpha) \cdot \min\{r_{\mathrm{i}+1}, r_{\mathrm{i}}\}\,. \qquad (18)$$

If the distance between two consecutive laser points is less than d_{th}, the points are associated to the same object. Otherwise, a new object starting point is assumed. 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 prior shape information has been incorporated.

To generate a reference point for each object, we calculate its center of gravity, which results in the object position $(x_{\mathrm{obj}}, y_{\mathrm{obj}})$.

VII. Data Association

In the last section, the generation of the object list containing the new detected objects was described. Now, still the task of assigning an already known and tracked object to one of the newly detected has to be performed. The easiest and fastest to achieve this is based on the distance. To this end, the predicted object position is compared to the positions of recently detected objects. If the smallest distance is below a given threshold, the right object was found. Despite more complex association methods exist, all of them are still based on the distance as the central criterion. Besides, there are already known alternatives which incorporate the object shape as well [10], but only for unknown shapes.

A. Iterative Closest Point method

In our contribution, we incorporate the shape as a additional criterion to compare the equality of an already known object and a newly detected one. The known ICP algorithm is used to minimize the distance between point clouds. As we need to calculate the shape correlation in real time, we had to modify the method according to the current problem. As the ICP error function, we define the scalar function:

$$f(\mathbf{t}) = \frac{1}{N} \sum_{i=1}^{N} \sqrt{(x_{\mathrm{obj},i} - x_{z,j}(\mathbf{t}))^2 + (y_{\mathrm{obj},i} - y_{z,j}(\mathbf{t}))^2}\,, \qquad (19)$$

which is the average error of the distance between the nearest shape points. Calculating the average error ensures a comparison of the resulting error independent of the amount of available object points. Since the scanning rate is high, the shape rotation between two scans does not differ drastically. Consequently, to minimize the error, we only consider a shape translation represented by the vector $\mathbf{t}$. Calculating the derivative $\mathrm{d}/\mathrm{d}\mathbf{t}$ of equation (19) leads to the gradient, which marks the desired translation to minimize the error:

$$\begin{aligned} x_{z,j,\mathrm{new}} &= x_{z,j} - \mu \frac{\partial f(\mathbf{t})}{\partial t_x}\,, \\ y_{z,j,\mathrm{new}} &= y_{z,j} - \mu \frac{\partial f(\mathbf{t})}{\partial t_y}\,, \end{aligned} \qquad (20)$$

where μ represents the translation coefficient. As is shown in the results, it controls the speed of convergence.

In the case of scan point outliers, minimizing the error function can lead to non-convergence. Therefore, these outliers have to be removed from the newly detected object. Following the work from [11], this task can be done by a method that incorporates the point distribution as well.

Now, with these modifications, the ICP algorithm is performed iteratively, until the residual does not differ significantly anymore.

B. Probabilistic association

In general, one can correlate all the newly detected objects with one already known object to find the right association. But, for the sake of online computation and due to a finite movement of objects, we restrict the analysis area. To this end, we build up an ellipse around the predicted object position. Increasing the size of the ellipse leads at the end to the just mentioned overall solution.

Let O_j denote the already j-th tracked object. B_i represents the i-th observed object. Then $p_\mathrm{s}(O_j \rightarrow B_i)$ characterizes the probability that the object O_j does not belong to object B_i due to shape non-conformity. Analogously, $p_\mathrm{d}(O_j \rightarrow B_i)$ describes the probability that the two objects cannot be assigned because of their distance. By the help of the knowledge from the paragraphs above, we can now determine the probabilities:

$$p_\mathrm{s}(O_j \rightarrow B_i) = \frac{d_{\mathrm{shape},i,j}}{\sum_{k=1}^{N}\sum_{l=1}^{M} d_{\mathrm{shape},k,l}} \tag{21}$$

and

$$p_\mathrm{d}(O_j \rightarrow B_i) = \frac{d_{\mathrm{dist},i,j}}{\sum_{k=1}^{N}\sum_{l=1}^{M} d_{\mathrm{dist},k,l}} \,. \tag{22}$$

Here, $d_{\mathrm{shape},i,j}$ means the minimum average error, determined by formula (19). $d_{\mathrm{dist},i,j}$ represents the distance between the reference points of the i-th and j-th object. Taking these two probabilities, we can now define the overall assigning probability:

$$p(O_j \rightarrow B_i) = c \cdot p_\mathrm{s}(O_j \rightarrow B_i) + (1-c) \cdot p_\mathrm{d}(O_j \rightarrow B_i)\,, \tag{23}$$

in which c controls the influence of the shape correlation and has to be adjusted in the range of $[0;1]$. By adjusting the constant, one can continuously control the association from a distance based to shaped based approach. If the distance or the average matching error exceeds a certain threshold, the probability is set to one in order to prevent unrealistic associations.

