Object Motion Analysis and Prediction in Stereo Image Sequences

Christoph HERMES, Alexander BARTH, Christian WÖHLER and Franz KUMMERT

Abstract

Future driver assistance systems will have to cope with complex traffic situations, especially at intersections. To detect potentially hazardous situations as early as possible, it is therefore desirable to know the position and motion of oncoming vehicles for several seconds in advance. For this purpose, we propose a combined approach that tracks the vehicle position and orientation over time based on a box model, where the vehicle motion state is predicted several seconds ahead based on simultaneous tracking of multiple hypotheses with a particle filter framework. The scene is observed by a stereo camera mounted on the ego-vehicle. Compared to a traditional constant acceleration and curve radius prediction model, we show that the accuracy of the proposed particle filter approach is superior during turning manoeuvres displaying complex motion patterns.

1 Introduction

Future driver assistance systems have to be able to interpret complex traffic situations, for example at intersections. Predicting the trajectories of other traffic participants is an essential task for many applications such as collision avoidance. The aim is to detect critical situations as early as possible to warn the driver or to induce an autonomous safety action.

A method for estimating the pose and motion state of vehicles using a stereo vision sensor has been proposed in (BARTH 2008). In this approach, objects are represented as rigid 3D point clouds and tracked by an Extended Kalman Filter. The movement of the point cloud is restricted to circular path motion, assuming constant acceleration, which is an adequate assumption for time intervals of about one second. However, this motion model is insufficient if it is desired to predict the future trajectory, i.e. future pose and motion states, of a tracked object several seconds ahead.

Humans are able to predict object movements based on a short motion sequence using an expectation of typical motion patterns. For example, an oncoming vehicle at an intersection that starts changing its orientation is likely to turn left or right, depending on the direction of the orientation change. In this contribution, we adopt the human capability of inferring potential movements from a short motion sequence, which is extracted using an extended version of the method proposed in (BARTH 2008), allowing for reliable predictions up to three seconds ahead.

SIDENBLADH et al. (2002) utilize a particle filter on a given trajectory set for a probabilistic search to predict and track human body motion. We will extend this idea to estimate the probability density function of the future trajectory of a vehicle. Within this approach, trajectories are compared to reference trajectories using a rotationally invariant distance metric.

2 Object Motion Estimation

The following object motion estimation method is used both for generating reference trajectories of typical driving manoeuvres and for estimating the motion state of an object for prediction at runtime.

2.1 Object Representation

A vehicle is represented by a rigid 3D point cloud attached to a local *object coordinate system*. The origin of the object coordinate system is defined at an arbitrary reference point, e.g. the centroid of the point cloud, projected to the ground plane (street). ${}^{o}x$, ${}^{o}y$ and ${}^{o}z$ represent the lateral axis, the height axis, and the longitudinal axis, respectively. The *ego coordinate system* is defined at the centre rear axis of the ego-vehicle (see Fig. 1).

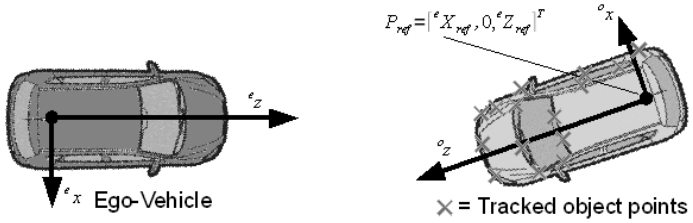

Fig. 1: Coordinate systems and pose parameters (bird's eye view)

