2014 IEEE Intelligent Vehicles Symposium (IV)
June 8-11, 2014. Dearborn, Michigan, USA

Crowdsourced intersection parameters: A generic approach for extraction and confidence estimation.

Christian Ruhhammer[1], Nils Hirsenkorn[1], Felix Klanner[1] and Christoph Stiller[2]

Abstract— **Digital maps within cars are not only the basis for navigation but also for advanced driver assistance systems. Therefore more and more up-to-date details about the environment of the vehicle are required which means that they have to be enriched with further attributes such as detailed representations of intersections. In the future we will be able to extract details of the environment out of the sensory data of connected cars. We present a generic approach for extracting multiple intersection parameters with the same method by analyzing logged data from a test fleet. Based on that a method for a feature based estimation of the confidence is introduced. The proposed approaches are applied in a completely automated process to estimate stop line positions and traffic flows at intersections with traffic lights. Altogether 203.701 traces of the test fleet were used for developing and testing. The performance of the method and the confidence estimation were analyzed using a ground truth, consisting of 108 stop line positions, which was derived from satellite images. The results show that the approach is fast and predictions with an absolute accuracy of 3.5 m can be achieved. Hence the method is able to deliver valuable inputs for driver assistance systems.**

I. INTRODUCTION

Many modern driver assistance systems are already based on an internet connection between the cars and a back-end server system. In the context of this paper, a back-end system means a central server within a client-server architecture. The clients of this architecture are cars.

Examples for back-end based driver assistance systems are the representation of current traffic information within vehicle navigation systems (e.g. BMW Real Time Traffic Information, Audi Traffic Information Online). This information is also already available through web browsers (e.g. through the Google Maps Traffic overlay) and smart-phone applications such as INRIX Traffic. Another online service in modern cars which requires a connection to a back-end server system is the streaming of music. By now the applications of internet based systems in cars are focused on navigation, local hazard warnings and entertainment. The popularity of internet based systems increases so the amount of cars connected to the internet rises continuously.

One very important aspect for the development of new assistance systems and the improvement of existing ones is knowledge about the context of the current and foresight on the upcoming driving situation. Therefore enhancements towards more detailed digital maps containing more information about the static environment are necessary. With detailed maps, the safety and comfort of driving can be improved furthermore [1], [2].

Examples of missing details to understand the driving situation are parameters of intersections such as the traffic control or the exact position of stop lines. Several existing and future driver assistance systems could benefit from the knowledge of these parameters. Possible benefits arise by

- delivering a-priori knowledge for camera based assistance systems (e.g. [3]),
- adapting the control strategy of the engine start-stop system,
- delivering input for driver intent inference [4] (e.g. collision avoidance),
- using stop lines as landmarks to improve the longitudinal positioning [5].

Currently this kind of information is generated by manual measurements and mapping with specially equipped cars, producing a grand effort. Another approach to get the detailed information is Crowdsourcing. In the context of this work, Crowdsourcing means that data is collected over many cars equipped with standard sensors. The individual cars can then be provided with the collectively learned information which can be used by driver assistance systems up to highly automated driving. One possibility to collect the data and to provide the extracted information is to use the internet.

The idea to extract street models out of numerous logged GPS trajectories was picked up in previous works [6], [7]. The aim was to extract lane information out of logged position data. Furthermore in [8] a method was presented to detect the existence of traffic lights by analyzing logged intersection approaches. More information about traffic lights has been extracted from velocity profiles in [9] and [10] to enable the development of more efficient driving strategies.

The described previous works in the domain of Crowdsourcing refer mainly to the geometric description of the road as well as extracting information about traffic lights. The presented approaches to extract information about intersections are special for specific parameters. For the validation of driver assistance systems the complexity of a system is an important factor. Therefore this work proposes a generic approach to learn several intersection parameters simultaneously with one method, which reduces the complexity. Knowing the confidence of the result is also a crucial information which is addressed in this paper.

Section II gives an overview of the approach and presents the accuracy achieved by applying it to a test dataset.

