

Smartphone Based HD Map Building for Autonomous Vehicles

Lóránt Szabó

Budapest University of Technology and
Economics, Faculty of Transportation
Engineering and Vehicle
Engineering, Department of Automotive
Technologies

Budapest, Hungary

szabo1994lorant@gmail.com

László Lindenmaier

Budapest University of Technology and
Economics, Faculty of Transportation
Engineering and Vehicle
Engineering, Department of Automotive
Technologies

Budapest, Hungary

lindenmaier94@gmail.com

Viktor Tihanyi, PhD

Budapest University of Technology and
Economics, Faculty of Transportation
Engineering and Vehicle
Engineering, Department of Automotive
Technologies

Budapest, Hungary

viktor.tihanyi@gjt.bme.hu

Abstract – *The development of autonomous vehicles requires accurate map information, about the surrounding static landmarks such as lane information. Building up the so called HD Map requires robust sensors, that mainly means special hardware requirements. Thus research in this topic starts with question of availability of an HD Map. This paper describes a method of creating a robust HD Map on highways, including accurate lane information based on a simple smartphone. The sensor data of the phone are filtered by a Kalman-filter. Hence the localization is more accurate during the offline building process and the online recall of lane information as well. Thus based on this lane information, an accurate and robust trajectory can be planned for the autonomous vehicles.*

Keywords—*HD Map, Autonomous vehicles, Lane model, Kalman filter, Dead-reckoning*

I. INTRODUCTION

In order to navigate a vehicle, an accurate representation of the environment is mandatory. It is possible to build up an environment model by onboard vehicle sensors, such as camera, radar and LIDAR. However in many cases it is difficult to provide accurate lane information by these sensors. For instance, in harsh weather conditions, the performance of these sensors is drastically reduced. To this end, the solution is an HD (High Definition) Map that provides indispensable information for autonomous vehicles about the features of its environment with high accuracy level.

Dozens of companies are working on these maps. However, only a small proportion of the road network is covered yet. In addition this service is really expensive. In addition development of localization algorithms is difficult, because different database make hard to compare the results.

This paper presents a production method of HD Map by using only a smartphone and a computer for making a first generation HD Map. This method helps the development of autonomous vehicles by ensuring accurate lane information. In addition allows the efficient evaluation of localization algorithms.

The importance of the topic is indicated by numerous works in it, mentioning varied methods of localization like Multiple Hypotheses Policy [1][2] or Particle Filtering[4] etc. Some of them not referring the exact map data itself what they use for map-matching. Others building map by using diversified and expensive sensor cluster [8][9][10]. For localization and parallel filtering process related works

were researched in the field of lateral vehicle models [11][13] and recursive filters [10][11][12].

Section II. introduces the measurement and the recorded sensors. Section III. describes the process of the sensor data and a filtering method extended with dead reckoning. In Section IV. a way of semi-automatic image processing method is defined, next to the architecture of offline HD Map building. Section V. describes an online localization method in order to analyze the built Map. Section VI. includes the results, conclusion and our future research goals are described in Section VII.

II. MEASUREMENT

The measurement was carried out by a Huawei P20 Pro smartphone, in the highway called M6 of Hungary. This location was chosen because its varied topography, tunnels and bridges, that makes this route a perfect environment to test our method capability. The covered distance is nearly 80 km, back and forth between the highlighted points. During the measurement the data of the GPS, the magnetometer and the linear accelerometer and the image of the camera were logged by the smartphone. One of the problems was the different data rates of the sensors: GPS (1Hz), magnetometer (100 Hz), linear Accelerometer (100 Hz). In addition to data synchronization, dead-reckoning (Section III.) was applied as a solution of different data rates.

In order to synchronize the logged video frames and sensor data we used the audio layer of the video and the acoustic sensor log. By a single clap at the beginning and at the end of the measurement, a relative zero time could be identified.