The unknown parameters are the pose, i.e. position $\left({}^{e}X_{ref}, {}^{e}Z_{ref}\right)$ and orientation (ψ in deg) with respect to the ego-vehicle, the motion state including velocity (v in m/s), acceleration (a in m/s²), and yaw rate ($\dot{\psi}$ in deg/s), as well as the exact structure of the 3D point cloud. These parameters are estimated using a standard Extended Kalman Filter (EKF) approach. In the following we will only present the main elements required for tracking, including state vector, system model and measurement model. Details regarding Kalman filtering can be found, for example, in (BAR-SHALOM 2001). The state vector of the unknown parameters x is defined as:

$$x = [{}^{e}X_{ref}, {}^{e}Z_{ref}, \psi, v, \dot{\psi}, a, {}^{o}X_0, {}^{o}Y_0, {}^{o}Z_0, \ldots, {}^{o}X_{M-1}, {}^{o}Y_{M-1}, {}^{o}Z_{M-1}] \quad (1)$$

We will denote ${}^{e}P_{ref} = \left[{}^{e}X_{ref}, 0, {}^{e}Z_{ref}\right]^T$ as the reference point in the following. The main difference to the approach in (BARTH 2008) is that the object point coordinates ${}^{o}P_m = \left[{}^{o}X_m, {}^{o}Y_m, {}^{o}Z_m\right]^T$, $0 \le m < M$, are also included in the filter state, i.e. the filter also estimates the structure of the (noisy) point cloud.

2.2 System Model

The non-linear system model f describes the dynamics of a tracked object and is used for predicting a state a given time interval Δt ahead, i.e. $\hat{x}(t+\Delta t) = f\left(x(t)\right)$, where $\hat{x}$ denotes

the predicted filter state before the measurements are incorporated. Vehicle movements are restricted to mostly longitudinal, or circular path movements, at normal driving conditions. Here, a constant curve radius and acceleration model is applied. In a time-discrete formulation the change of state x can be written as $f(x(t)) = x(t) + \Delta x$ with

$$\Delta x = [{}^{e}X_{ref}, {}^{e}Z_{ref}, 0, 0, \dot{\psi}\Delta t, a\Delta t, 0, 0, \underbrace{0, \ldots, 0}_{3M}] \tag{2}$$

The centre rear axis plays an important role at rotational movements as it is the rotational reference centre (rotation point). The predicted reference point position P'_{ref} is computed as follows (in homogeneous representation):

$$P'_{ref} = M_{ego} M_{e2o} M_o \left({}^{o}P_{ref}\right) \tag{3}$$

First, M_o transforms the object origin ${}^{0}P_{ref} = [0,0,0]^T$ within the object coordinate system following a circular path motion model with

$$M_o = \begin{bmatrix} R(\dot{\psi}\Delta t) & {}^{o}T \\ 0 & 1 \end{bmatrix} \tag{4}$$

and

$${}^{o}T = \left[\frac{v + a\Delta t}{\dot{\psi}} (1 - \cos(\dot{\psi}\Delta t)), \frac{v + a\Delta t}{\dot{\psi}} \sin(\dot{\psi}\Delta t) \right]^T. \tag{5}$$

$R(\alpha)$ denotes a 3x3 rotation matrix around the height axis by an angle α. Then, M_{e2o} transforms the new position in object coordinates to the previous ego system:

$$M_{e2o} = \begin{bmatrix} R^{-1}(\psi) & {}^{e}P_{ref} \\ 0 & 1 \end{bmatrix} \tag{6}$$

Finally, the motion of the ego vehicle between the previous and the current frame is compensated by M_{ego} using the approach proposed in (BADINO 2004).

Thus, $\Delta P_{ref} = \left[\Delta^{e}X_{ref}, 0, \Delta^{e}Z_{ref}\right]^T = P'_{ref} - P_{ref}$ in Eq. (2).

2.3 Measurement Model

The measurement vector z consists of M triples (u_m, v_m, d_m), $0 \le m < M$, with (u_m, v_m) the subpixel accurate image position of an observed object point ${}^{o}P_m$ and d the image disparity between left and right image of a rectified stereo image pair. The image position of each object point is tracked in the image using the KLT feature tracker (TOMASI et al. 1991).