[1]Christian Ruhhammer, Nils Hirsenkorn and Felix Klanner are with BMW Group, Research and Technology, D-80788 Munich, Germany. {firstname.lastname}@bmw.de

[2]Christoph Stiller is with Karlsruhe Institute of Technology, Department of Measurement and Control, D-76131 Karlsruhe, Germany. stiller@kit.edu

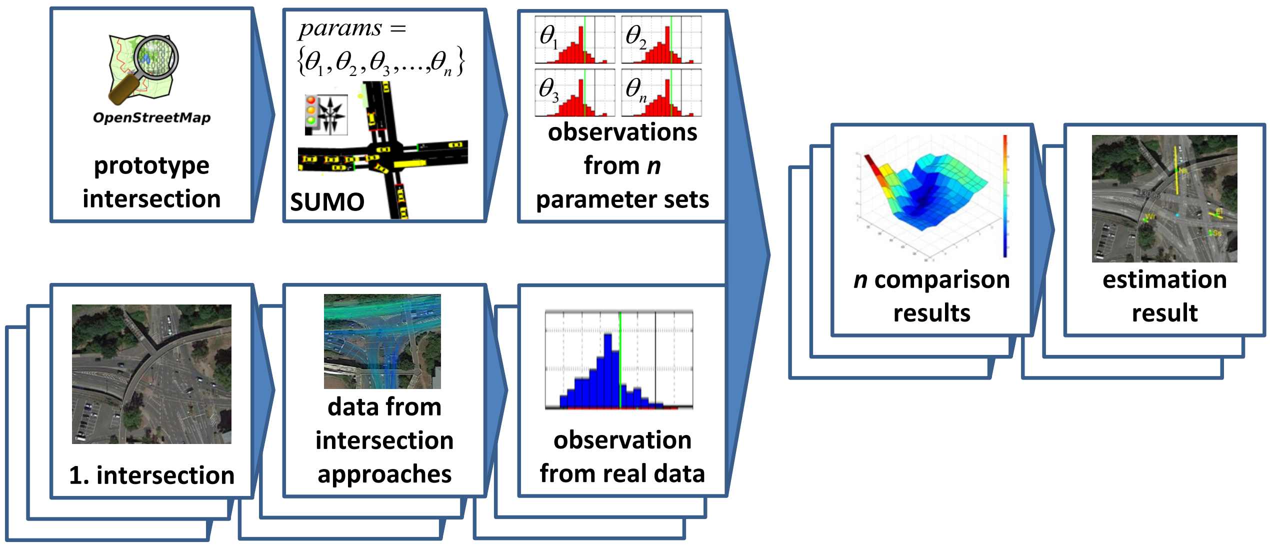

Fig. 1. Intersection parameter estimation by comparing simulated observations with different parameters with real observations. The simulated observations are the result of using different parameter sets as input for the open source traffic simulator SUMO.

It is essential for many applications to know about the confidence of an estimation. Therefore in section III a feature based approach for estimating the confidence is described, which can also be used as a general method for confidence estimation. Finally in section IV the approach and the results on the test dataset are discussed and the work ends with the conclusions in section V.

II. EXTRACTION OF INTERSECTION PARAMETERS WITH CROWDSOURCING

With Crowdsourcing of standard on-board sensors, a variety of information about the cars environment can be extracted through intelligent aggregation. This work focuses on the extraction of intersection parameters and especially on traffic flows at intersection entries and stop line positions.

A. Simulation based approach for parameter estimation

A possible solution for the extraction of intersection parameters from logged crossings is the analysis of observable variables of which the unobservable parameters depend on. To estimate the position of stop lines for example, the distribution of waiting locations could be analyzed. A description or a model of the dependence between the stop line position and the distribution of waiting locations would be the solution for our problem. The probabilities for stopping at a certain position vary across different parameters of an intersection such as

- number and connectivity of lanes,
- probabilities of straight and turning maneuvers,
- position of the stop line if it exists,
- absolute traffic flow at intersection entries.

The application of a machine learning approach to learn stop line positions would require a lot of training data. That means data from a lot of intersections is needed because of the high variability between them. The arrival of cars at intersections is a random queuing process with a queue for every driving lane. Since cars change lanes the process extends to a linked queuing problem which cannot be formulated with the queuing theory.

Therefore in [11] an approach was presented which is based on a simulation of the traffic with the open source simulator SUMO [12] at a specific intersection for one day. As an input to the simulation, known parameters like turning probabilities are extracted out of data from real intersection crossings. *OpenStreetMap* (OSM) [13] is used to get the topology and geometry of the simulated intersection. The simulation was conducted multiple times, varying the unknown parameters within a discrete space. For every combination of the parameters the resulting data of the corresponding simulation was compared to the data logged from intersection approaches of real cars. Basis for the comparison are observable variables in both datasets, e.g. the aforementioned distribution of the stopping positions. The parameter combination with the best match between the simulated and the real approaches is the estimation of the parameters. With this simulation based approach it is not necessary to build a model for the learning of intersection parameters with training data. The approach is completely based on a microscopic traffic simulation which has not to be trained.