III. FILTERING OF SENSOR DATA

The necessary sensor data (GPS, yaw rate, etc.), is noisy and often insufficient. According to a related work [14] the accuracy of a simple GPS is about 5 m. Also our measurements confirm this theory. This accuracy value is not enough in application field of localization of autonomous vehicles. Also the frame rate of the GPS is lower than the expected from the side of trajectory planner. Thus a Kalman-filter is applied in order to filter the measured data and to provide those state variables that are needed but not at all or not in each time steps are ensured by the phone. Firstly, a compact vehicle model is needed in the form described as following expression:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{v}_k \quad (1)$$

In (1) $\mathbf{x}_k$ represents the vehicle state in the k-th time step, $\mathbf{F}$ is the transition matrix and $\mathbf{v}_k$ represents the process noise vector. Equation (2) shows the state vector which contains all of the necessary state variables.

$$\mathbf{x}_k = [\psi \quad \dot{\psi} \quad d_x \quad d_y \quad v_x \quad v_y \quad a_x \quad a_y]^T \quad (2)$$

In (2) ψ is the heading angle, $\dot{\psi}$ is the yaw rate, d_x , d_y , v_x , v_y , a_x and a_y are the longitudinal and lateral displacements between two time steps, the velocities and the accelerations. Note that the velocities and accelerations are interpreted in each time step with the respect of actual ego-vehicle coordinate system. After granting the vehicle state, the transition matrix can be given as followings:

$$\mathbf{F} = \begin{bmatrix} 1 & t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & t & 0 & t^2/2 & 0 \\ 0 & 0 & 0 & 1 & 0 & t & 0 & t^2/2 \\ 0 & 0 & 0 & 0 & 1 & 0 & t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

In (3) t represents the sample time, which is set to 0.01 milliseconds.

In the next step a sensor model has to be given. The sensor model is described by (4).

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \boldsymbol{\eta}_k \quad (4)$$

In equation (4) $\mathbf{z}_k$ represents the measurement vector, $\mathbf{H}_k$ is the sensor model matrix and $\boldsymbol{\eta}_k$ is the measurement noise vector. The yaw rate and the magnetic field can be measured by the smartphone. Because the heading can be calculated easily by using the magnetometer data, the heading is assumed to be measured as well. In addition, based on the latitude and longitude GPS coordinates the displacements can be calculated by using the haversine formula in every second. Also the linear accelerations can be measured. Because the fact that the frame rate of the GPS is different from the other sensors of the smartphone, the measurement vector and the sensor model can be given by two cases. The two cases are given by (5) and (6).

$$\mathbf{z}_k = \begin{cases} \mathbf{z}_{k,1}, & \text{if GPS data is not ensured} \\ \mathbf{z}_{k,2}, & \text{if GPS data is ensured} \end{cases} \quad (5)$$

$$\mathbf{H}_k = \begin{cases} \mathbf{H}_{k,1}, & \text{if GPS data is not ensured} \\ \mathbf{H}_{k,2}, & \text{if GPS data is ensured} \end{cases} \quad (6)$$

The two measurement vectors are described by (7) and (8).

$$\mathbf{z}_{k,1} = [\psi \quad \dot{\psi} \quad a_x \quad a_y]^T \quad (7)$$

$$\mathbf{z}_{k,2} = [\psi \quad \dot{\psi} \quad d_x \quad d_y \quad a_x \quad a_y]^T \quad (8)$$

The $\mathbf{H}_{k,1}$ and $\mathbf{H}_{k,2}$ sensor model matrices can be given by (9) and (10).

$$\mathbf{H}_{k,1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$$\mathbf{H}_{k,2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

The process and measurement noises are assumed to be normally distributed noise with zero mean value. The deviations of process and measurement noise vectors are given by (11) and (12).

$$\begin{aligned} \mathbf{D}(\mathbf{v}_k) &= \sigma_{\dot{\psi},noise} \begin{bmatrix} \frac{t^2}{2} & t & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T + \\ &+ \sigma_{wn,x} \begin{bmatrix} 0 & 0 & \frac{t^3}{3} & 0 & \frac{t^2}{2} & 0 & t & 0 \end{bmatrix}^T + \\ &+ \sigma_{wn,y} \begin{bmatrix} 0 & 0 & 0 & \frac{t^3}{3} & 0 & \frac{t^2}{2} & 0 & t \end{bmatrix}^T \end{aligned} \quad (11)$$

$$\mathbf{D}(\boldsymbol{\eta}_k) = \begin{bmatrix} \sigma_{\psi,noise} \\ \sigma_{\dot{\psi},noise} \\ \sigma_{dx,noise} \\ \sigma_{dy,noise} \\ \sigma_{ax,noise} \\ \sigma_{ay,noise} \end{bmatrix} \quad (12)$$