The non-linear measurement model h computes a predicted measurement vector $\hat{z}$ based on the predicted state estimate $\hat{x}$, i.e. $\hat{z} = h(\hat{x})$. It results from the perspective camera model:

$$\begin{aligned} u_m &= h_{m,1}(\hat{x}) = f_u \frac{{}^cX_m}{{}^cZ_m} + u_0 \\ v_m &= h_{m,2}(\hat{x}) = -f_v \frac{{}^cY_m}{{}^cZ_m} + v_0 \\ d_m &= h_{m,3}(\hat{x}) = f_u \frac{b}{{}^cZ_m}, \end{aligned} \tag{7}$$

where f_u and f_v are the horizontal and vertical focal length of the camera, (u_0, v_0) the principal point, and b the base line of the stereo system.

${}^cP_m = \left[{}^cX_m, {}^cY_m, {}^cZ_m\right]^T$ is the point ${}^oP_m$ in camera coordinates. The total transformation is composed of an object to ego transformation (parameterized by the state variables ψ, ${}^eX_{ref}$, and ${}^eZ_{ref}$), and the constant transformation between ego and camera system (including the extrinsic camera parameters).

For each measurement, a 3x3 noise matrix $R_m = diag(\sigma_u, \sigma_v, \sigma_d)$ has to be provided, yielding the total measurement noise matrix $R = blkdiag(R_0, R_1, \ldots, R_{M-1})$. Here, the measurement noise is assumed to be uncorrelated and constant.

2.4 Object Detection and Filter Initialization

Before tracking can be started, an object has to be detected. Here, a method fusing stereo and optical flow to track single 3D points in the scene, named 6D vision (FRANKE et al. 2005) is used. For each point, a 3D position and 3D velocity vector is estimated by means of Kalman filtering.

A group of points within a local neighbourhood, moving in the same direction with equal velocity, is assumed to belong to the same object. Clustering of the 6D vision data yields candidate objects. Such candidates consist of a set of 3D points $\left\{{}^eP_0, \ldots, {}^eP_{M-1}\right\}$ in ego coordinates and an average velocity vector $\overline{V} = \left[v_x, v_y, v_z\right]^T$. For each candidate object a new filter state, as proposed in Section 2, is initialized as follows:

$$x = \left[{}^eC_x, {}^eC_z, 0, 0, \psi_0, |\overline{V}|, 0, 0, {}^oX_0, {}^oY_0, {}^oZ_0, \ldots, {}^oX_{M-1}, {}^oY_{M-1}, {}^oZ_{M-1}\right]^T \tag{8}$$

with ${}^eC = \left[{}^eC_x, 0, {}^eC_z\right]^T$ the centroid of the initial point cloud in ego coordinates, and $\psi_0 = \arccos\left(v_z / |\overline{V}|\right)$ the angle of the average moving direction, defining the initial object

coordinate system. ${}^{o}X_m, {}^{o}Y_m, {}^{o}Z_m$, $0 \le m < M$, correspond to the coordinates of point ${}^{e}P_m$ in this object system.

It is possible to restrict the initialization method to objects exceeding a certain velocity threshold, motion direction, or dimensional constraint.

3 Motion Prediction

The main idea for motion prediction is that we can infer the future trajectory from the currently estimated trajectory, i.e. from the object tracking results up to the current time step. The future states of similar reference trajectories are taken as hypotheses for the current trajectory.

3.1 Similarity of Trajectories

A trajectory $X = \left((x(t_1), t_1), \ldots, (x(t_N), t_N) \right)$ is given as a series of object or vehicle states x_i with a time stamp t_i. Here, $x_i = \left[{}^{w}X_{ref}, {}^{w}Z_{ref}, \psi, v, \dot{\psi}, a \right]^T$ denotes a truncated version of state x (see Eq. (1)). Note that the position, corresponding to the centre rear axis, is given in *world coordinates,* i.e. a constant coordinate system outside the ego system.

First we define a metric to be able to compare trajectories. The following requirements have to be considered: (i) Insensitivity to outliers, since noisy data are likely to occur;
(ii) different lengths of trajectories, i.e. different motion patterns do not depend on the starting point; (iii) translational invariance, i.e. similar motion patterns do not depend on the starting point; (iv) rotational invariance, i.e. similar motion patterns do not depend on the orientation, and their comparison needs to be independent of the observer's viewpoint.