After determining the assigning probabilities, one can now make the association decision. This means the search for the lowest probability for each detected object.

VIII. Tracking Results

The following paragraph shows the tracking result of the presented approach.

A. Particle filter

To investigate the performance of the particle-based approach, we contrasted it with a Kalman filter for a track without any wrong association. Figure 2 shows the resulting error for the particle approach. Figure 3 shows the error for the same track with an underlying Kalman filter. Here, one can see clearly the convergence phase. In contrary, the particle filter does not show this behaviour that significantly, but the error is always smaller than that of the Kalman filter. In this example, the particle filter error is always 80 % smaller than the Kalman filter error.

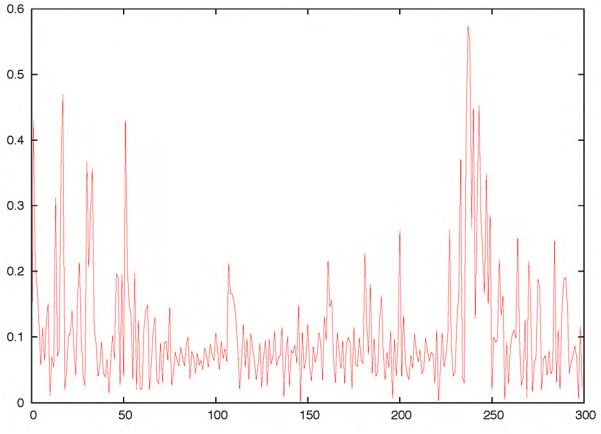

Fig. 2. Particle filter error in metres over timesteps.

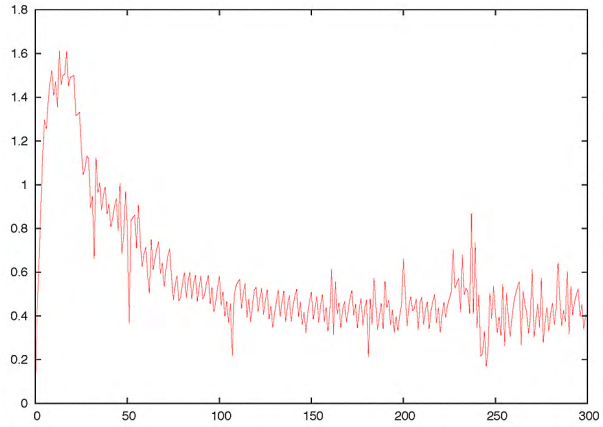

Fig. 3. Kalman filter error in metres over timesteps.

B. Association

Figure 4 shows a detected car within a traffic scene. On the left hand side, the predicted (green color) and the newly detected (red color) shapes a shown within the same chart. On the right hand side, the result out of the modified ICP is illustrated. Especially in the lower part of the car one can see the resulting shift. Since the two shapes match very well, the ICP converges after approximately ten iterations with a very small residual.

Figure 5 shows already another correlation result. In contrary to the first one, the two shapes differ significantly in one part due to an occlusion. But still, the common part is matched the right way.

The next figure 6 illustrates the resulting error plotted against the performed iteration steps of the last two shapes. Dependent on the choice of the shift coefficient μ out of equation (20), one can see a periodic behaviour. In general, decreasing the coefficient leads to a longer computational time but decreases the oscillation around the remaining residual.

IX. Conclusion

We have presented a method that fulfills all requirements to perform an object tracking based on one-dimensional lidar

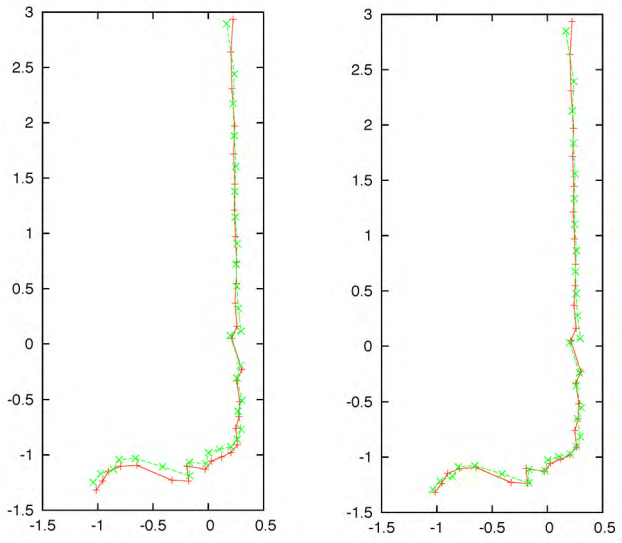

Fig. 4. Two L-shapes of a car.