B. Prototype based approach for parameter estimation

The simulation based approach has two major drawbacks:

- Detailed information about the topology and geometry of the intersection is required
- and the approach is time-consuming, as the simulation has to be executed for every parameter combination and every intersection.

Hence in this work the simulation based approach was adapted to a prototype based approach in order to counteract these disadvantages. Figure 1 gives a brief overview of the new concept. The upper path in the figure shows that a generic intersection representation from OSM is selected first. For this prototype intersection, simulations with SUMO are conducted only once but with different combinations of the parameters we want to estimate. The observable variables can be extracted out of every single simulation. For n different parameter combinations $\theta_i \in \mathbf{P}$ we get a set of n observations

$$\mathbf{O} = o_{\theta_1}, o_{\theta_2}, o_{\theta_3}, ..., o_{\theta_n}. \quad (1)$$

In parallel to the simulation, data for multiple intersections is extracted from a fleet of cars. The processing of the logged crossings results in an observation o_{real}. The compare function

$$\text{compare} : \mathbf{O} \times \mathbf{O} \rightarrow \mathbb{R} \quad (2)$$

maps two observations $o \in \mathbf{O}$ to a real number $w \in \mathbb{R}$, which serves as a measurement of equality between the observations. All simulated observations $o_{\theta_i} \in \mathbf{O}$ are compared to the observation o_{real} of the logged intersection crossings. With this comparison we have a measurement

$$\forall i \in \{1, ..., n\} : w_i = \text{compare}(o_{\theta_i}, o_{real}) \quad (3)$$

for every parameter combination $\theta_i \in \mathbf{P}$ used for simulation. If the equality measurement is seen as a distance of the observations, the estimation result is the parameter combination $\theta_{i^*} \in \mathbf{P}$ belonging to the observation $o_{\theta_i^*}$ where

$$\forall i \in \{1, ..., n\} : \text{compare}(o_{\theta_i}, o_{real}) \geq \text{compare}(o_{\theta_i^*}, o_{real}). \quad (4)$$

This means that the observation $o_{\theta_i^*}$ with the minimal distance to the observation o_{real} from the logged intersection crossings belongs to the simulation with the best matching parameter combination.

Compared to the simulation based approach, the prototype approach has the advantages that:

- No detailed map information about the intersection is required, as only one prototype intersection is simulated
- and the approach is fast, as the simulation has not to be executed for every intersection but only for one prototype.

C. Estimation of stop line positions and traffic flows

In this work, the generic prototype based method for estimating intersection parameters is used to extract stop line positions and traffic flows at intersections with traffic lights. These parameters influence the distribution of vehicle stops relative to the intersection center. Therefore the distribution was used as observational variable.

The logged crossings of an intersection were allocated to specific intersection entries. There are different stop lines at some intersections for the different possible maneuvers turning right, going straight and turning left. By calculating the distribution of stop positions for every path through the intersection, it is possible to estimate the parameters separately. A "path" is the combination of an intersection entry and a maneuver.

To model the dependence on the distribution of stop points to different traffic flows, SUMO was used to create prototypes for these observations. The stop line position is defined as the distance of the line to the center of an intersection projected on the driving direction of the corresponding entry. Intersection centers where extracted automatically out of OSM. It becomes apparent that the stop line position has no influence on the shape of the stop point distribution. It only affects the translational position of the distribution in reference to the center of the intersection. Therefore only the traffic flow parameter has to be simulated within a discrete parameter space to generate prototypical distributions, whereas the stop line position is set to any constant value. To get the complete set $\mathbf{O}$ of observations for n different parameter combinations, the prototypical distributions of different traffic flow parameters are moved translational to the center of the intersection. The movement is conducted within a specified parameter range for the unknown stop line position.

Hence we get the set of all possible parameter combinations

$$\mathbf{P} = \mathbf{Q} \times \mathbf{S} \quad (5)$$

by combining the discrete codomains $\mathbf{Q}$ and $\mathbf{S}$ of the parameters, traffic flow $q \in \mathbf{Q}$ and stop line position $s \in \mathbf{S}$. For the traffic flow we choose the range $\mathbf{Q} = [50 : 20 : 450]\,1/\mathrm{h}$ and for the stop line position we choose $\mathbf{S} = [0 : -1 : -30]\,\mathrm{m}$. The negative sign of the stop line position means that it is placed before the center of the intersection.

As the stop positions are based on GPS measurements, it is important to model the noise in the simulation output in order to get comparable observations. We used Gaussian white noise without mean as a simple model for the GPS noise. For the comparing function between observations we used the "Match Distance", also known as "Earth Mover's Distance" [14] according to the best results in [11].

