Track-Based Forecasting of Pedestrian Behavior by Polynomial Approximation and Multilayer Perceptrons

Michael Goldhammer, Sebastian Köhler, Konrad Doll and Bernhard Sick

Abstract We present an approach for predicting continuous pedestrian trajectories over a time horizon of 2.5 s by means of polynomial least-squares approximation and multilayer perceptron (MLP) artificial neural networks. The training data are gathered from 1075 real urban traffic scenes with uninstructed pedestrians including starting, stopping, walking and bending in. The polynomial approximation provides an extraction of the principal information of the underlying time series in the form of the polynomial coefficients. It is independent of sensor parameters such as cycle time and robust regarding noise. Approximation and prediction can be performed very efficiently. It only takes 35 µs on an Intel Core i7 CPU. Test results show 28 % lower prediction errors for starting scenes and 32 % for stopping scenes in comparison to applying a constant velocity movement model. Approaches based on MLP without polynomial input or Support Vector Regression (SVR) models as motion predictor are outperformed as well.

M. Goldhammer (✉) · S. Köhler · K. Doll
University of Applied Sciences Aschaffenburg, Würzburger Straße 45,
63743 Aschaffenburg, Germany
e-mail: michael.goldhammer@h-ab.de

S. Köhler
e-mail: sebastian.koehler@h-ab.de

K. Doll
e-mail: konrad.doll@h-ab.de

B. Sick
Intelligent Embedded Systems Lab, University of Kassel,
Wilhelmshöher Allee 73, 34121 Kassel, Germany
e-mail: bsick@uni-kassel.de

Y. Bi et al. (eds.), *Intelligent Systems and Applications*,
Studies in Computational Intelligence 650, DOI 10.1007/978-3-319-33386-1_13

1 Introduction

1.1 Motivation

In the World Health Organisation's last comprehensive status report on road safety, traffic deaths are listed as the eighth leading cause of death with annually more than 1.2 million global cases. Current estimates suggest the possibility that until 2030 traffic accidents will even become the fifth leading cause of death unless urgent action is taken [18]. Furthermore, for every recorded traffic fatality 4 permanent, 8 serious and 50 minor injuries are estimated with costs for the society of more than 100 billion Euro per year, only for Europe [4]. During the last decades large efforts were made to continually improve vehicle safety. Although 27 % of the victims are vulnerable road users (VRUs), e.g., pedestrians and cyclists, VRU protection remained an ongoing problem due to the absence of possibilities for effective passive or active safety mechanisms.

Only recently, vehicles get more often equipped with several types of sensors allowing them to perceive the local surrounding and thus offering advanced comfort and safety functionality to the driver. Common available examples of those "intelligent driving" applications are park distance control, lane assistant, traffic sign recognition or emergency brake assistant. This development offers a unique chance to address the great challenge of VRU safety effectively using early recognition of potentially critical situations to initiate active countermeasures at an early stage. However, mastering this task is complex: in a first processing step VRUs have to be detected and classified. Current state-of-the-art methods of pattern recognition and sensor data as well as image processing have to be performed in real-time generally using embedded hardware units in vehicles. Their computational power increases continuously. The latest developments lead to massive parallel processing by graphics processing units (GPUs) and hardware implementations of computationally complex algorithms using field programmable gate arrays (FPGAs), e.g., [13].

The second major step is understanding the current traffic scene based on information about the own (ego) vehicle, other road users, and the environment (road geometry, obstacles, etc.). The situation has to be continuously analyzed in order to detect critical situations. The criticality is defined by the potential for an accident and thus requires the prediction of the future behavior of potentially involved road users, in our case of the ego vehicle and the VRU. While the ego vehicle's behavior is relatively well known and predictable due to available on-board sensor data and existing models, the VRU has to be considered as an external system. The prediction of the VRU behavior has to be based only on external observations and on prior knowledge of his behavior. While many current approaches use the enhancement of the last movement state for this purpose (e.g., with Kalman filtering (KF) [1]), the VRU behavior is much more complex within the time ranges relevant for these applications. To gain and use the knowledge about VRU behavior for a more realistic movement prediction, models with higher complexity are needed. A new approach of movement prediction by polynomial approximation and supervised learning of multilayer perceptron (MLP) neural networks is in the focus of this publication. The subsequent step is using the gathered information for an adequate reaction of the intelligent vehicle to actively prevent the predicted accident. This may include the driver (information, warning) or it may happen in a completely autonomous way (breaking, evasive maneuver) if the timing constraints, the most important being the time to collision (TTC), fall into the range of human reaction times.

The state of the art in VRU motion modeling and path prediction is dominated by conventional movement models such as constant velocity (CV). As input information the measured position as well as further image and context information may be used. For a more detailed description of the state of the art we refer to own, preliminary work [10].

1.2 Our Contribution

The main contribution of our publication is a novel approach for self-learning trajectory prediction based on polynomial least-squares approximation and multilayer perceptron neural networks. Training and evaluation is done using trajectory data of uninstructed pedestrians in urban traffic scenarios whereby we assume that the method may also be applied to other VRU types. We focus on the prediction itself as major step, not a finished application in a vehicle. The method has the advantage not to be limited to certain types of movements, but it is able to handle all motion types included in training data. As self-learning predictor, it is independent of specific movement models since the network contains all required knowledge implicitly. The least-squares approximation of the discrete input track allows an extraction of the principal information of the current behavior of the pedestrian with the advantages of independence of sensor parameters such as cycle time and improved noise resistance. The output of the proposed method is a continuous position estimate up to a certain time horizon instead of only single trajectory points.

The article is structured as follows: In Sect. 2 the proposed path prediction approach is described. Section 3 outlines the methods used for evaluating the prediction quality while the according test results are set out in Sect. 4. A concluding summary is given in Sect. 5.

2 Methodology

In this section, we describe the usage of the measured trajectory information to predict the behavior of pedestrians represented by their future trajectory.

As the prediction is supposed to be invariant to the current global position and orientation of the pedestrian, the input data of the predictor is based on a time series of the absolute velocity $|v(t)|$ and the angular velocity $\omega(t)$ instead of directly on the global position measurements. The time series are approximated with multiple polynomials in sliding windows in a fixed position relative to the current time t_c in order to extract the polynomial coefficients every time cycle. They are serving as descriptor since they contain the principal information of the observed pedestrian behavior. The prediction output is also supposed to be invariant to the global position and orientation, so we use coefficients of polynomials describing the future velocity profile $v_{lon}(t)$, $v_{lat}(t)$ in the pedestrian's ego coordinate system for this purpose. As output, this representation shows better results than an output based on $|v(t)|$ and $\omega(t)$. The estimated future trajectory is rebuild from the predicted information by numerical integration and a retransformation into the global coordinate system.