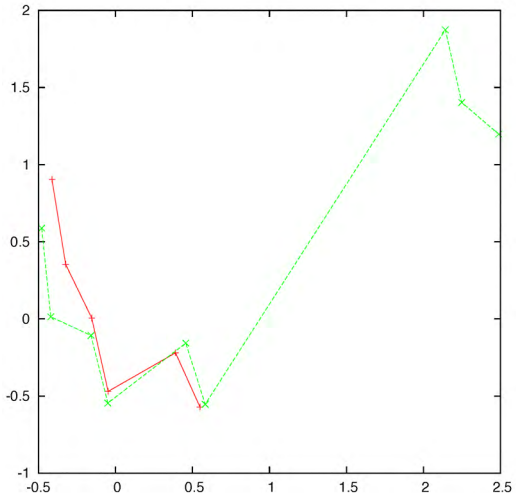

Fig. 5. Two L-shapes of a car.

data. First, the basics of the chosen Monte Carlo approach that leads to the applied particle filter have been presented. Afterwards, we have introduced a non-linear process model and the corresponding observation model. Furthermore, all noise sources within the process chain have been characterized.

After the composition of the resulting filter, modifications of the ICP method have been introduced. Based on this, a new association strategy has been proposed.

At the end, tracking results have demonstrated the performance and accuracy of the presented filter. To this end, the implemented particle filter has been compare with a Kalman filter approach. Together with a shape- and distance-based association, a robust tracking system could be achieved.

Future work will incorporate the theory of a joint probabilistic data association filter (JPDAF) for the association stage. Furthermore, the particle filters will be extended to consider also multimodal distributions.

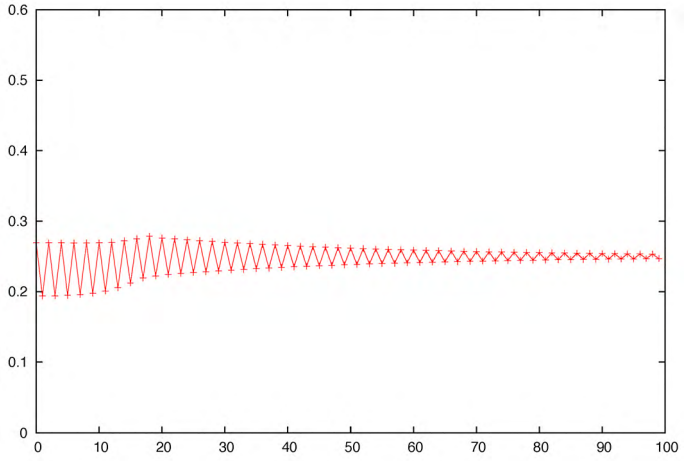

Fig. 6. ICP residual in metres over iteration index.

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. Hoffmann, T. Dang, and C. Stiller, “Vehicle detection fusing 2d visual features,” in *Proc. IEEE Intelligent Vehicles Symposium*, 2004, pp. 280–285.

[2] M. Thuy and A. Sandmair, “Object classification based on stereo data,” in *Reports on Distributed Measurement Systems*, F. Puente León, Ed. Shaker Verlag, 2008, pp. 141–154.

[3] A. Bachmann and T. Dang, “Multiple object detection under the constraint of spatiotemporal consistency,” *Intelligent Transportation Systems Conference, 2006. ITSC ’06. IEEE*, pp. 295–300, Sept. 2006.

[4] F. Rattei, M. Goebl, and G. Färber, “Beitrag zur Robustheitssteigerung videobasierter Fahrerassistenzsysteme durch frühe Rückkopplungen zur Sensorebene,” in *Bildverarbeitung in der Mess- und Automatisierungstechnik*. VDI-Berichte, VDI Verlag, Düsseldorf, 2007.

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

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

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

[8] J. Almeida and R. Araujo, “Tracking multiple moving objects in a dynamic environment for autonomous navigation,” *Advanced Motion Control, 2008. AMC ’08. 10th IEEE International Workshop on*, pp. 21–26, March 2008.

[9] A. Kapp, “Ein Beitrag zur Verbesserung und Erweiterung der Lidar-Signalverarbeitung für Fahrzeuge,” Ph.D. dissertation, Institut für Mess- und Regelungstechnik, Universität Karlsruhe (TH), 2007.

[10] N. Kämpchen, M. Bühler, and K. Dietmayer, “Feature-level fusion for free-form object tracking using laserscanner and video,” *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pp. 453–458, June 2005.

[11] Z. Zhang, “Iterative point matching for registration of free-form curves,” *Rapports de Recherche, Institut National de Recherche en Informatique et en Automatique*, 1992.