D. Test results for stop line positions

The developed methods were tested with logged data from the German research project sim^{TD} [15]. Out of this data, connected velocity and position (GPS) values of single intersection approaches are extracted. For every intersection path within our set of intersections a different number of traces was recorded. More traces induce more stop points, so the true distribution of stop points can be approximated more exactly.

All together 203.701 intersection traces over 54 intersections were used which result in 108 possible maneuver paths from 91 intersection entries. The test was carried out for stop line positions only because there was no ground truth data available for the traffic flow. The source of the ground truth for the stop line positions were satellite images. Because the approach does not need any training data, every single maneuver path can be used for testing. Figure 2 shows the test results of the prototype based approach under variation of the used number of stops according to table I.

We randomly decreased the number of traces per intersection path to investigate the dependency of the number of stops on the performance. If more stops should be used, the number of intersections in our dataset declines with the minimum required number of stops. The reason is that the dataset contains intersections with only a few traces. A positive error in the results equates to a prediction after the real stop line position referred to the driving direction. The box plots show for example that by utilizing a minimum of 700 stops, an estimation within a range of $[-3.5\,\mathrm{m},\, 2.2\,\mathrm{m}]$ can be achieved if the three outliers are ignored.

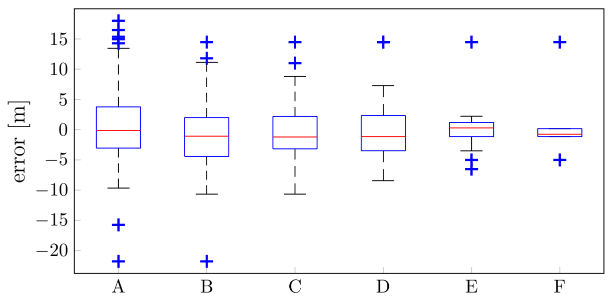

Fig. 2. Test results of the prototype based approach. For every result (A-F) a differnt number of stops was used out of all stops available for the path. Table I shows the different test configurations.

TABLE I

CONFIGURATIONS FOR THE RESULTS IN FIGURE 2.

box label	A	B	C	D	E	F
used nr. of stops	20	200	400	700	1400	2000
# of paths with enough stops	108	60	42	29	14	6

III. CONFIDENCE ESTIMATION WITH SPARSE TEST DATA

The knowledge about the confidence of a specific estimation is nearly as important as a good result. The outcome of our estimation is not based on probabilistic models therefore the confidence cannot be calculated specifically for every result. A basic approach is to calculate a constant value for a given confidence level out of all resulting estimation errors of a test set. This value could be used for further estimations. The idea behind the proposed feature based method of this work is to get a *specific* confidence interval for every result directly. The method uses features out of the parameter estimation process introduced in section II. Specific features are selected which reflect the reliability of the result. So basically variables have to be found which correlate with the estimation error, whereas overestimation of the error is allowed but underestimation should be avoided. The object is to estimate an upper boundary of the absolute error and not the error directly. This means that we assure a specific absolute error for a given confidence level.

A. Features for estimating confidence intervals

In the presented application of the prototype approach we identified three possible features for estimating a confidence of a result,

- the utilized number of stop positions as basis for the observed variable ”stop position distribution” (f_{NrSt}),
- the value of the observation comparison for the estimated result, which means the global minimum of comparison values (f_{MinComp}),
- the second derivative of the comparison values at the certain point which corresponds to the estimated result (f_{SecDer}).

The distribution of stop positions is used as the observed variable and therefore the complete method is based on that distribution. We make the assumption that the better the distribution reflects the real distribution, the better the intersection parameters can be estimated. To get a good representation of the distribution as much stop positions as possible should be used. Therefore we suppose that the number of used stop positions contains information about the estimation error.

Other features can be extracted out of the set of comparison measurements for the different parameter combinations. The parameter combination which belongs to the minimum of all the distance measurements is the estimation result. Therefore the absolute value of this minimal measurement contains information about the confidence of the result. The better the observations match to each other, the more confident is the parameter estimation.

Another feature considers the values of the comparison measurements around the global minimum. If there are more values around the minimum with almost similar values, the confidence should be low. Otherwise, if the distance values near to the global minimum tend to be high, there are nearly no other parameter combinations which also fit to the observation. This can be expressed by the second derivative of the observation comparison values at the global minimum.

The following work is based on estimating the confidence of the stop line position estimation. The vehicle stops at stop lines are biased towards negative offsets because most of the drivers stop in front of the line. Hence the following predicted errors of the test set where centered around the mean offset.