The relation between the measured (input) and the future (output) pedestrian movement is established by a multi layer perceptron neural network model, which is capable of predicting all coefficients of the output polynomials based on those of the input polynomials within a single instance. As kernel-based comparison method we also evaluated Support Vector Regression. However, it requires one instance for each output value. The three consecutive steps of pedestrian tracking and data preprocessing, approximation with polynomials and prediction of polynomial coefficients of the future trajectory are described in the following Sects. 2.1–2.3.

2.1 Pedestrian Tracking and Data Preprocessing

The proposed method for pedestrian movement prediction is based on a short time tracking of the horizontal 2D position. The method is independent of the underlying sensor technology as long as it is capable of providing object positions in real world coordinates. Typical sensor setups are stereo cameras, lidar or radar sensors.

In our concrete setup we make use of a wide angle stereo camera system installed at a public urban intersection to generate pedestrian track data needed for training and testing of the self-learning algorithms. The field of view covers two crosswalks with pedestrian lights and two sidewalks of the crossing roads (see Fig. 1, left). The observed scenes provide a large variety of movement types such as straight walking, starting, stopping or bending in. The test site and the sensor setup are described in [9, 11].

The center of the pedestrian's head is serving as reference point for stereo triangulation and tracking. As the human gait can physically be described by the model of an inverted pendulum [17], the upper body and, in particular, the head indicates changes in motion very early (see Fig. 1, right). Furthermore, the center of the head can be recognized and located relatively stable from all directions by computer vision algorithms. This leads to a more robust trajectory measurement and potentially faster detection of upcoming movement changes compared to reference points which are based on the detection of whole persons, e.g., the center point of a detection window. The further major processing steps of the path prediction methodology described below are depicted in Fig. 2.

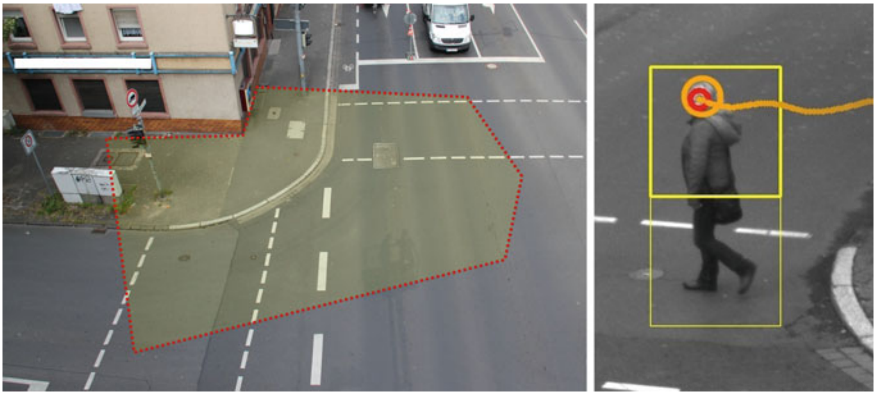

Fig. 1 *Left* field of view for 3D tracking by the wide angle stereo system at a public test intersection used to create the track database. *Right* optical people detection, head detection and -tracking to obtain pedestrian movement data

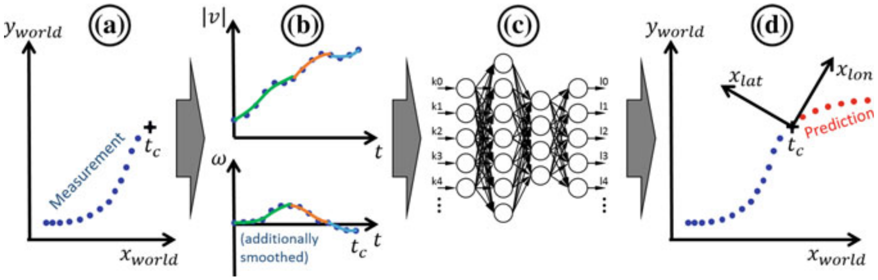

Fig. 2 Overview of the proposed path prediction method

In an own preliminary publication [10] we used the approach of transforming the measured past track into the pedestrian's ego coordinate system, in order to receive an input vector based on $v_{lon}(t)$ and $v_{lat}(t)$ independent of the global position x_{world}, y_{world} and orientation φ_{world}. This transformation has to be reapplied to the track at every time cycle since the pedestrian's position and orientation is continuously changing. In the approach described here, the transformation is substituted by an extraction of the absolute horizontal velocity $|v(t)|$ and the angular velocity $\omega(t)$. Those parameters are invariant to the current values of x_{world}, y_{world} and φ_{world}. Therefore, they have to be calculated only for the current time step while the preceding values remain the same. This sliding window behavior of the extracted time series has great advantages regarding the computational efficiency of the subsequent polynomial fitting, as the fast update algorithms of our *Fast Approximation Library* [7] can be applied. In return, $\omega(t)$ contains more noise compared to the previous approach and, therefore, an additional on-line exponential low-pass IIR filtering of the time series with

$$\omega_{sm,t} = \alpha \cdot \omega_t + (1 - \alpha) \cdot \omega_{sm,t-1} \tag{1}$$

is added with ω_t being the current, $\omega_{sm,t}$ the current filtered and $\omega_{sm,t-1}$ the preceding filtered angular velocity value. The smoothing factor α is included as additional input parameter to the optimization process for the prediction quality.