In (11) $\sigma_{\dot{\psi},noise}$ represents the standard deviation of yaw acceleration noise. $\sigma_{wn,x}$ and $\sigma_{wn,y}$ are the deviations of the longitudinal and lateral white noise. In (12) $\sigma_{\psi,noise}$ is the deviation of the measured heading, $\sigma_{\dot{\psi},noise}$ of the yaw rate, $\sigma_{dx,noise}$ and $\sigma_{dy,noise}$ of the longitudinal and lateral displacements, $\sigma_{ax,noise}$ and $\sigma_{ay,noise}$ are the deviations of longitudinal and lateral accelerations. Based on these noise models, the covariance matrices can be assumed as the followings:

$$\mathbf{Q}_k = E(\mathbf{v}_k \mathbf{v}_k^T) = \mathbf{D}(\mathbf{v}_k) \mathbf{D}(\mathbf{v}_k)^T \quad (13)$$

$$\mathbf{R}_k = E(\boldsymbol{\eta}_k \boldsymbol{\eta}_k^T) = \mathbf{D}(\boldsymbol{\eta}_k) \mathbf{D}(\boldsymbol{\eta}_k)^T \quad (14)$$

The initial state is given by the first measurement extended with zeros at hidden state. This step is represented by (15).

$$\mathbf{x}_0 = [\hat{\psi}_0 \quad \frac{d}{dt} \hat{\psi}_0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \hat{a}_{x,0} \quad \hat{a}_{y,0}]^T \quad (15)$$

In (15) $\hat{\psi}_0$, $\frac{d}{dt} \hat{\psi}_0$, $\hat{a}_{x,0}$ and $\hat{a}_{y,0}$ are the measured values of heading, yaw rate, longitudinal and lateral accelerations in the 0-th time step.

The initial covariance matrix is given by the process noise covariance matrix, which is represented by (16)

$$\mathbf{P}_0 = \mathbf{Q}_0 \quad (16)$$

Note that after obtaining the state of the ego vehicle, the longitudinal and lateral displacements have to be transformed into GPS coordinates by using haversine formula. After this step, the vehicle state can be defined recursively with high accuracy even when GPS measurement is not ensured due to tunnels or other reasons.

One can consider that this process does not ensure accurate GPS coordinates just filtered ones. If the GPS coordinates of the starting point were drifted, the whole localization would be so. But in case of trajectory planning the goal is to localize the ego vehicle on the prebuilt HD map. Considering, during the trajectory process the

localization has the same shifting, the trajectory can be planned with high accuracy.

IV. HD MAP BUILDING (OFFLINE PROCESS)

First of all, a simple fast method was developed for scanning lanes on the road. The farther the lane is, the harder to calculate its accurate position. In addition, there is no point in perceiving further points, since every farther one will be scanned later, during the whole movement.

The lane detection was based on fusion of deep neural network and the combination of color and gradient approach. In HSV color space the raw picture is processed based on the upper 20% spectrum of the histogram. Area opening is used to remove small artifacts and particles. Final step on the binary image was labeling of connected components in order to match with the result of gradient approach. Labeling also describe the ROI-s on the picture. The red color identifies a section of the picture which can be ignored during the gradient approach. This can be seen on Fig. 1.

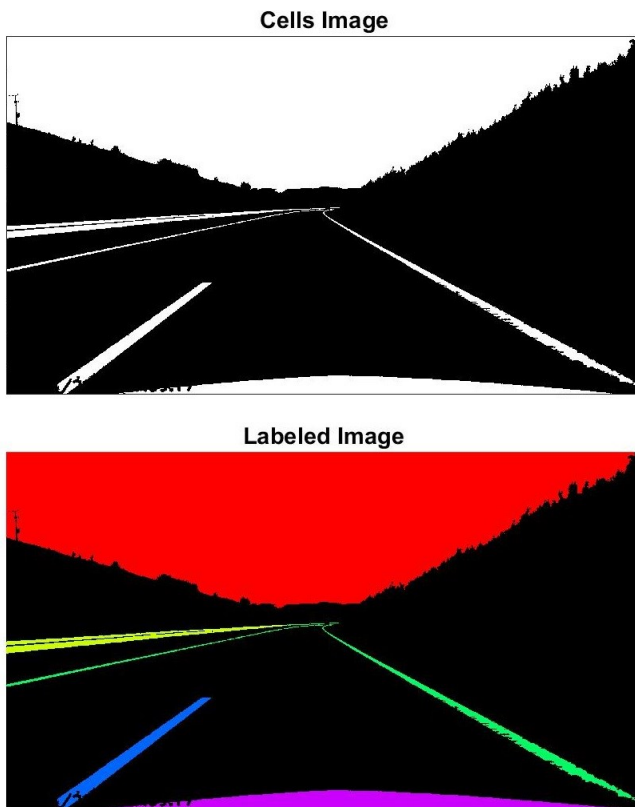


Fig. 1. Cells Image is the result of the process of raw image, while Labeled Image shows the different components of the picture

Gradient approach was used for classifying if the labeled component is a lane. Sobel algorithm was used to detect the edges, and Hough transform to detect lines on the image. Lines was filtered based on their slope and length. This can be seen on Fig. 2.