The stars in figure 4 and the points in figure 5 show the dependency of one normalized feature on the absolute error of estimated stop line positions. As an exemplary feature we used the number of available stop positions for the plots.

B. The confidence estimation function

To get the confidence of a new result we define a function

$$\mathrm{confEstimator} : \mathbf{F} \rightarrow \mathbb{R} \tag{6}$$

which maps a specific feature value $f \in \mathbf{F}$ to an absolute error estimation $\hat{e}_i \in \mathbb{R}$ for every result $i \in \{1, ..., m\}$. In the case of estimating the position of a stop line, $e_i \in \mathbb{R}$ is the absolute error of the result in meters centered around the mean and $\hat{e}_i$ equates to the estimated absolute error resulting from the function *confEstimator*. For a given confidence level $1 - \alpha$ the condition

$$\frac{|\{\hat{e}_i | \hat{e}_i \geq e_i, i \in \{1, ..., m\}|}{m} \geq 1 - \alpha \tag{7}$$

must be hold. This means that the function *confEstimator* should not underestimate the real error for more than $\alpha \cdot n$ estimations. To create a function which meets this requirement, four methods are introduced.

1) Iteratively reweighted least squares (IRWLS): The first proposed method to generate the confidence estimation function is based on a weighted least squares estimator whereas the weights are updated iteratively [16]. For the first step of the process a usual weighted least squares estimator

$$\hat{\beta} = (X^T W X)^{-1} X^T W \mathbf{e} \tag{8}$$

is applied to the data. The weights matrix W is initialized as an identity matrix. In every following iteration t the weights matrix stays a diagonal matrix where the weights for every test data i are updated by multiplying the previous one with a factor γ

$$w_i^{(t)} = \gamma(\hat{e}_i - e_i) \cdot w_i^{(t-1)}. \tag{9}$$

To meet the condition for a given confidence level, the weights should be increased for points where the error was underestimated in the previous iteration. Therefore we heuristically defined a function for the factor

$$\gamma(x) = \begin{cases} 1 + a \cdot (\exp(b \cdot x) - \exp(c \cdot x)) & , x \leq 0 \\ 1 - d \cdot (\exp(-b \cdot x) - \exp(-c \cdot x)) & , x > 0 \end{cases}. \tag{10}$$

The function of the factor for the weights is plotted in figure 3. For underestimated points close to the curve $((\hat{e}_i - e_i) < 0)$, the factor increases the weights. Therefore the points move closer to the estimated function in further iterations or even change to overestimated points. Outliers should not influence the curve, hence the factor for underestimated points far away from the curve $((\hat{e}_i - e_i) << 0)$ tends to 1. The factor has an inverted behavior for overestimated points.

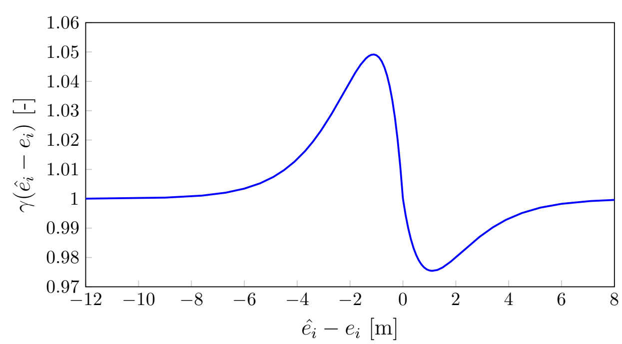

Fig. 3. Heuristically defined weights function for the iteratively reweighted least squares algorithm.

Figure 4 shows how the weights function changes the curve of the least squares estimator between two iterations. A third order polynomial is used for the estimator in this case. The resulting confidence estimation function is generated by adding the condition of a monotony falling curve to the polynomial. Additionally a minimum estimated error of 2 m was set as a further condition. In this figure, points with underestimated errors lie above the curve of the estimator. The weights "pull" the curve which leads to an overestimation of more points.

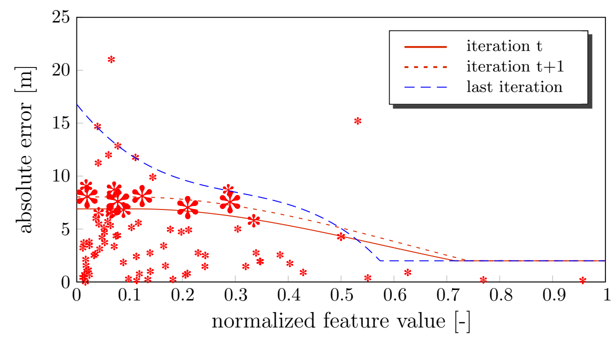

Fig. 4. Three iteration results (t, $t+1$ and the last one calculated) of the iteratively reweighted least squares algorithm. The size of the markers is proportional to their weights of iteration t.

