2008 IEEE Intelligent Vehicles Symposium
Eindhoven University of Technology
Eindhoven, The Netherlands, June 4-6, 2008

The Unscented Kalman Filter For Pedestrian Tracking From a Moving Host

Mirko Meuter*, Uri Iurgel†, Su-Birm Park† and Anton Kummert*

* Faculty of Electrical Engineering and Media Technologies
University of Wuppertal
D-42119 Wuppertal, Germany
{Meuter, Kummert}@uni-wuppertal.de

† Delphi Electronics & Safety
Advanced Engineering
D-42119 Wuppertal, Germany
{Uri.Iurgel, Su-Birm.Park}@delphi.com

Index Terms— **Pedestrian Tracking, Active Safety, Image Processing, Moving Host**

***Abstract*— In this paper we present a time-efficient estimation framework for camera-based pedestrian tracking from a moving host car using a monocular camera. An image processing system processes the camera output to find the location of objects of interest in each frame. The position and sensor information about the host translation and rotation are passed to a tracking module. The module uses the position of the detected object's foot point as measurement input and connects them over time to estimate the movement of the objects of interest in order to reduce noise and single frame failures in the detection process. We have developed a new method to estimate the target movement which takes into account the host movement and allows to exploit prior information about the intrinsic and extrinsic camera parameters. The basic idea is to assume that host and target movements can be modelled as 2-dimensional movements on a flat ground-plane. Our developed motion model is based on this assumption and includes host motion as well as the target ego motion. A measurement is modelled as a perspective projection of a point on the ground-plane to the image plane. The motion and the measurement model are combined by an Unscented Kalman filter. This filter is relatively new and has not been applied for pedestrian tracking before. Finally, we present a new logical initialization strategy for the selected filter, a part that is left out by most other publications. First results indicate that our approach gives good tracking results and allows to track pedestrians from a moving host in real time.**

I. Introduction

In the last years, there has been much research for the development of computer vision based driving assistance systems. This kind of system shall detect potentially dangerous situations and warn the driver. One application for such a system is the detection of vulnerable road users like pedestrians in order to avoid lethal accidents. Object tracking is an important topic for such an application. It allows to extract the movement information from objects of interest to bridge missing detections and to reduce the noise that single frame detections usually contain. While there is much literature about tracking of multiple objects using a static camera, there are only few publications about tracking of pedestrians from a moving host. Philomin et al. [1] use a particle filter for optical contour tracking. They avoid modelling the host and target movement and compensate it by using high process noise combined with a high number of particles, which is computational very expensive. Binelli et al. [2] use an infrared camera for the detection of pedestrians. They implement a standard Kalman filter and a simple $\alpha\beta\gamma$-filter for position based tracking. For simplicity, the influence of the rotation and translation from the host is handled separately. The rotation is converted to a fixed translation vector that is added to all target positions despite of their distance. Then, the host translation is incorporated by mapping the detection to 3D space by an analysis of the foot-point of the bounding box. The points are moved according to the travelled distance and finally mapped back to image coordinates. Gavrila et al. [3] use symmetry for detection. Then they generate bounding boxes based on the detection result and refine the bounding boxes using stereo camera information. The resulting bounding box size and position is fed into an Extended Kalman filter to estimate the position in world coordinates. In the paper the projection functions are linearised and under the assumption that the camera angle is very small, camera angle dependent terms are discarded. They present no movement model in their paper.

Our system uses a black and white mono camera. We generate position measurements of the foot-point of a pedestrian or a pedestrian like object for each frame using image processing methods. As contribution to the existing literature, we present a complete estimation method for position based pedestrian tracking for one target from a moving host. We assume that the target as well as the host move on a flat ground-plane. Using this assumption, we can combine the movement of the target and the knowledge about the host movement in the estimation process. The final movement equations are reformulated as a simple Kalman filter prediction step. The measurements are modelled as the result of a perspective projection of points in the virtual ground plane to the image plane perturbed by measurement noise, which allows to exploit knowledge about the intrinsic and extrinsic camera calibration parameters in the projection step. In order to update the state estimates by image plane measurements, we apply a new filter that has recently shown up in literature: The Unscented Kalman filter [4], [5]. From our knowledge,

978-1-4244-2569-3/08/$20.00 ©2008 IEEE.

this filter has not been applied for pedestrian movement estimation before. Finally, we present a new logical initialization strategy based on the Unscented Transformation, a part that is left out in most other papers. With our approach, we can track pedestrian movements in real time and first examples show very promising results.