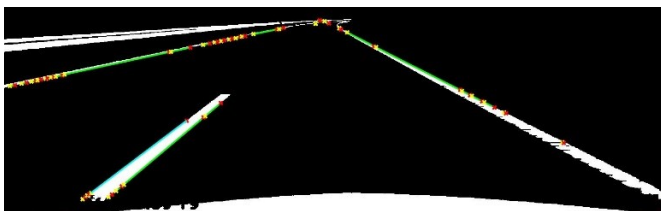


Fig. 2. The detected Hough Lines on the Cells Image

By combining the points of the Hough lines and the labeled components, ROI-s could be calculated. Averaging the white pixel's coordinates, lane points could be found as the Fig. 3 presents.

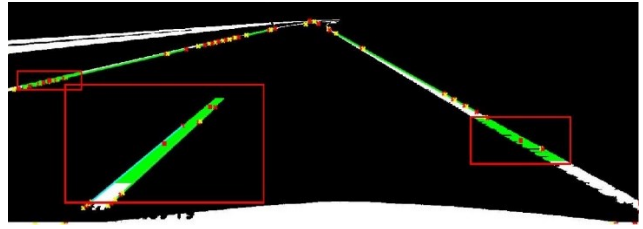


Fig. 3. The detected lanes

The combination of color and gradient algorithm was successfully applied on nearly 20.000 frames. Based on this dataset, neural network was trained for lane detection. Averaging the results of the neural network and our algorithm, better accuracy guaranteed during the further measurements. With this two method nearly 70.000 frames were processed.

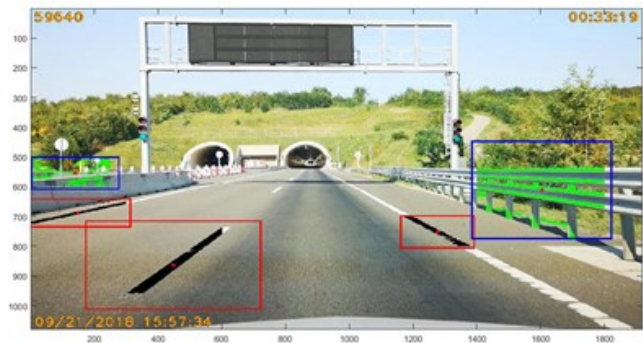


Fig. 4. Processing of the frames, red squares represents the detected lanes, blue squares shows the unimportant areas.

A homography is made in a motorway rest, and applied in order to calculate the detected lane position respect to our vehicle coordinate system. The result of the homography can be seen on Fig. 5.



Fig. 5. The result of the homography

After granting lane points in the respective of the ego-vehicle coordinate system, the lane points have to be transformed into GPS coordinates. For this step, an accurate localization of the ego-vehicle is needed. The ego-vehicle position and heading are ensured by the Kalman filter, presented in Section III. With the known ego-vehicle GPS

coordinates and lane position in vehicle coordinate system, the GPS position of the detected lanes can be calculated by using haversine formula. On Fig. 6 the map building method can be seen.

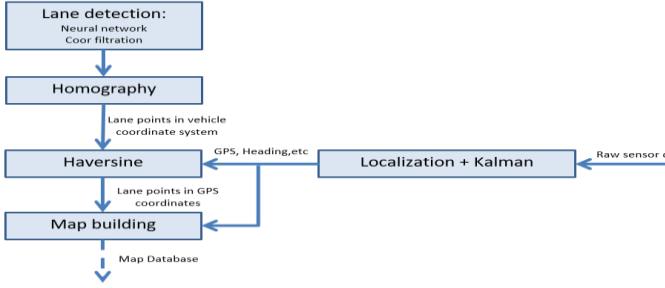


Fig. 6. Architecture of the building process

As the result of this process HD Map is generated, which is presented by Fig. 7. For storing the map [16] a matrix is used. On the left side the pre-planned route can be seen, whilst on the right side the built HD map projected on Google maps. The precision of the built map is indicated by the similarity of the two routes.