The longest common sub-sequence (LCS) metric (VLACHOS et al. 2005) on trajectories has been shown to be an adequate metric and can handle the first two stated requirements. It originates in the field of string matching algorithms and returns the length of the longest common sub-string matched by two strings. To apply this technique to trajectories, a similarity matching function between two states (points) a_i and b_j from the given trajectory points $\left(a_i, t_i^{(a)}\right) \in A$ and $\left(b_j, t_j^{(b)}\right) \in B$ has to be defined. VLACHOS et al. (2005) use the minimum standard deviation $std_{min}^{(dim)} = min\left\{ std\left(A^{(dim)}\right), std\left(B^{(dim)}\right) \right\}$ in each dimension dim as a decision boundary and apply a sigmoid function to smooth the distance value in the range $\left[0, std_{min}^{(dim)}\right]$. In our approach it is sufficient to use a linear function to obtain the distance between a_i and b_j, where $L^1(\cdot)$ denotes the L1 norm (Manhattan distance):

$$dist\left(a_i, b_j\right) = \begin{cases} 0 & if\ \exists\, dim \in D : L^1\left(a_i^{(dim)}, b_j^{(dim)}\right) > std_{min}^{(dim)} \\ \dfrac{1}{D} \sum_{dim=1}^{D} \left(1 - \dfrac{L^1\left(a_i^{(dim)}, b_j^{(dim)}\right)}{std_{min}^{(dim)}} \right) & otherwise \end{cases} \tag{9}$$

The sizes of the trajectories A and B are denoted by N_A and N_B, respectively, corresponding to the number of motion states they comprise, and the sequence $\left[\left(a_1, t_1^{(a)}\right), \ldots, \left(a_{N_A-1}, t_{N_A-1}^{(a)}\right)\right]$ by $head(A)$. We then define the LCS on trajectories as follows:

$$LCS(A,B) = \begin{cases} 0 & if\ N_A = 0 \wedge N_B = 0 \\ LCS(head(A), head(B)) + dist\left(a_{N_A}, b_{N_B}\right) & if\ dist\left(a_{N_A}, b_{N_B}\right) \neq 0 \\ max\{LCS(head(A), B), LCS(A, head(B))\} & otherwise \end{cases} \tag{10}$$

The distance between two trajectories A and B can then be obtained by $dist_{LCS}(A,B) = 1 - (LCS(A,B) / min\{N_A, N_B\})$, with $dist_{LCS}(A,B) \in [0,1]$. In order to get the translational and rotational invariance of this metric, we applied the method of KEARSLEY (1989) which finds the optimal orthogonal transformation to superimpose two point sets based on quaternions. This is done by applying the transformation on the well known Dynamic Programming version of Eq. (10), where partial best matches (sub-sequences) are stored in a table. The result is a metric called QRLCS (quaternion-based rotationally invariant LCS).

3.2 Prediction

The proposed motion prediction method utilizes as probabilistic tracking framework. Given a history of object states, i.e. a trajectory $X_{1:t}$ up to a current time t, we intend to predict the object state φ_T at a specific point in time T in the future. The uncertainty of this prediction can be formulated as a distribution $p(\varphi_T \mid X_{1:t})$, which is rewritten as $p(\varphi_T \mid X_{1:t}) = p(\varphi_T \mid \Psi_t)\, p(\Psi_t \mid X_{1:t})$, where we have incorporated the current object state Ψ_t. In the context of trajectories, Ψ_t represents a sequence of trajectory points (a sub-trajectory) including the position at the current time t and its history over a given travelled distance d_{tr}. We choose the distance window instead of a time window because the characteristics of vehicle motion are represented by the travelled distance, while the motion history especially of objects which are standing or moving slowly may be less meaningful when integrated over a uniform time interval.