2.2 Approximation with Polynomials

The time series $|v(t)|$ and $\omega(t)$ are approximated with polynomials. This approximation is based on a least-squares error and orthogonal basis polynomials. The coefficients of the orthogonal expansion of the approximating polynomial serve as principal information sources as they represent the temporal development of the pedestrian's movement. The coefficients of the orthogonal expansion can be regarded as optimal estimators of average, slope, curve, change of curve, etc. of the time series in a considered time window [5, 6]. The approximating polynomial $f(t)$ is a linear combination of the basis polynomials $f_k(t)$

$$f(t) = \sum_{k=0}^{K} w_k \cdot f_k(t) \tag{2}$$

at a finite set of points in time t_0, ..., t_N. The objective is to solve the least squares problem

$$\min_{\mathbf{w}} \|\mathbf{F}\mathbf{w} - \mathbf{s}\|^2 , \tag{3}$$

where $\|\ldots\|$ is the Euclidean norm, **s** are the time series values (targets), **w** is the vector containing the coefficients of the respective polynomials and **F** is a matrix (the design matrix) of form

$$\mathbf{F} = \begin{pmatrix} f_0(t_0) & \ldots & f_K(t_0) \\ \vdots & \ddots & \vdots \\ f_0(t_N) & \ldots & f_K(t_N) \end{pmatrix} . \tag{4}$$

As the approximation has to be performed in a sliding window manner on our approach, the coefficients can be computed with extremely fast update algorithms if certain kinds of orthogonal basis polynomials, in our case discrete Legendre polynomials, are used. The approximation is capable of reducing the dimensionality of a feature vector serving as input for the subsequent self-learning predictor. At the same time the influence of measurement noise is reduced due to implicit data smoothing. A third advantage is that the coefficients can be calculated independent of sensor parameters such as the sampling rate. That is, a predictor could be trained with one and used with another sensor and sampling rate. Even changes of the sampling rate within one time series or handling of missing measurements are conceivable.

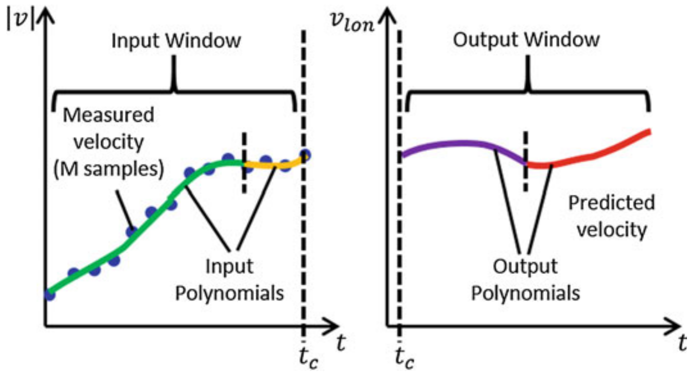

Fig. 3 Schematic representation of polynomial approximation for movement prediction for the case of $|v|$ (*left*, similar for ω, not shown). The coefficients of the fitted input window polynomials are used to estimate those of the v_{lon}, v_{lat} timeseries in the output window (*right*, only v_{lon} is shown). t_c indicates the current time step

In the proposed method we make use of multiple polynomials fitted in different temporal sub-windows together representing the overall input window. The example in Fig. 3 shows two consecutive input polynomials (left: green, orange) approximating the measured velocity (blue dots) where t_c indicates the current time step. The technique allows, e.g., to split the input window into separate sub-windows for short- and long-time observation as shown in this example. Another exemplary variant are multiple small input windows in order to get a closer approximation to periodic velocity variations, which occur within the human gait.

According to the procedure for the input time window the method also uses a polynomial representation for the predicted output information. However, contrary to the input, the 2D velocity in the pedestrian's ego system $v_{lon}(t)$, $v_{lat}(t)$ is taken as output time series as experiments show that even small prediction errors of $\omega(t)$ lead to relatively large errors when transforming back into estimated global positions. As the polynomial approximation for the output window has only to be performed to generate the training output coefficients but not during an on-line processing where the future track is generated from the predicted output coefficients, this does not yield a disadvantage regarding processing time.

For the training of the MLP-based prediction process polynomials are fit in a single or multiple defined output sub-windows. The polynomial coefficients are taken as targets for the MLP models in order to predict them based on the input window coefficients in an on-line mode. The output polynomials can thus be evaluated to obtain a continuous prediction of the future movement for the learned output time window. While overlapping sub-windows are possible as input only non-overlapping consecutive sub-windows are feasible for the output in order to get a unique position estimate for each future point in time. In the visualized example two consecutive output polynomials are predicted (right of t_c: violet, red). The output window in the example is also splitted into two sub-windows, here with constant size.

The number of polynomials, their temporal position and length, as well as the degree of each polynomial can be varied to optimize the prediction quality. The possible upper limit of the variation of polynomial degree K to get an unique approximation solution is $N - 1$ where N is the number of measured velocity values within the considered time window. This limit is only relevant when short windows and low sampling rates are used at the same time.

The coefficients $k = 0, \ldots, K$ are estimators of the average, slope, curve, change of curve, etc., of the polynomial. The underlying reference time unit is the cycle time $T_{cyc} = 1/f_s$ of the time series, with f_s being the sampling rate of the source data. As we want the coefficients to be independent of the sampling rate, they are transformed to a reference time of 1 s by multiplying the scaling factor f_s^k with each coefficient. The transformation results in normalized coefficients with the physical units m/s, m/s^2, m/s^3, etc.

The variation of parameters leads to some special cases at the edges of the parameter space for the input sliding window configuration:

- One single polynomial with $K = 0$: Corresponds to the average (absolute and angular) velocity of the regarded time window. Degree $K = 1$ adds the average acceleration, and so on.
- The case of N polynomials with $K = 0$ corresponds to the direct usage of the time series $|v(t)|$ and $\omega(t)$ as input pattern for the subsequent predictor. This is similar to the technique used for MLP prediction in [8].

2.3 Prediction of Polynomial Coefficients Using a Multilayer Perceptron

As predictor we use a feed-forward artificial neural network (ANN) in form of a multilayer perceptron (MLP, see e.g. [12]). A MLP is capable of predicting multiple output values at the same time what constitutes a great advantage when multiple time steps and dimensions shall be estimated in parallel (Fig. 2c). The network consists of neurons with the sigmoid activation function

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{5}$$

and is trained with the Resilient Backpropagation (RPROP) algorithm [16]. The normalization of the input data is done using the statistical z-transformation

$$z_i = \frac{x_i - \overline{x}}{s} \tag{6}$$

where x_i are the original input values, $\overline{x}$ the mean and s the standard deviation of training values per input dimension and z_i the z-transformed values. The size of the input layer is determined by the number of polynomials N_{Pol} and their degrees K_i with

$$N_{in} = 2 \cdot \sum_{i=0}^{N_{Pol}} (K_i + 1). \tag{7}$$

The size and number of hidden layers are variable and part of the optimization process. The network output consists of polynomial coefficients of the future trajectory represented in the ego velocity time series $v_{lon}(t)$ and $v_{lat}(t)$, which were determined for training by polynomial approximation. The output may also consist of multiple polynomials for separate time windows. To extract the trajectory estimation the output polynomials are evaluated for the required future points in time. From the current time on the predicted $v_{lon}(t)$ and $v_{lat}(t)$ are numerically integrated to obtain the positions $x_{lon}(t)$, $x_{lat}(t)$. Afterwards, the prediction is transformed back into the world coordinate space $x_{world}(t)$, $y_{world}(t)$ (see Fig. 2d).