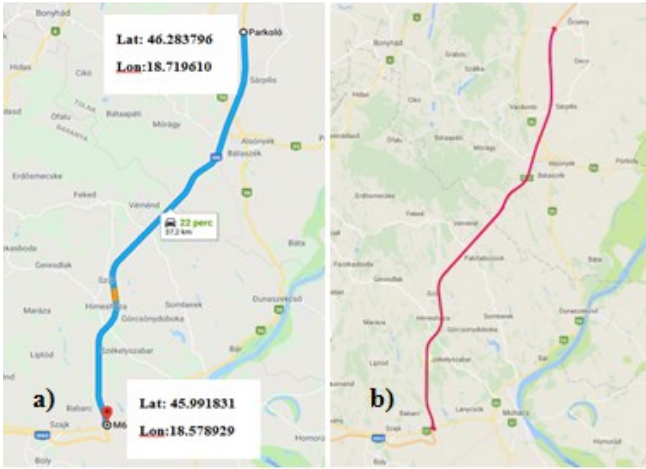


Fig. 7. a) The pre-planned route on Google maps b) The built HD map projected on Google maps

V. LOCALIZATION ON THE MAP (ONLINE PROCESS)

After building up the HD Map, in order to navigate an autonomous vehicle, localization on the prebuilt map is needed. This step is also called the online process. For this step, additional measurements were taken on the same route independently from the map building. The sensor data of the smartphone was logged again. In online process Kalman filtering is also applied to provide an accurate localization. For each frame, the closest lane points are searched for. After this step, the lane points that are contained in the prebuilt HD map are recalled from the closest ones up to 150 meters ahead. Based on the GPS coordinates and heading as the result of the Kalman filter and on the recalled line points, the line points can be transformed to the coordinate system of the ego-vehicle by using the haversine formula again. Finally, a second order polynomial is fitted on them for each lane by using least square method. Thus, each line is represented by a second order polynomial in the form, described by (17).

$$y_{j,i} = a_{2j}x_{j,i}^2 + a_{1j}x_{j,i} + a_{0j}$$

(17) Fig. 10. Built HD Map in tunnel

In (17) $y_{j,i}$ and $x_{j,i}$ are the lateral and longitudinal coordinates of i -th point of j -th line in the respective of ego-vehicle coordinate-system, a_{2j} , a_{1j} , and a_{0j} are the coefficients of the j -th line polynomial. Thus, the result of the online process is the coefficients of the polynomial of each line. Hence, the trajectory of lane keeping can be given in a simple way. On Fig. 8 the method of online localization is represented.

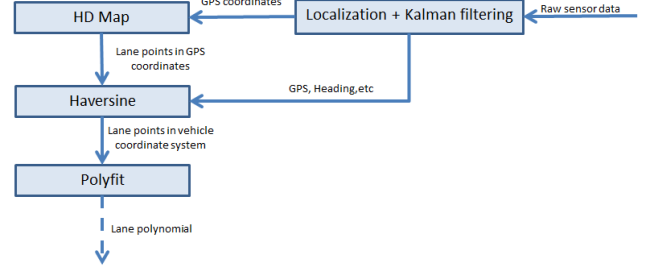


Fig. 8. Architecture of the online localization

VI. RESULTS

First of all, to visualize the result of the prebuilt HD Map, line points were projected to the satellite image of Google maps. This step is represented by the Fig. 9 and Fig. 10.



Fig. 9. Built HD Map



Fig. 10. Built HD Map in tunnel

Fig. 10 represents the result of dead reckoning based on Kalman filtering approach. The HD map was built also in tunnels and under bridges despite of lack of GPS.

In order to validate our results, we made further measurements and evaluated them. First of all, a DGPS was used for validating the localization process based on Kalman filtering. This step is needed to ensure that the localization during the building process was accurate enough to adopt the map on other smartphones with other GPS. Fig. 11 represents the result of localization comparing with the DGPS that has an accuracy of 2 centimeters.



Fig. 11. Smartphone GPS compared to DGPS signal

The lane width is also a good aspect of accuracy. It is represented by Fig. 12, that the expected value is around 3.7 m with a distribution of 10 cm. Compared to Hungarian highway regulation (3.7 m), its less than 3% difference in average.