The distribution $p(\varphi_T \mid \Psi_t)$ is the likelihood that the predicted state φ_T occurs when the sub-trajectory Ψ_t is given. Since we are using motion samples as reference, where each sub-trajectory has a deterministic extrapolation, the value of this likelihood can be set as constant and thus be neglected. Applying Bayes rule to the remaining distribution $p(\Psi_t \mid X_{1:t})$ results in an estimation of the current state based on the current measurement and the previous states as follows (η is a normalization constant):

$$p(\varphi_T \mid X_{1:t}) = \eta\, p(X_{1:t} \mid \Psi_t) \int p(\Psi_t \mid \Psi_{t-1})\, p(\Psi_{t-1} \mid X_{1:t-1})\, d\Psi_{t-1} \tag{11}$$

This distribution is represented by a set of S samples or particles $\{\Psi_t^{(s)}\}_S$, which are propagated in time using a particle filter (BLACK et al. 1998). Therefore, each particle $\Psi_t^{(s)}$ represents a sub-trajectory of the current state. The distribution $p(X_{1:t} \mid \Psi_t)$ represents the likelihood that the measurement trajectory $X_{1:t}$ can be observed when the model trajectory is given; it can be obtained by the QRLCS metric. According to (SIDENBLADH et al. 2002), it is sufficient to sample the particles from the distribution $p(\Psi_t \mid \Psi_{t-1})$ from a motion database as follows, resulting in an efficient implicit probabilistic motion model.

In a first step, the trajectory database is constructed by creating samples with overlapping windows of equal travelled distances d_{tr}. Since this procedure creates sub-trajectories with different numbers of points, we applied the Chebyshev decomposition to the velocity and yaw angle components of the trajectories to obtain a vector of Chebyshev coefficients $[c_v, c_a]$ for the velocity and the yaw angle, respectively. Then, a dimensionality reduction is performed using principal component analysis (PCA). The particles are also transformed to this low-dimensional coefficient space. The database of samples is then converted into a binary tree using the previously determined coefficients. The top node in the tree corresponds to the coefficient that captures the dimension of largest variance in the database, where lower levels capture the finer motion structure. At each level l, a sub-trajectory i is assigned to the left sub-tree when its coefficient $c_{i,l} < 0$ and assigned to the right one if $c_{i,l} \geq 0$. Each of the leaf nodes contains an index into the motion database.

SIDENBLADH et al. (2002) argue that sampling particles from the state transition distribution $p(\Psi_t \mid \Psi_{t-1})$ can be approximated by a probabilistic search in the database. When a particle reaches a leaf, the prediction step is performed by shifting the particle (i.e. the sub-trajectory) with the appropriate time over the trajectory to which the leaf points. The probabilistic search depends on the particle represented by its PCA-transformed Chebyshev coefficients c_i. At each level l in the binary tree it is decided with the probability

$$p_{right} = p(c_{t,l} \geq 0 \mid c_{i,l}) = \frac{1}{\sqrt{2\pi\beta}\sigma_l} \int_{z=-\infty}^{c_{i,l}} e^{-\frac{z^2}{2\beta\sigma_l^2}} \, dz \tag{12}$$

whether the particle is moved to the right subtree, otherwise the left one is chosen. The value β is a temperature parameter describing the spreading deviation around each particle c_i. The higher the value of β, the more likely the new regions of interest are explored. The variances σ_l^2 are normalisation factors and correspond to the eigenvalues of the covariance matrix computed for determining the PCA of the Chebyshev coefficients.

Since the distribution of the predicted states $p(\varphi_T \mid X_{1:t})$ is approximated by means of the particle filter, the estimated states σ_l^2 can be obtained by looking ahead for a specific time interval ΔT from the current object states $\Psi_t^{(s)}$ on the associated trajectories. This results

in many hypotheses which often lie closely together. To condense this set into a small number of hypotheses, we apply a mean shift method (COMANICIU et al. 2002). The key idea is to estimate local densities of the predicted states $\varphi_T^{(s)}$ by constructing a kernel over each state and then to shift the states iteratively towards higher densities.