3 Evaluation of Prediction Quality

To evaluate the quality of the proposed methods we regard a time window of $[-1, 0]$ s for the input and $]0, 2.5]$ s for the output of the predictor, relative to the time stamp of the current measurement. The older the input measurements are, the less influence they have on the result. Also, longer input windows require longer initialization time after the first detection of a pedestrian in a practical application, until the first prediction is available. Tests using MLP prediction based on direct input of the velocity measurements show that input windows larger than 1.5 s do not result in further improvements of the prediction quality on our data. With an input window of 1 s a prediction quality of over 97 % of the measured optimum at 1.5 s is reached. To make the prediction results of different configurations comparable we set the overall input window length to this value. The prediction horizon is set to 2.5 s since this value suffices for autonomous reactions of a vehicle as well as for effective driver warnings in advanced driver assistant system (ADAS) scenarios [15].

As major quality indicator we evaluate the average Euclidean error (*AEE*) from the predicted position to ground truth (GT). As ground truth the tracked head center position is used whereby the tracking in the stereo images and in world coordinates are manually inspected to avoid tracking errors and outliers. The *AEE* is defined for all P predictions in the test data with a specific prediction time step t_{pred} as

$$AEE(t_{pred}) = \frac{1}{P} \sum_{i=1}^{P} \sqrt{(x_{pred}(i) - x_{GT}(i))^2 + (y_{pred}(i) - y_{GT}(i))^2} \tag{8}$$

with $(x_{pred}; y_{pred})$ being the predicted and $(x_{GT}; y_{GT})$ being the ground truth positions. As the *AEE* is based on the Euclidean norm, the results are independent of the underlying coordinate system (here: global or ego coordinates). In order to evaluate the prediction quality for the longitudinal and lateral directions the average position error is also evaluated separately for these dimensions (AE_{lon} and AE_{lat}).

As the minimization of the *AEE* can only provide an optimal overall parameter setting for a specific prediction time step t_{pred} but not for the whole predicted output time window, another indicator is required as sole optimization target value in this case. Since the prediction error naturally rises with increasing t_{pred} an averaging of all *AEE* values would outweigh more recent prediction times. To avoid this problem we decided to consider the *AEEs* with regard to the respective t_{pred} and to calculate the weighted average of the AEEs as target value which has to be minimized:

$$ASAEE = \frac{1}{N} \sum_{i=1}^{N} \frac{AEE(t_{pred}(i))}{t_{pred}(i)}, \tag{9}$$

with N being the number of discrete subsequent prediction steps within the considered prediction horizon and *ASAEE* being the average specific *AEE*.

For our experimental studies we use a database of 1075 pedestrian tracks recorded with the sensor setup described in [9] at a public test intersection. All tracks have a sampling rate of 50 Hz and lengths between 4 and 10 s or 200–500 samples, respectively. They are divided into four types of movement: “Waiting”, “Starting”, “Walking” and “Stopping”. Scenes of type “Waiting” are scenes with persons standing and usually waiting for the light signal to cross the road performing only small movements not exceeding a head velocity of 0.3 m/s. “Walking” consists of straight walking with constant or changing velocities as well as bending-in scenarios. “Starting” and “Stopping” include the corresponding motion transition with one second before and three seconds after. The track database is split into training (60 %, 643 tracks) and test data (40 %, 432 tracks) with equal split ratio for each included scene type (see Table 1). For the optimization of the individual stages of the prediction method the training data is further divided into 70 % training to 30 % test data. Due to the large number of variable parameters a k-fold cross-validation is not performed.

Table 1 Total number of trajectories sorted by training data, test data, and type of movement

Type	Training	Test	Total
Waiting	117	78	195
Starting	184	124	308
Walking	204	137	341
Stopping	138	93	231
All	643	432	1075

4 Results

In this section we present the tested parameter ranges and resulting prediction quality values based on the evaluation methods introduced in Sect. 3. As baseline method for comparison we use CV Kalman Filtering.

As the different phases of the proposed method (input polynomial approximation, MLP, output polynomial prediction) provide a large amount of parameters to minimize the overall prediction error, an extensive grid search in the entire parameter space is not feasible. Instead, the phases are optimized separately in alternating manner. The parameters of the separate stages are optimized with coarse-to-fine grid searches, if possible (MLP network structure), or by manual variation of parameters (the polynomial stages). The varied parameters include

- the exponential smoothing factors of the input time series,
- the number, temporal position, length and degrees of input and output polynomials,
- the number and size of hidden layers of the MLP,
- the parameters C, γ and ε of the alternatively used SVR method with RBF kernels.

4.1 Variation of Polynomial Parameters

In this section we evaluate the effect of changes in size, position, length, and degree of the input and output polynomials. The overall window sizes are thereby held constant at the values defined in Sect. 3: 1 s for the input and 2.5 s for the output window. Figure 4 shows the results of a variation of the polynomial degree K for four sample input configurations.

As base configuration for a comparison of different input configurations, we use $|v|$ and ω data directly (but normalized) as input for the MLP. This corresponds to an input layer size of 100 neurons in our sensor setup with 50 Hz and 1.0 s total input time. The configuration is equivalent to 50 polynomials per dimension with a degree of 0. For the following evaluations of the input structure, the structure of the MLP

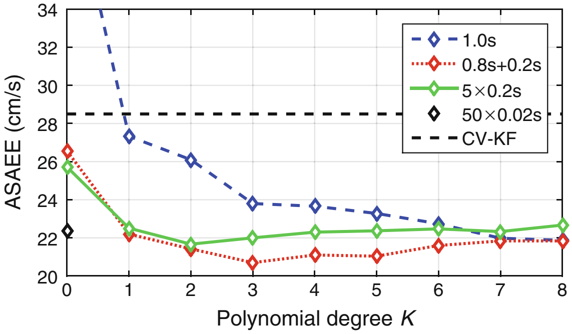

Fig. 4 Quality indicator *ASAEE* depending on input polynomial degree K, input sub-window numbers, positions, and sizes in comparison to the baseline technique of CV Kalman filtering (CV-KF). Lower *ASAEE* values are better. The examples only show an excerpt of the evaluated input configurations

and the output polynomials were kept constant at their optimized final configuration (two hidden layers with 20 and 8 neurons, 5 consecutive output polynomials with degree 2 and 0.5 s length). The given configuration leads to an *AEE* of 20.7 cm for 1.0 s and 71.0 cm for 2.5 s prediction time. The resulting *ASAEE* value is 22.4 cm/s (Fig. 4, single black marker at $K = 0$).