II. The Pedestrian Recognition System

We will give a short overview over the processing chain of our pedestrian recognition system as the system is not in the scope of this paper. Pedestrian candidate detections are generated by searching for strong vertical edges in a region of interest using the Inverse Perspective Matching profile [6]. The vertical bottom point of the object is found by conducting a footpoint search on the image. The horizontal position is refined by searching for a horizontal symmetry peak. The points are clustered and points which are very close together are merged. The resulting points are fed as measurement input into a multi-target tracking system as well as velocity and yaw rate data from the CAN bus. This data is used to calculate the host path which is later used by the filter described in this paper. The measurement to track assignment is done using gating followed by a Nearest Neighbor selection algorithm [7]. The filtered object positions are used to generate bounding boxes. The boxes are used to extract local features for a Neural Network classifier which decides whether the candidate is truly a pedestrian or not. The temporal stability is further improved by a track based decision fusion voting scheme [8]. The system is depicted in figure 1.

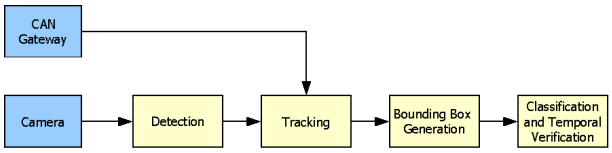


Fig. 1. Pedestrian Recognition System Overview

III. Filtering Pedestrian Movements Using The Unscented Kalman Filter

The Unscented Kalman filter is an extension of the normal Kalman filter for systems with nonlinear process or measurement equations. Literature states [4] that the filter shows a superior performance compared to an extended Kalman filter that is based on the linearization of the nonlinear equations, as it more accurately captures the mean and covariance. The filter is based on the Scaled Unscented Transformation which approximates the statistics of a random variable that undergoes a nonlinear transformation using the following technique. From a random variable with known mean and covariance that is approximately normally distributed, a small set of points and corresponding weights is generated. The idea behind the Unscented Transformation is that this small set of points is enough to represent the distribution. These so called sigma points are propagated through the nonlinear function to generate a new sigma point set. From this set, the resulting statistics like mean and covariance are calculated. We will briefly explain the Scaled Unscented Transformation and the necessary mathematical steps to give a better understanding how the technique works. Let the mean and the covariance of a random variable x with dimension n_x be given by $\bar{x}$ and P_x. From these statistics, a set of $2n_x + 1$ sigma points χ^i is generated by

$$\chi^i = \bar{x} \qquad i = 0 \tag{1}$$

$$\chi^i = \bar{x} - (\sqrt{(n_x + \lambda)P_x})(i) \qquad i = 1, \ldots, n_x \tag{2}$$

$$\chi^i = \bar{x} + (\sqrt{(n_x + \lambda)P_x})(i) \qquad i = n_x + 1, \ldots, 2n_x. \tag{3}$$

In these equations λ is a scaling factor which is defined as $\lambda = \alpha^2(n_x + \kappa) - n_x$. The parameter α determines the spread of the sigma points around the mean $\bar{x}$ and is set to a small positive value $1 \leq \alpha \leq 1e-4$. κ should be selected as $\kappa \geq 0$ to ensure positive definiteness of the covariance matrix. $\sqrt{(n_x + \lambda)P_x})(i)$ is the i-th column of the matrix square root of $(n_x + \lambda)P_x$. Because the covariance matrix is positive semi-definite, it is possible to apply a Cholesky factorization to calculate the matrix square root. The specific value of κ is not critical hence a good default choice is 0. A weight is assigned to each sigma point according to

$$w_m^0 = \lambda/(n_x + \lambda) \tag{4}$$

$$w_c^0 = \lambda/(n_x + \lambda) + (1 - \alpha^2 + \beta) \tag{5}$$

$$w_m^i = w_c^i = 1/\{2(n_x + \lambda)\} \qquad \forall i > 0. \tag{6}$$

β is used to incorporate additional knowledge about the prior distribution. For Gaussian distributions, $\beta = 2$ is optimal [5]. The χ^i sigma points are propagated through the nonlinear function f_t. The resulting sigma points

$$\gamma^i = f_t(\chi^i) \tag{7}$$

and their corresponding weights are used to approximate the resulting values for the expectation and the covariance

$$\bar{y} = \sum_{i=0}^{2l} w_m^i \gamma^i \tag{8}$$

$$P_{yy} = \sum_{i=0}^{2l} w_c^i [\gamma^i - \bar{y}][\gamma^i - \bar{y}]'. \tag{9}$$

The process is shown in figure 2 with the sigma points

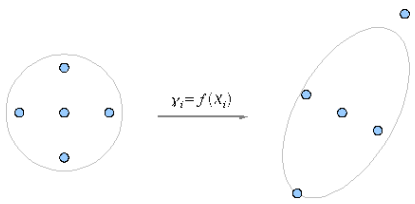


Fig. 2. Sigma Points

shown in blue and the covariance visualised using gray ellipsoids.