Fig. 2: Estimation results of an oncoming vehicle first detected at 48 m distance. The predicted driving path indicating curve radius and velocity for the next second and the tracked 3D points are shown as well as a bounding box indicating the object pose.

4 Experimental Results

For testing, a database of reference trajectories has been set up based on 110 training sequences, including different turn manoeuvres and straight movements, using the approach proposed in Section 2. These observations have been extracted on real world data using the following scenario. A vehicle moves straight toward the stationary ego vehicle ($v = 7.5\, m/s$ on average) and then turns right (left from the perspective of the ego vehicle) at 20 m distance (Fig. 2). This database is further used as a motion knowledge base in the particle filter system as stated in Section 3. One trajectory is left out for testing and is thought as *prediction ground truth* in this experiment, because the whole trajectory, i.e. previous and future movement, is known.

For comparison with the proposed state prediction method, we simultaneously apply a standard extrapolation technique assuming constant acceleration and curve radius with respect to the current vehicle state. In the particle filter system we use $S = 200$ particles, 50 Chebyshev coefficients in each dimension, a tree depth of 12, a temperature parameter $\beta = 0.8$, a travelled distance of $d_{tr} = 25\, m$, and a mean-shift kernel width of $h = 4.0\, m$. In Fig. 3(b) an example is shown for the prediction at a current time step, where approximately four metres before the turning manoeuvre, the two kinds hypotheses “right turn” and “straight” turn up to be most significant. The turning hypothesis is correctly chosen as the predicted object's state. Since the particle filter is a probabilistic approach, each run on the same trajectory will give slightly different results. To examine if the standard deviation of the prediction result remains small, the particle filter system was run ten times for a prediction interval of 2 s on the test trajectory (cf. Fig. 3(a)). The resulting average prediction

error and its standard deviation for each time step are shown in Fig. 4 along with the prediction error of the standard model. Besides the position error, we also state the errors for the velocity, yaw angle, and yaw rate, as they represent important characteristics of the predicted state. The prediction behaviour of the particle filter approach is clearly superior compared to the standard model, especially during the turning manoeuvre.

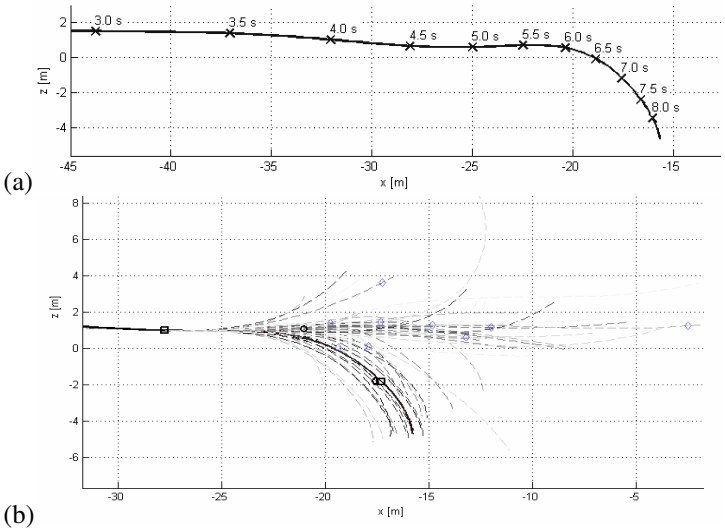

Fig. 3: (a) Test trajectory for different time steps. The camera is looking from the right side at the scene. (b) Particle filter prediction for a prediction interval of 2 s. The strong line depicts the movement of a vehicle, whereas the dashed lines represent the motion hypotheses. On the left side, the black square depicts the current vehicle position; the circle denotes the predicted state by the standard method and the diamond sign the predicted state by the particle filter system. Nearby, the black square shows the ground truth's state.