The evaluation for different input polynomial structures generally shows a trend to decreasing prediction errors with increasing polynomial degrees, according to the gain of usable information content for the neural network. The improvement slows down with higher degrees until a saturation plateau is reached, whereby a stronger splitting into more input polynomials leads to an earlier flattening of the *ASAEE* curve. With still higher polynomial degrees a slight increase of the prediction error is observable. In Fig. 4 the blue dashed line shows the configuration with a single polynomial covering the whole input time horizon of 1.0 s. Here, $K = 0$ describes the minimum tested input size using only the average two dimensional velocity. This already leads to an *ASAEE* of 41.8 cm/s. The step from $K = 0$ to $K = 1$ adds the values of mean acceleration to the input, which leads to a benefit of a 35 % lower *ASAEE* (27.3 cm/s) on our test data and already slightly outperforms CV Kalman filtering (28.5 cm/s, see Fig. 4, line “CV-KF”). Using a single polynomial the minimum error plateau is reached at degree $K = 8$ with 21.87 cm/s *ASAEE* according to an input layer size of 18. The green line represents the result for an input of five sequential input polynomials with the same window lengths of 0.2 s. In this case, polynomial degrees of 2 already suffice for the best prediction quality, but due to the higher number of sliding windows the input layer size grows to 30. The red dotted line represents an asymmetrical input configuration with a 0.8 s window followed by a 0.2 s window. The intention is to set an additional attention on short-time features in the second window, while the first covers the longer time distance up to 1.0 s. The configuration also reaches the quality of the previous configurations at a degree of 2 but already with a small input layer size of 12. The best result is reached at degree 3 and layer size 18 with an *ASAEE* of 20.7 cm/s. This input configuration will be used for further quality measurements presented in this work. Other tested input structures lead to similar results, e.g., 2×0.5 s, $0.4 + 0.3 + 0.2 + 0.1$ s, 10×0.1 s and configurations with overlapping sub-windows of 1.0 and 0.2 or 0.1 s. Altogether we can state that it is possible to use only 2 windows without reduction of the prediction quality.

The parameters of output polynomials have less influence on the prediction quality than the input parameters. The *ASAEE* also improves slightly with increasing polynomial degree but with a much earlier entering the plateau at degree 1–2 depending on window number and size. The overall *ASAEE* difference between a single and multiple output sub-windows is only approximately 2 % using a degree of 2. For special cases, e.g., the acceleration phase after an initial movement, the slight improvement is visible in velocity plots (see Fig. 5). For all further evaluations we choose an output of 5 consecutive polynomials of degree 2 and 0.5 s length each.

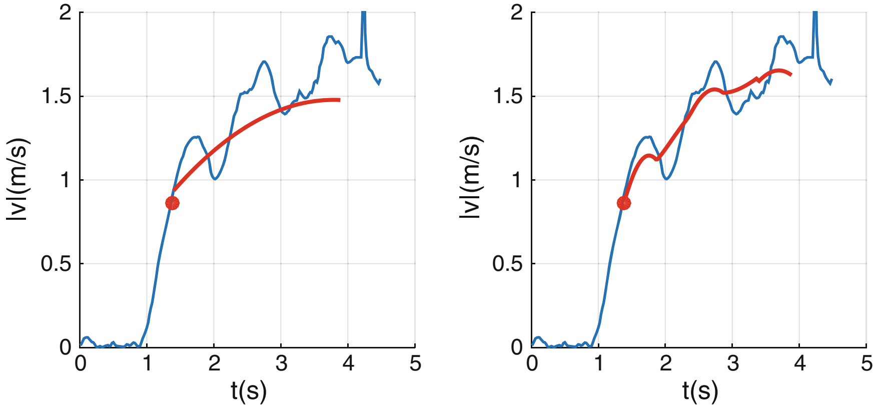

Fig. 5 Comparison of a sample prediction of a starting movement with two different output polynomial configurations in a velocity magnitude plot. The *blue line* represents the actual measurements over time, the *red dot* marks the current point in time, the *red line* represents the prediction. *Left* single polynomial with $K = 2$ and 2.5 s window length. *Right* 5 consecutive polynomials with $K = 2$ and 0.5 s window length each

4.2 Variation of Neural Network Structure

Though the size of the input and output layers is defined by the polynomial structure only size and number of hidden layers can be varied. We performed a coarse-to-fine grid search for network topologies with 0–3 fully connected hidden layers and numbers of 2–40 neurons per layer, including different layer sizes. The results show remarkable improvements for networks with two hidden layers in comparison to those with one or zero while an extension to three hidden layers shows no further advantages. The best result of a grid search with the 0.8 + 0.2 s input configuration and degree 3 is given with 20 neurons in the first and 8 neurons in the second hidden layer (see Fig. 6). Similar network configurations with two hidden layers, e.g., using 4 instead of 8 neurons in the second hidden layer, lead to very similar results, in this case a slight increase of the *ASAEE* of 1.5 %.

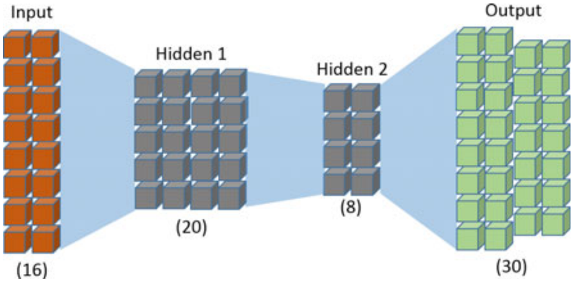

Fig. 6 Architecture of the finally used ANN. The best results were archived with a fully connected MLP with two hidden layers and 16–20-8–30 neurons