2) Convex hull peeling (CHP): Another investigated approach to get a confidence estimation function is based on convex hull peeling. The method first builds a convex hull in the feature-error space over all points of the test set. The error estimation function is generated by extracting an upper boundary of the hull and extending it to a monotonous falling function. Again the condition of a minimum estimated error of 2 m is set. The steps are repeated by applying the peeling method, which means that the points of the convex hull from the last iteration are not considered for extracting the new hull. The principle is shown in figure 5.

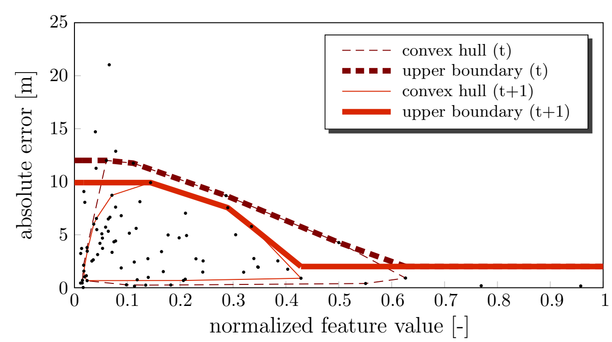

Fig. 5. Two iteration results (t and $t+1$) of the convex hull peeling method.

3) Brute force: Besides using the IRWLS approach to fit a polynomial to the features of the test data we also tried a brute force approach with a 3rd order polynomial. Altogether 6656 different parameter combinations for four parameters were tested.

4) Constant absolute error (reference approach): As a reference we also tested the application of a constant error estimation based on the test results which are centered around the mean. In this approach no features are used, so no specific confidence intervals are estimated, but a constant value within a discrete codomain.

C. Dimensionality reduction of the feature space

Besides using only one feature, we also tested the methods proposed in III-B in a multidimensional space with more features. The results were not satisfying as the confidence estimation seems to overfit with our small test data set. To avoid this problem without losing too much information, we developed a transformation to a one dimensional feature space.

To be able to apply the methods on a one dimensional feature space without losing information of other features we reduced the three features to one combined feature. In order to find a good transformation for this dimensionality reduction, an inverse regression was applied. Basis for this method is the definition of an "optimal" confidence estimation function. This function is defined to be a fixed line with a negative gradient in the feature-error space where the feature space of the training set is normalized in a range $[0, 1]$. The maximum observed error e_{max} maps to a feature value of 0 and the maximal feature value 1 maps to the minimum possible absolute error of $0\,\mathrm{m}$. The optimal feature value f_{opt_i} for every observation of the test data set can be calculated by the error e_i of the data and the inversion of this fixed line:

$$f_{opt_i} = 1 - \frac{e_i}{e_{max}}. \tag{11}$$

The object of the transformation of the three features to a new transformed feature value $\hat{f}_i$ is to minimize the distance $f_{opt_i} - \hat{f}_i$ to the defined optimal feature value. Therefore we perform a dimensionality reduction on the previously introduced features by applying a least squares minimization on the linear combination of the features including powers of every feature. With the resulting coefficients we can calculate the transformed feature values $\hat{f}_i$ for every observation.

D. Results for confidence estimation approaches

To compare all different approaches for confidence estimation, every approach was applied to our test data set. For this analysis a *leave-one-out cross-validation* (LOOCV) was implemented. The three different features (f_{NrSt}, f_{MinComp}, f_{SecDer}) were used alone as well as combined in the transformed feature $\hat{f}$.

For a generated confidence estimation function, two values are significant to the performance of the function. One value is the confidence level

$$1 - \hat{\alpha}_k = \frac{|\{\hat{e}_j|\hat{e}_j \geq e_j|}{m} \tag{12}$$

which is reached on the test data (with LOOCV) with this function. The confidence level should be as high as possible, but the minimal requirement depends on the application. The other value is the average of the error estimations

$$\overline{\hat{e}_k} = \frac{\sum\limits_{j \in \{j|\hat{e}_j \geq e_j\}} \hat{e}_j}{|\{j|\hat{e}_j \geq e_j\}|} \tag{13}$$

for all data where the error was correctly overestimated. This value should be as low as possible, as we want to estimate the confidence interval as narrow as possible. The decision for a specific confidence estimation function is a trade of between these two values. This trade of is illustrated in figure 6 for different parameters of the proposed methods. Depending on the requirements of the application a specific working point on the curve of a method leads to the associated parameters for the estimation function.