5 References

Badino H. (2004): *A robust approach for ego-motion estimation using a mobile stereo platform,* in 1st Intern. Workshop on Complex Motion (IWCM04), Günzburg, Germany

Bar-Shalom, Y., Rong Li, X. & Kirubarajan, T. (2001): *Estimation with Applications To Tracking and Navigation*. John Wiley & Sons, Inc

Barth, A. & Franke, U. (2008): *Where Will the Oncoming Vehicle be the Next Second?*, in Proc. of IEEE Intelligent Vehicles Symposium, pp. 1068-1073

Black, M. J. & Jepson, A. D. (1998): *A Probabilistic Framework for Matching Temporal Trajectories: CONDENSATION-Based Recognition of Gestures and Expressions,* ECCV, Springer, pp. 909-924

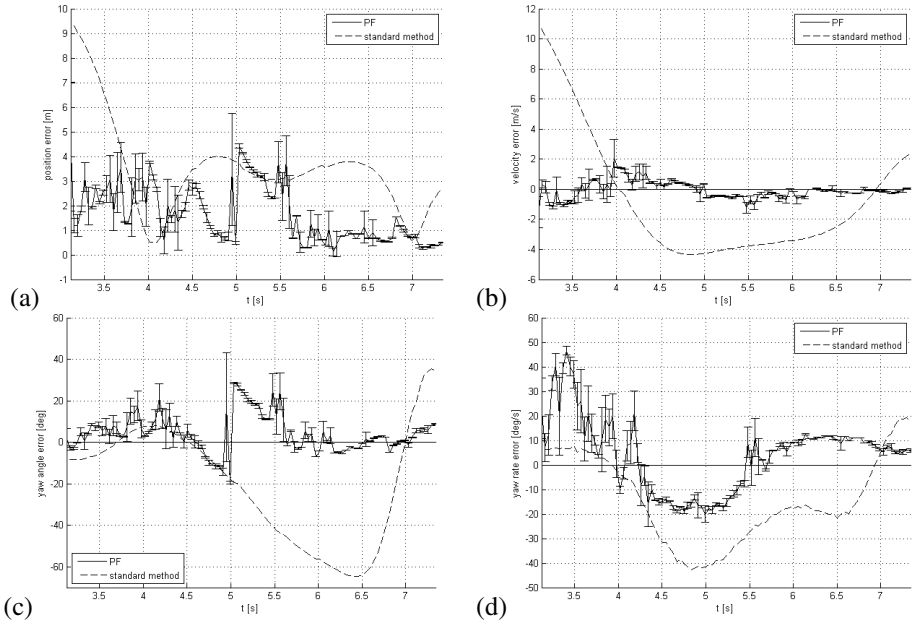

Fig. 4: Errors and standard deviations over time for a single test trajectory (cf. Fig. 3(a), prediction interval is 2 s, ten runs of the particle filter. The turning manoeuvre begins at $t = 6\,s$. (a) Position error, (b) velocity error, (c) yaw angle error, (d) yaw rate error.

Comaniciu, D. & Meer, P. (2002): *Mean shift: a robust approach toward feature space analysis,* IEEE Transactions on PAMI, vol. 24, pp. 603-619

Franke, U., Rabe C., Badino, H. & Gehrig S. (2005): *6D-vision: Fusion of stereo and motion for robust environment perception*, in 27th DAGM Symposium, pp. 216-223

Kearsley, S. K. (1989): *On the orthogonal transformation used for structural comparisons*, Acta Cryst., A45, pp. 208-210

Sidenbladh, H., Black, M. J. & Sigal, L. (2002): *Implicit Probabilistic Models of Human Motion for Synthesis and Tracking,* ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I, Springer-Verlag, pp. 784-800

Tomasi, C. & Kanade T. (1991): *Detection and tracking of point features*, Carnegie Mellon University, Tech. Rep. CMU-CS-91-132

Vlachos, M., Kollios, G. & Gunopulos, D. (2005): *Elastic Translation Invariant Matching of Trajectories,* Mach. Learn., Kluwer Academic Publishers, vol. 58, pp. 301-334