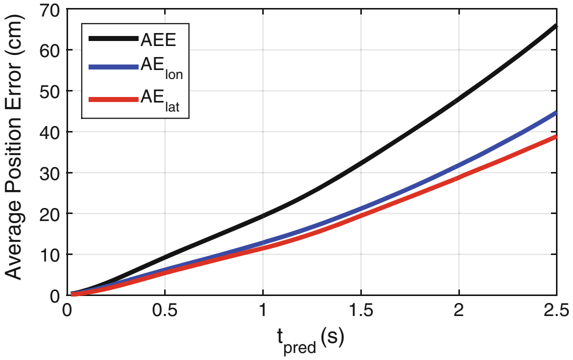

Fig. 7 Average Euclidean (*AEE*), average longitudinal (AE_{lon}) and lateral position errors (AE_{lat}) for the complete prediction horizon up to 2.5 s

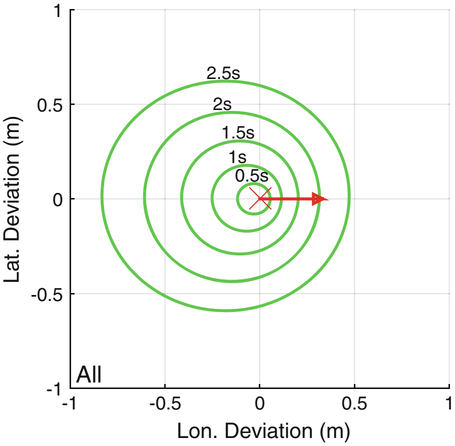

Fig. 8 Error ellipses for five specific prediction times from 0.5 to 2.5 s for all types of movement. The *red arrow* shows the direction of movement, the ellipses indicate the mean and standard deviation of the estimated position relative to the ground truth in the ego coordinate system of the pedestrian

4.3 Quality Depending on Prediction Horizon and Type of Movement

In order to compare the prediction error for different prediction times t_{pred} in this section we evaluate the *AEE* performance for several time steps. Figure 7 visualizes the change of the *AEE* over the regarded prediction time horizon of 2.5 s for all test data predictions. As one can observe the 2D prediction error increases slightly disproportionate with growing prediction time t_{pred}. The longitudinal and lateral components (average absolute errors AE_{lon} and AE_{lat}) show almost equal behavior over time.

The error ellipses for five specific prediction times are drawn in Fig. 8. Each ellipse indicates the mean and standard deviation of the estimated position to the ground truth in ego coordinates of the pedestrians. This overall evaluation shows a slight tendency of predicted velocities that are to low in longitudinal direction.

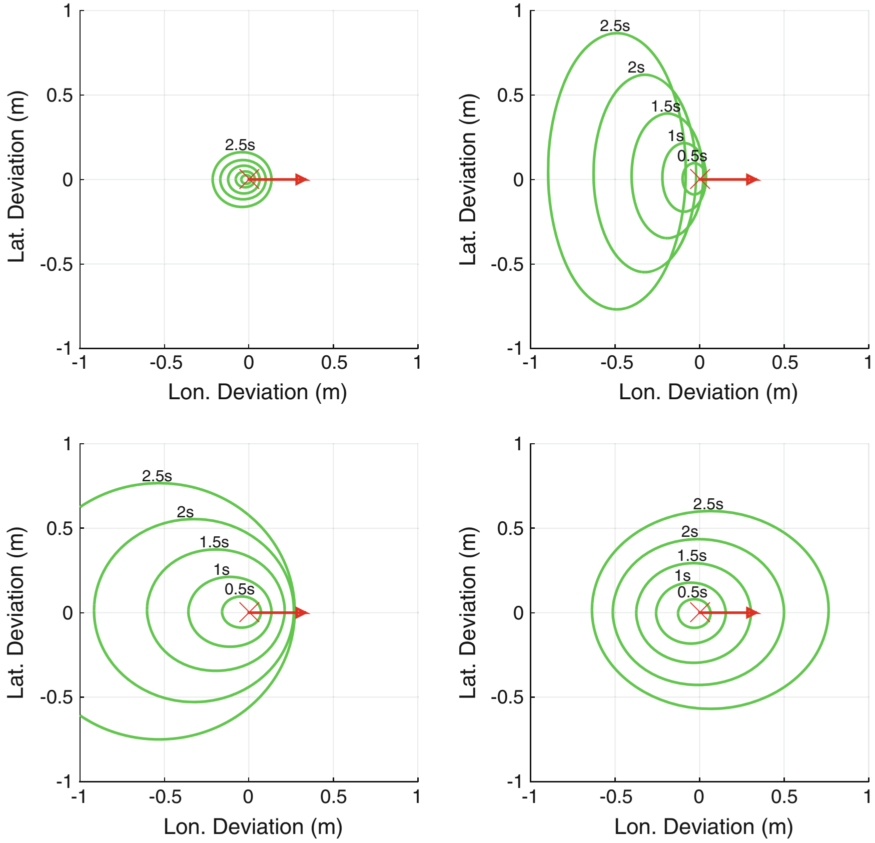

Fig. 9 Error ellipse plots for the four different types of movement labeled in the test data. Waiting (**a**), walking (**b**), starting (**c**), and stopping (**d**)

To investigate this aspect further a similar plot is generated for each of the four labeled movement types separately in Fig. 9. In Fig. 9a the result for people standing and waiting on the sidewalk shows relatively small errors, as expected. The error ellipses appear as concentric circles while the standard deviation even for the maximum prediction time of 2.5 s is just 1 cm. Regarding the evaluation of walking motion in Fig. 9b a shift of the means against the moving direction and a dominant lateral standard deviation are visible. This effect is mainly based on changes of walking direction generating a lateral deviation while the original movement vector in longitudinal direction gets shorter. Figure 9c shows the same plot for starting motions, where even more significant shifts of the mean prediction errors against the moving direction are visible. They arise from the fact that a prediction of the initial movement from a standing position over several seconds is a very difficult task, for machine vision as well as for human observers. The discrepancy between the prediction of an almost constant position and the person starting to move forward in the ground

Table 2 Quantitative prediction error results for the proposed polynomial-MLP method for the four investigated types of pedestrian movement and overall

Mov. type	AEE (1.0 s) (cm)	AEE (2.5 s) (cm)	ASAEE (cm/s)	Comp. to KF (%)
Overall	19.4	85.9	20.7	−27.4
Waiting	5.3	14.6	5.8	−0.8
Starting	26.6	97.6	28.6	−28.0
Walking	21.4	85.9	24.4	−23.1
Stopping	22.3	73.4	23.2	−32.0

In the right columns the percentage difference of *ASAEE* to CV Kalman Filter prediction is listed for comparison

Table 3 Quantitative errors of CV Kalman Filter prediction

Mov. type	AEE (1.0 s) (cm)	AEE (2.5 s) (cm)	ASAEE (cm/s)
Overall	27.9	95.0	28.5
Waiting	6.1	14.6	5.9
Starting	38.0	136.1	39.7
Walking	30.9	99.1	31.7
Stopping	33.2	122.8	34.1

The values are used as baseline for the evaluation of the proposed method in Table 2

truth data leads to this effect. The lateral errors show that estimating the movement direction for standing persons is also a challenging task for the algorithm. Ways to solve this problem could be the consideration of additional input information for prediction, such as the viewing direction of the pedestrian or the orientation of the road she or he intends to cross. In Fig. 9d the opposite effect is visible: For stopping persons the algorithm slightly tends to overestimate the velocity, such that a mean shift in direction of movement occurs. Since the stopping motions of people in traffic scenarios take generally more time than the starting motions (see [9]) the predictor has more time to react on appearing characteristic features.