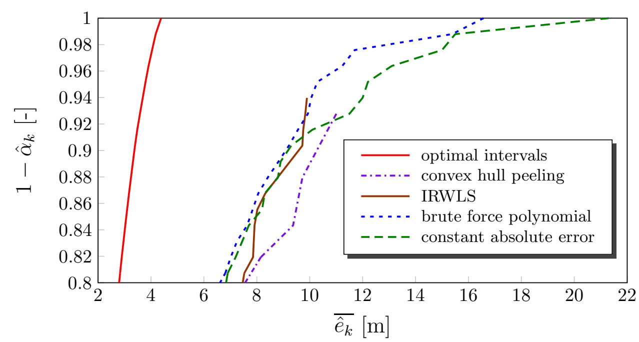

Fig. 6. Test results for different confidence estimation methods by applying *leave-one-out cross-validation* for every method on the test data set. Basis for the test is the transformed feature $\hat{f}$.

IV. DISCUSSION

Compared to the approach in [11], the execution time of estimating the parameters of one intersection could be reduced from several hours to less than one second on a standard desktop computer. Therefore the approach is scalable to whole cities. For example in Munich 14942 intersection centers were extracted from OSM. Furthermore the prototypical approach does not necessarily need a-priori-knowledge about the topology of the intersection. Hence the proposed method resolves the two major drawbacks of the simulation based approach. An exemplary result for the estimation of six stop lines is shown in figure 7.

For intersection where more than 1400 stop points exist, the prototypical approach achieves an estimation range of $[-3.5\,\mathrm{m}, 2.2\,\mathrm{m}]$ without considering three outliers (see figure 2). This result is even better than the one was achieved in [11] where the offsets where in a range of $[-3\,\mathrm{m}, 5\,\mathrm{m}]$. The error increases to a range of $[-8\,\mathrm{m}, 7\,\mathrm{m}]$ in the case that only a minimum of 700 stops per intersection entry exist. Therefore it can be shown that with a decreasing number of utilized stops, the error increases.

Results of the validation of the different methods for confidence estimation imply that the brute force approach performs best on our test data set. That approach has the disadvantage of a much longer execution time compared to the other approaches (see table II). The IRWLS approach in contrast is fast and reaches a confidence of $94\,\%$ with a very good value for the mean confidence of $9.9\,\mathrm{m}$. For this level of confidence it performs even better than the brute force approach which reaches $10.1\,\mathrm{m}$. But with IRWLS, higher confidence values cannot be reached because an

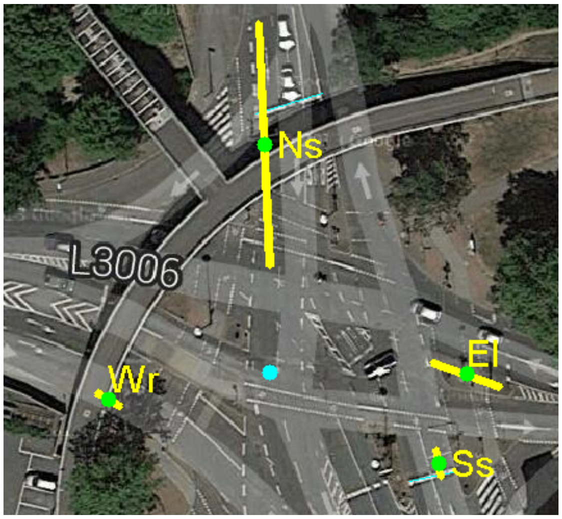

Fig. 7. Estimation of stop line positions and the confidence intervals for a sample intersection in Frankfurt based on the sim^{TD} data set. The predictions are marked with the entrance cardinal direction and the maneuver. For example "Sl" means the stop line of the southern entrance for left turning maneuvers.

overestimation of every point in the test data is not possible by applying a least squares regression. The performance of the CHP method at the LOOCV leads to the assumption that it overfits our test data. This approach reaches a maximum of $92\,\%$ for the confidence level.

The constant error estimation serves as a reference to the other approaches. If levels below $90\,\%$ are required, the complex approaches do not have great advantages. But for applications which require very high confidence levels, the IRWLS and brute force polynomial approaches for confidence estimation have performance advantages over the reference approach. For example the IRWLS estimates about $2\,\mathrm{m}$ less confidence intervals in average than the constant approach for a confidence level of $94\,\%$. At a level of $97\,\%$ the brute force approach is able to estimate a mean confidence of only $11.7\,\mathrm{m}$ compared to $15\,\mathrm{m}$ with the constant estimation.