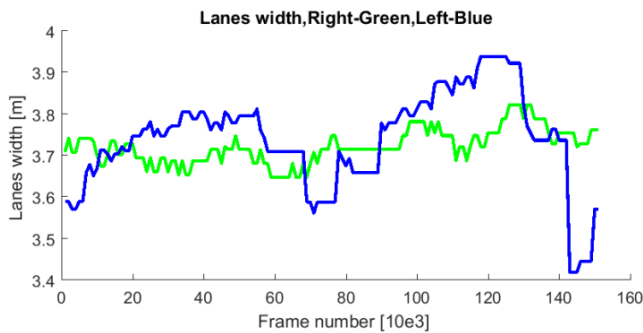


Fig. 12. Lanes width of the built map

In the next step, the result of the online process was evaluated. The forward 100 m lane data was projected on the frames of the measurement of online process. The matching features were analyzed. As Fig. 13 shows, the results are remarkable. Because of the accurate localization the building and also the recall process is accurate enough to provide high performance lane recognition. The method also works in bad weather conditions and visibility. In addition because of the

dead reckoning the lane information is also ensured in tunnels where GPS does not provide localization.

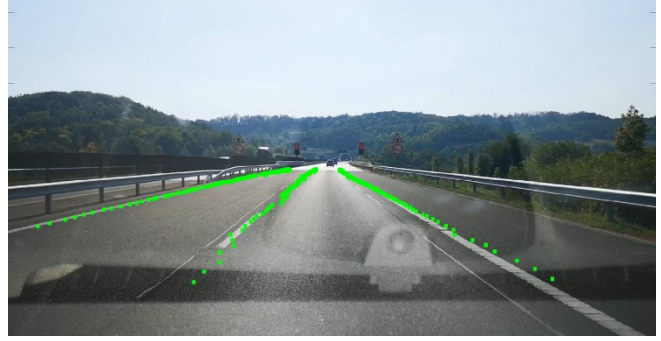


Fig. 13. Lanes projection in online process

The longitudinal accuracy can be measured by time delay of lane type changing between the video log, and the map feature. 20 opening lanes were captured during the measurement, on the right side. The average delay was measured as 5 frames. Based on the longitudinal velocity (120 km/h) and the camera frame rate (60 FPS), the longitudinal accuracy can be calculated. As the result of this calculation, the accuracy is less than 3 meters. On Fig. 14 this lane type change can be seen.

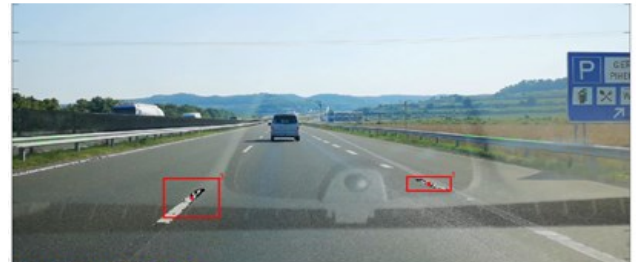


Fig. 14. Detection of lane type change

The longitudinal accuracy is acceptable in highway scenarios. However lateral accuracy is much more informative about the performance of lane recognition. By logging lane change maneuvers, lateral accuracy could be measured. While on Fig. 15 the lane projection can be seen, on Fig. 16 shows the lines, represented by polynomials, give back an accurate model of the relative position in the ego lane. Thus the lateral accuracy can be assumed good enough to navigate an autonomous vehicle.