A. Movement Model

In this section we describe the applied motion model, which contains a pedestrian motion model as well as an ego-motion compensation. Our model is based on the assumption that the host and the target both move on the same virtual ground-plane. Most time, the road in front of the car is flat so that the model gives a good approximation to the real world situation. For an object of interest, we assume that it moves with nearly constant velocity, which is a reasonable model for pedestrians [9]. The state vector for this model is given by $x^t = \begin{bmatrix} p_x v_x p_y v_y \end{bmatrix}$ with p_x and p_y being the position in x and y direction and v_x and v_y being the speed in x and y direction in the ground-plane. We make a new extension to this model to take into account the host movement. Internal sensors measure the yaw rate and velocity of the host. This data is used to calculate the translation followed by a rotation of the host on the ground-plane. The sensor data is assumed to be relatively accurate, thus we neglect the failures of the velocity and yaw rate sensors in the estimation process. We define the host to be located in the centre of the plane coordinate system and that the host is always aligned to the axis of the coordinate system. Using this definition, a rotation of the host maps into state space as rotation of the coordinate system into the opposite direction. Let ϕ define the rotation angle expressing how much the direction of the car has changed between the current and the last frame. A rotation of the coordinate system means that position and velocity components of the state vector have to be rotated into the opposite direction by the following matrix.

$$R_h = \begin{bmatrix} cos(-\phi) & 0 & -sin(-\phi) & 0 \\ 0 & cos(-\phi) & 0 & -sin(-\phi) \\ sin(-\phi) & 0 & cos(-\phi) & 0 \\ 0 & sin(-\phi) & 0 & cos(-\phi) \end{bmatrix} \tag{10}$$

The translation of the coordinate system caused by the host movement that shifts the objects relatively to the rotated coordinate system is given by

$$t_h = \begin{bmatrix} -cos(-\phi)t_x + sin(-\phi)t_y \\ 0 \\ -sin(-\phi)t_x - cos(-\phi)t_y \\ 0 \end{bmatrix}. \tag{11}$$

where t_x and t_y denote the translation of the host related to the original unrotated coordinate system. For the motion of the targets, we apply a constant velocity ego motion model by multiplying the state vector with the matrix [9]

$$T_t = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{12}$$

In this matrix dt is the time that has passed between the previous and the current frame. Finally, each target must be shifted according to the host translation. The complete process is shown in figure 3. Sudden unexpected movement changes are modelled as additive white acceleration noise

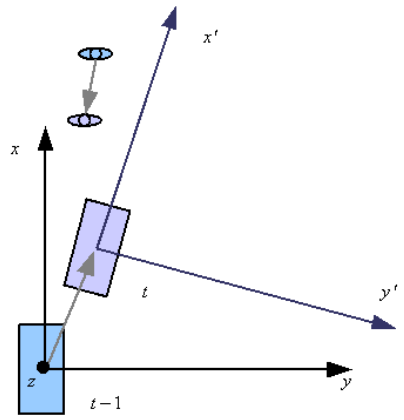


Fig. 3. Movement Model

that is uncorrelated in x and y direction. Since we assume the position uncertainty from the host sensors to be neglectable, the process noise matrix is given as [9]

$$Q = \begin{bmatrix} \frac{a_x^2 t^3}{3} & \frac{a_x^2 t^2}{2} & 0 & 0 \\ \frac{a_x^2 t^2}{2} & a_x^2 t & 0 & 0 \\ 0 & 0 & \frac{a_y^2 t^3}{3} & \frac{a_y^2 t^2}{2} \\ 0 & 0 & \frac{a_y^2 t^2}{2} & a_y^2 t \end{bmatrix}. \tag{13}$$

In matrix Q, a_x and a_y is the spectral amplitude of the noise in x and y direction. We assemble the steps to a standard Kalman filter prediction equation by calculating the system matrix as

$$A = T_t R_h. \tag{14}$$

Finally, we use the following equations to predict the state vector and the corresponding covariance matrix:

$$x_{t|t-1} = A x_{t-1|t-1} + t_h \tag{15}$$

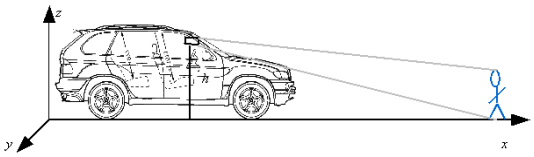
$$P_{t|t-1} = A P_{t-1|t-1} A^t + Q. \tag{16}$$