Detailed numerical results for two prediction times 1.0 and 2.5 s as well as the *ASAEE* are given in Table 2. A comparison to the *ASAEE* values of the KF method taken as baseline (see Table 3) is shown in the right column (reduction of the *ASAEE* in percentage) and as bar plot in Fig. 10. The results show an overall improvement of the prediction result of 27.4 % compared to the Kalman Filter. Considering the individual movement types, the most benefit of our method occurs at starting (28.0 %) and stopping scenes (32.0 %) but also the prediction of walking scenes including bending is improved (23.6 %). The waiting scenarios show almost equal results for the proposed method and the Kalman Filter.

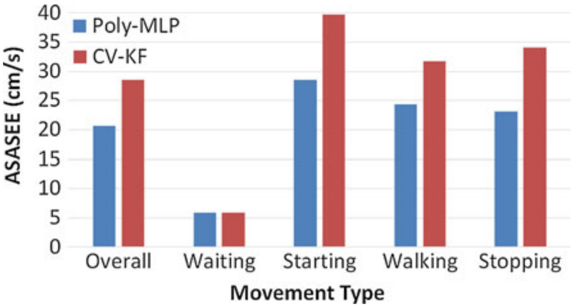

Fig. 10 Visual comparison of ASAEE values for the proposed polynomial-MLP and the baseline Kalman Filter method

4.4 Early Recognition of Initiation of Gait

As shown in the previous section the prediction of a starting motion for a standing pedestrian is a challenging task. At the same time this case is one of the most common and important ones in public traffic scenarios. Therefore we investigate the detection time and precision of the initial movement detection with the proposed method. The test data for this evaluation consists of 40 tracks of non-instructed pedestrians waiting at the roadside and then crossing the road. We define the moment of the heel-off of the first foot moving as reference time $t = 0$ s and labeled the associated time stamp by manually observing the video data. The initial movement is considered as detected successfully when the predicted position for $t_{pred} = 1.0$ s exceeds a specified Euclidean distance from the current position. This corresponds to a threshold for the absolute velocity averaged over the first second of the predicted time span. In order to vary the sensitivity of the motion detection this threshold can be shifted, what which is equivalent to moving the operating point of the detector. A lower velocity threshold accelerates the recognition of the starting motion but increases the probability for false alarms during the standing phase. For our evaluation we define already a single violation of the threshold during the standing phase of a scenario as false alarm (FA), while the correct detection during or after the labeled heel-off without prior false alarms counts as true positive (TP). The precision P over all scenes is defined as $P = TP/(TP + FA)$ and depends on the chosen threshold and, thus, on the detection time relative to heel-off.

The resulting relationship of precision and detection time is plotted in Fig. 11 (red line). The evaluation shows that already at the moment of heel-off a precision of 80 % is reached. This confirms the suitability of the head position tracking as input source for the neural network since 80 % of the initial movements are correctly detected before there is any movement of a foot. During the next 300 ms after heel-off the reachable precision exceeds 95 %. It should be mentioned that the predictor tested here is not optimized for this motion detection application, especially it is not designed to comply with the defined false alarm rate criteria of exceeding a low threshold during the waiting phase. The objective function in the training has been the goal to minimize the 2D position prediction error, not the binary movement state

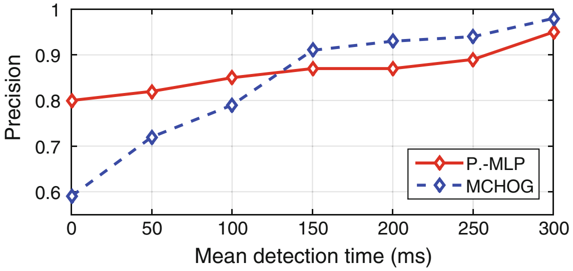

Fig. 11 Precision over mean relative detection time to heel-off for initial movement detection using the proposed prediction method (*red*) and the MCHOG method (*blue*)

(standing/walking). As comparison the same scenes are evaluated using the video-based initial movement detection method presented in [14] (MCHOG, blue dashed line). As one can see, our method shows advantages for the short prediction times up to 100–150 ms after heel-off while the MCHOG shows slightly higher precision rates afterwards.

4.5 Runtime Performance

The computation time of the used trajectory-based algorithms is very short compared to the cycle time of commonly used sensors and algorithms for pedestrian detection. Nevertheless, in this section the runtime performance of the proposed method is evaluated.

Using the multi-polynomial configuration evaluated in the previous section (two input and five output polynomials) the prediction from input to output track requires 35 µs on an Intel Core i7-3770 CPU with 3.4 GHz. So, under the preference of an available pedestrian detection and tracking system, the algorithms should operate on small embedded systems. Compared to the originally proposed method using v_{lon} and v_{lat} as prediction input [10] we could improve this value from 252 µs by a factor of 7, which is mainly due to faster preprocessing and exploiting the fast sliding window polynomial update functions for the $|v|$ and ω time series. The processing times of single steps are shown in Table 4.

Table 4 Processing time of the single modules, averaged over the entire test data

Module	Processing time (μs)
Track preprocessing	< 1
Polynomial fitting	14
Neural network prediction	5
Prediction reconstruction	15

Table 5 Comparison of *AEE* of MLP to SVR method for different prediction times

Pred. time (s)	0.5	1.0	1.5	2.0	2.5
AEE (cm) (MLP)	9.3	19.4	32.3	48.0	66.1
AEE (cm) (SVR)	9.4	19.9	33.4	49.5	68.6
Comparison (SVR: 100 %)	−1.1 %	−2.5 %	−3.3 %	−3.0 %	−3.6 %

The offline training of the predictor was performed on the same machine. It needed computational times between 20 s and 6 min for 643 tracks depending on the used polynomials and neural network topology.