TABLE II

EXECUTION TIMES AND EXEMPLARY PERFORMANCE OF CONFIDENCE ESTIMATION APPROACHES.

approach	**IRWLS**	**CHP**	**Brute force**	**Constant**
execution time [ms]	147	11	1468	21
$\overline{\hat{e}_k}$ [m] for 1 $\hat{\alpha}_k = 0.92$	9.8	11.0	9.9	11.5

V. CONCLUSIONS

The proposed methods deliver a generic approach for the estimation of intersection parameters out of logged data from car fleets. Compared to the already proposed simulation based approach in [11], the main improvements of this work are that the algorithm needs less time for execution and that no detailled maps of intersections are required anymore.

A framework was implemented which automatically extracts intersection centers and performs the parameter and confidence estimation for a given test data set. Introducing the estimation of a confidence level is a very important step towards the integration of crowdsourced static environment parameters into future driver assistance systems.

Further improvements can be achieved by creating more observation prototypes through simulating more different intersections with different topologies. If we have knowledge about the intersection it is also possible to intelligently preselect the prototypes.

REFERENCES

[1] V. Blervaque, K. Mezger, L. Beuk, and J. Loewenau, "ADAS Horizon - How Digital Maps can contribute to Road Safety," in *Advanced Microsystems for Automotive Applications 2006* (J. Valldorf and W. Gessner, eds.), VDI-Book, pp. 427–436, Berlin: Springer-Verlag, 2006.

[2] B. Thomas, J. Lowenau, S. Durekovic, and H.-U. Otto, "The ActMAP - FeedMAP framework A basis for in-vehicle ADAS application improvement," in *IEEE Intelligent Vehicles Symposium*, pp. 263–268, IEEE, June 2008.

[3] S. Gehrig, S. Wagner, and U. Franke, "System architecture for an intersection assistant fusing image, map, and GPS information," in *IEEE Intelligent Vehicles Symposium. Proceedings*, pp. 144–149, IEEE, 2003.

[4] M. Liebner, C. Ruhhammer, F. Klanner, and C. Stiller, "Generic Driver Intent Inference based on Parametric Models," in *IEEE Intelligent Transportation Systems (ITSC 2013)*, 2013.

[5] O. Pink, F. Moosmann, and A. Bachmann, "Visual features for vehicle localization and ego-motion estimation," in *IEEE Intelligent Vehicles Symposium*, pp. 254–260, IEEE, June 2009.

[6] S. Rogers, P. Langley, C. Wilson, and P. Alto, "Mining GPS Data to Augment Road Models," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999.

[7] S. Rogers, "Creating and Evaluating Highly Accurate Maps with Probe Vehicles," in *IEEE Intelligent Transportation Systems*, 2000.

[8] S. Hu, L. Su, H. Liu, H. Wang, and T. Abdelzaher, "SmartRoad : A Crowd-Sourced Traffic Regulator Detection and Identification System," in *Proceedings of the 12th international conference on Information processing in sensor networks*, 2013.

[9] M. Kerper, C. Wewetzer, and M. Mauve, "Analyzing vehicle traces to find and exploit correlated traffic lights for efficient driving," *IEEE Intelligent Vehicles Symposium*, June 2012.

[10] M. Kerper, C. Wewetzer, A. Sasse, and M. Mauve, "Learning Traffic Light Phase Schedules from Velocity Profiles in the Cloud," *International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, May 2012.

[11] C. Ruhhammer, A. Atanasov, F. Klanner, and C. Stiller, "Crowdsourcing als Enabler für verbesserte Assistenzsysteme: Ein generischer Ansatz zum Erlernen von Kreuzungsparametern," in *9. Workshop Fahrerassistenzsysteme: FAS2014, Walting*, Uni-DAS e.V., Mar. 2014.

[12] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent Development and Applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, 2012.

[13] M. M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps," *IEEE Pervasive Computing*, vol. 7, pp. 12–18, Oct. 2008.

[14] Y. Rubner, C. Tomasi, and L. J. Guibas, "The Earth Mover's Distance as a Metric for Image Retrieval," *International Journal of Computer Vision*, vol. 40, pp. 99–121, Nov. 2000.

[15] C. Weiß, "V2X communication in Europe - From research projects towards standardization and field testing of vehicle communication technology," *Computer Networks*, vol. 55, pp. 3103–3119, Oct. 2011.

[16] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Communications in Statistics - Theory and Methods*, vol. 6, pp. 813–827, Jan. 1977.