B. Measurement Model

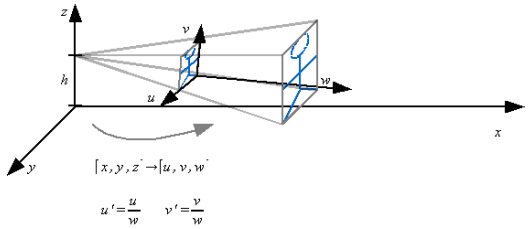
In this section, we model the connection between the measurements and the estimate. The basic idea is to describe this relation as a perspective projection of points from the virtual ground-plane to the image-plane using a pinhole camera model 4(a). We assume the data about the intrinsic and extrinsic camera calibration as known and fixed. The data is used to assemble a precomputed camera calibration matrix C. This step is assumed to be known, and we refer to [10] for details. Let $w_c^t = \begin{bmatrix} x & y & 0 & 1 \end{bmatrix}$ denote a point in world coordinates in the ground-plane and w_h^t the corresponding vector in homogenous coordinates. In order to calculate the expected point in the image plane, first a vector $i_h^t = \begin{bmatrix} u, v, w \end{bmatrix}$ is generated using

$$i_h = C w_h. \tag{17}$$

From this vector, the final image plane coordinates can be obtained by dividing both u and v by w 4(b). In the upcoming part, we will denote f as the function performing



(a) Ideal Road Model



(b) Coordinate Transformation

Fig. 4. Measurement Model

the mapping of points on the ground plane to the image plane. Detection failures are modelled as additive Gaussian noise in the image plane given by the following noise covariance matrix [9]

$$R = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}. \tag{18}$$

Since the measurement projection equations are nonlinear, we use the Unscented Kalman filter for the final update of the estimate. We use the predicted state $x_{t|t-1}$ and the predicted covariance matrix to generate a sigma point and a corresponding weight set according to eq. 1 and 4. After discarding the unobservable velocity state components the sigma point set is passed through function f:

$$\gamma_{t|t-1}^i = f(\chi_{t-1}^i). \tag{19}$$

The resulting set is used to create the predicted measurement and the predicted measurement covariance

$$y_{t|t-1} = \sum_{i=0}^{2l} w^i \gamma_{t|t-1}^i \tag{20}$$

$$P_{yy} = \sum_{i=0}^{2l} w^i [\gamma_{t|t-1}^i - \hat{y}_{t|t-1}][\gamma_{t|t-1}^i - \hat{y}_{t|t-1}]' + R. \tag{21}$$

After this step the state-measurement cross-correlation matrix is calculated and used to obtain the Kalman Gain by

$$P_{xy} = \sum_{i=0}^{2l} w^i [\chi_{t|t-1}^i - \hat{x}_{t|t-1}][\gamma_{t|t-1}^i - \hat{y}_{t|t-1}]' \tag{22}$$

$$K_t = P_{xy} P_{yy}^{-1}. \tag{23}$$

The Kalman Gain maps the difference between the predicted measurement and the actual measurement to the estimates. The final estimate including the newest observation and the updated state covariance matrix is calculated by

$$v_t = y_t - \hat{y}_{t|t-1} \tag{24}$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t v_t \tag{25}$$

$$P_{t|t} = P_{t|t-1} - K_t P_{yy} K_t'. \tag{26}$$

C. Initialization

In order to start the estimation process, it is necessary to initialize the filter. This requires to initialize the state vector as well as the initial covariance matrix. For the initialization, we use the same model we have used in the movement and the measurement process. The measurement is normally distributed with a mean given by the position of the measurement and a covariance matrix given by eq. 18. We use the Unscented Transformation to transform these image-plane statistics to the ground-plane. From the position vector and the covariance matrix, we generate a set of sigma points according to eq. 1. This sigma point set is projected to the ground-plane. This is done by calculating the sigma point position in world coordinates and creating a straight line between the sigma point and the camera focal point. The intersection point of this line with the ground-plane is the resulting sigma point position. From the resulting point set, the mean position and the position covariance matrix is generated according to eq. 8. The mean position is used to initialize the position terms in the state vector, the position covariance is used to initialize the position covariance terms in $P_{1|1}$. We set the velocity terms in the state vector to 0 and want to use the filter to estimate the velocity in subsequent steps. The velocity covariance is modelled as uncorrelated in x and y direction and the corresponding sigma of the velocity is set to typical pedestrian running speeds. Initially, the velocity and measurement errors are uncorrelated, thus the corresponding terms in $P_{1|1}$ are also set to 0.