4.6 Comparison to Support Vector Regression

For the purposes of comparison to a kernel-based alternative to MLP we also investigated Support Vector Regression (SVR, see [2]). Therefore, the MLP prediction module is substituted by an ε-SVR based on the LibSVM library [3] while the polynomial input remains unchanged.

We applied a radial basis function (RBF-) kernel and optimized the parameters C, γ, and ε by a coarse-to-fine grid search. Because SVR only allows a single output, we trained two instances to predict 2D positions for single prediction times in each case instead of a continuous estimation over a timespan based on the output polynomial coefficients. The resulting *AEE* values and a comparison to the above approach are set out in Table 5.

The evaluation shows a slight advantage of the MLP method. It yields smaller *AEE* values for all tested prediction horizons while the improvement rises with higher prediction times up to 3.6 % for 2.5 s. Besides the lower prediction errors a major benefit of the neural network is the capability to predict positions for a continuous future timespan with a single instance at the same time.

5 Conclusion

In this publication we proposed a method for the prediction of pedestrian trajectories by polynomial least-squares approximations in combination with MLP neural networks. The method is trained and tested on 1075 different tracks of pedestrians in real urban scenarios. Our implementation features the prediction of a continuous future trajectory for a time horizon of 2.5 s using camera-based head tracking data of the most recent time interval of 1.0 s as input. Due to the usage of a self-learning method as "implicit" movement model, a sole prediction technique is capable of handling different movement types crucial for traffic safety, e.g., starting and stopping. The polynomial approximation of the input tracks provides great flexibility as it allows for independence from sensor type and sampling rate. For a prediction time of 1 s our tests result in average Euclidean errors of 27 cm for starting and 22 cm for stopping scenes, and for 2.5 s in 98 and 86 cm, respectively. The proposed method outperforms the prediction quality of a CV model Kalman filter by 27.4 % over all test data, by 28 % for starting and by 32 % for stopping scenes.

Our future work will include the application of the method to bicyclists, who constitute another important proportion of VRUs. A vehicle-based implementation of the method requires an additional ego-motion compensation as the pedestrian track in global coordinates is used as input. Also, systems based on a forward-looking stereo camera will be investigated. As cars have a limited view on traffic scenarios and because not all intersections will be equipped with sensors, we envision a scenario where the protection of VRUs is realized in a cooperative way. The collective intelligence of cars, complemented by information from infrastructure (where available) and VRUs themselves (if equipped with intelligent devices such as smartphones) will be exploited to detect the intention of VRUs in a distributed way. This approach will not only provide an essential component for future traffic automation, it will also increase the safety of road users.

Acknowledgments This work partially results from the project AFUSS, supported by the German Federal Ministry of Education and Research (BMBF) under grant number 03FH021I3, and the project DeCoInt^2, supported by the German Research Foundation (DFG) within the priority program SPP 1835: "Kooperativ Interagierende Automobile", grant numbers DO 1186/1-1 and SI 674/11-1.

References

1. Bar-Shalom, Y., Li, X.R., Kirubarajan, T.: Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software. Wiley, New York (2001)
2. Chang, C.C., Lin, C.J.: Training ν-support vector regression: theory and algorithms. Neural Comput. **14**(8), 1959–1977 (2002)
3. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 27:1–27:27 (2011). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. Accessed 9 Dec 2015
4. Euro NCAP: Euro NCAP 2020 Roadmap (2015). http://euroncap.blob.core.windows.net/media/16472/euro-ncap-2020-roadmap-rev1-march-2015.pdf. Accessed 9 Dec 2015
5. Fuchs, E., Gruber, T., Nitschke, J., Sick, B.: On-line motif detection in time series with SwiftMotif. Pattern Recognit. **42**(11), 3015–3031 (2009)
6. Fuchs, E., Gruber, T., Nitschke, J., Sick, B.: Online segmentation of time series based on polynomial least-squares approximations. IEEE Trans. Pattern Anal. Mach. Intell. **32**(12), 2232–2245 (2010)
7. Gensler, A., Gruber, T., Sick, B.: Fast approximation library. http://ies-research.de/Software. Accesed 17 Dec 2013
8. Goldhammer, M., Doll, K., Brunsmann, U., Gensler, A., Sick, B.: Pedestrian's trajectory forecast in public traffic with artificial neural networks. In: Proccedings of the 22nd International Conference on Pattern Recognition (ICPR), pp. 4110–4115 (2014)

9. Goldhammer, M., Hubert, A., Köhler, S., Zindler, K., Brunsmann, U., Doll, K., Sick, B.: Analysis on termination of pedestrians' gait at urban intersections. In: Proceedings of the IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), pp. 1758–1763 (2014)
10. Goldhammer, M., Köhler, S., Doll, K., Sick, B.: Camera based pedestrian path prediction by means of polynomial least-squares approximation and multilayer perceptron neural networks. In: Proceedings of the SAI Intelligent Systems Conference, pp. 390–399 (2015)
11. Goldhammer, M., Strigel, E., Meissner, D., Brunsmann, U., Doll, K., Dietmayer, K.: Cooperative multi sensor network for traffic safety applications at intersections. In: Proceedings of the IEEE 15th International Conference onIntelligent Transportation Systems (ITSC), pp. 1178–1183 (2012)
12. Haykin, S.: Neural Networks and Learning Machines. Prentice Hall, New York (2009)
13. Kalarot, R., Morris, J.: Comparison of fpga and gpu implementations of real-time stereo vision. In: Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 9–15 (2010)
14. Koehler, S., Goldhammer, M., Bauer, S., Zecha, S., Doll, K., Brunsmann, U., Dietmayer, K.: Stationary detection of the pedestrian's intention at intersections. Intell. Transp. Syst. Mag., IEEE **5**(4), 87–99 (2013)
15. Naujoks, F.: How Should I Inform my Driver? (2013). http://ko-fas.de/files/abschluss/ko-fas_c1_4_effective_advisory_warnings_based_on_cooperative_perception.pdf. Accessed 9 Dec 2015
16. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 586–591 (1993)
17. Winter, D.A.: Human balance and posture control during standing and walking. Gait Posture **3**(4), 193–214 (1995)
18. World Health Organization: Global Status Report on Road Safety 2013: Supporting a Decade of Action (2013). http://www.who.int/violence_injury_prevention/road_safety_status/2013/en. Accessed 9 Dec 2015