IV. Results

We have executed the proposed algorithm on a 3 GHz Pentium IV PC with 2 Gigabytes of RAM. For the algorithm, we use a precomputed camera calibration matrix and have measured the average run-time required by each algorithm step shown in table I. The result shows that the algorithm

Algorithm Step	Average run time
state and measurement prediction	0,010 ms
state update	0,005 ms
initialization	0,006 ms

TABLE I

can be considered very fast and is capable of handling multiple targets in real-time even on weaker hardware. The filter was implemented in the system described in section II. Figure 5 depicts the performance of the tracker in various scenes. The images on the left shows all tracks in the scene. As explained before, the tracker follows not only pedestrians but all pedestrian candidates. The black dotted line shows the estimated movement trajectory for the last 15 frames. For each frame, the trajectory is connected to a

Fig. 5. The Tracker on Various Scenes

cross that shows the location of the raw measurement used for estimation. The circle around the newest point shows the gating region in which the association for the current time step took place. The figures on the right hand show the corresponding top view of the scenes from the left. The dots mark the estimated position of each target together with the corresponding uncertainty shown as elliptical region. The figures 5(a) and 5(c) show simple artificial scenes. In the

(a) (b)

(c) (d)

Fig. 6. Final System Output

first image, a pedestrian is crossing the street while the car is standing. The pedestrian is tracked correctly and the filter significantly reduces the detection noise. The second scene shows a pedestrian on the sidewalk, while the car is driving. Again the target is correctly tracked. Note that our tracker does not lag behind if the target gets close, a common problem with a constant velocity image plane tracker. The images 5(e) and 5(g) show real scenes in a city containing many pedestrians, the car is driving in both cases. Scene 5(g) is especially difficult because the host car makes a rolling turn to the right. It can be seen, that in both cases the tracking performance is quite good and the trajectories are correctly extracted, which also improves the performance of the next processing steps. In figure 5(h) all objects are correctly remapped. Due to the ego motion compensation, the target velocities of the background objects is correctly estimated as almost zero. The performance of the final system in the scenes after classification is shown in figure 6. The candidates which were finally identified by the system as pedestrians are shown in yellow boxes.

V. Conclusion

We have presented a new filter algorithm for visual tracking of pedestrians from a moving host. We presented a way to consider the necessary ego-motion compensation in the prediction equations. For the update step, we have used an Unscented Kalman filter to model the nonlinear projection of points from a virtual ground-plane to the image-plane. We have also presented a new initialisation strategy for the proposed filter based on the same modelling assumptions used for the filtering process. The measured computational time shows that the filter is capable of tracking multiple targets in real-time. In the future, we will try to add additional constraints like a maximum speed in the tracker to improve the reliability and we will try to improve the target to measurement association and try to create a real-time system which is capable to maintain multiple hypothesis under association uncertainty especially for dense multi-target scenarios.

References

[1] V. Philomin, R. Duraiswami, and L. Davis, "Pedestrian tracking from a moving vehicle." IEEE Intelligent Vehicles Symposium, 2000.

[2] E. Binelli, A. Broggi, T. Graf, and M. Meinecke, "A modular tracking system for far infrared pedestrian recognition," 2005.

[3] D. Gavrila, J. Giebel, and S. Munder, "Vision-based pedestrian detection: The protector system." IEEE Intelligent Vehicles Symposium, Parma, Italy 2004, 2004.

[4] J. Uhlmann and S. Julier, "A new extension of the kalman filter to nonlinear systems," April 1997.

[5] R. van der Merwe, N. de Freitas, A. Doucet, and E. Wan, "The unscented particle filter," Nov 2001.

[6] G. Ma, S. Park, S. Mueller-Schneiders, A. Ioffe, and A. Kummert, "Vision-based pedestrian detection -reliable pedestrian candidate detection by combining ipm and a 1d profile," in *Intelligent Transportation Systems Conference*, pp. 137–142, IEEE, Sep, 30. 2007-Oct. 3 2007 2007.

[7] Y. B. Shalom and X. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS, 3 ed., 1995.

[8] L. Xu, A. Krzyzak, and C. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition." IEEE Transactions on Systems, Man and Cybernetics, June 1992.

[9] M. Kohler, "Using the kalman filter to track human interactive motion - modelling and initialization of the kalman filter for translational motion," tech. rep., Universität Dortmund, 1997.

[10] V. Lepetit and P. Fua, "Monocular model-based 3d tracking of rigid objects: A survey." Foundations and Trends in Computer Graphics and Vision Vol. 1, No 1 (2005) 189, 2005.