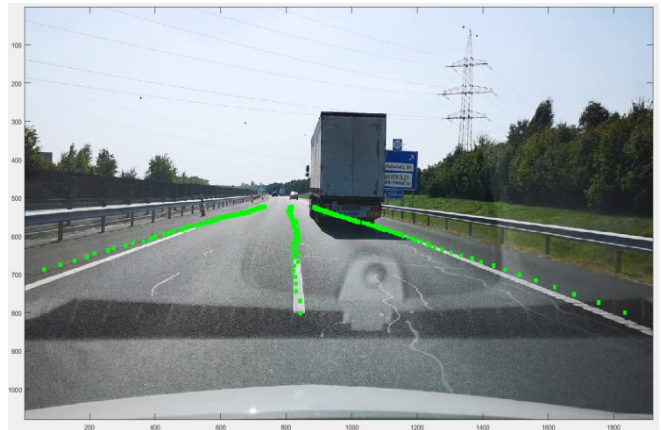


Fig. 15. Line polynomials during lane change compared with camera frame

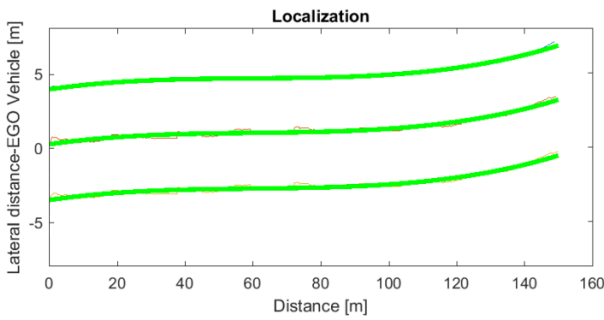


Fig. 16. Line polynomials during lane change compared with camera frame

VII. CONCLUSION

After considering the results, based on several measurements we can state that a map of the highway is generated with remarkable sub-meter accuracy. Although, numerical results scarcely can be formed, lateral accuracy can be considered high enough for lane keeping assist and lane change maneuvers. This statement can be derived by the low error ($<3\%$) of the lane width and based on the visual results of lane changing. In longitudinal direction the accuracy was calculated less than 3 meters at 120 km/h traveling speed based on line type changing.

Drawing the conclusion, based on semi-automatic validation methods, supporting autonomous vehicles by HD Map can be carried out with an onboard camera and IMU-s in the future.

VIII. ACKNOWLEDGMENT

The project has been supported by the European Union, co-financed by the European Social Fund. EFOP-3.6.2-16-2017-00002.

REFERENCES

- [1] M. A. Qudus, W. Y. Ochieng, and R. B. Noland, "Integrity of map-matching algorithms," *Transportation Research Part C: Emerging Technologies*, vol. 14, no. 4, pp. 283-302, 2006.
- [2] M. Jabbour, P. Bonnifait, and V. Cherfaoui, "Map-Matching Integrity Using Multihypothesis Road-Tracking," *Journal of Intelligent Transportation Systems*, vol. 12, no. 4, pp. 189-201, oct 2008.
- [3] C. Fouque and P. Bonnifait, "Matching raw gps measurements on a navigable map without computing a global position," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 887-898, June 2012.
- [4] M. Qudus, R. Noland, and W. Ochieng, "A High Accuracy Fuzzy Logic Based Map Matching Algorithm for Road Transport," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 10, no. 3, pp. 103-115, sep 2006.
- [5] Franck Li, Philippe Bonnifait, Javier Ibanez-Guzman2 and Clément Zinoune "Lane-Level Map-Matching With Integrity on High-Definition Maps", 2017 IEEE Intelligent Vehicles Symposium, June 11-14, 2017, Redondo Beach, CA, USA
- [6] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108-117, September 2006
- [7] J. Civera, A. J. Davison, and J. Montiel. Inverse depth parametrization for monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932-945, October 2008.
- [8] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (SLAM): Part I. *IEEE Robotics & Automation Magazine*, 13(2):99-110, June 2006.
- [9] A. J. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Patterns Analysis and Machine Intelligence*, 29(6):1052-1067, June 2007
- [10] Christian Lundquist, "Sensor Fusion for Automotive Applications", Department of Electrical Engineering Linköping University, SE-581 83 Linköping, Sweden, 2011 pp.100-127
- [11] Carola Otto: "Fusion of Data from Heterogeneous Sensors with Distributed Fields of View and Situation Evaluation for Advanced Driver Assistance Systems", KIT Scientific Publishing 2013 pp 27-33, 101-102
- [12] R. Omar Chavez-Garcia: "Multiple Sensor Fusion for Detection, Classification and Tracking of Moving Objects in Driving Environments", HAL, 2014
- [13] R. Rajamani, *Vehicle Dynamics and Control*, Ch. 2. Lateral Vehicle Dynamics, page 15-45.
- [14] E. D. Kaplan and C. J. Hegarty, *Understanding GPS*, 2006.
- [15] M. S. Grewal, V. D. Henderson, and R. S. Miyasako, "Application of Kalman filtering to the calibration and alignment of inertial navigation systems," *IEEE Transactions on Automatic Control*, vol. 36, pp. 3-13, 1991
- [16] Jie Du and M. Barth, "Next-Generation Automated Vehicle Location Systems: Positioning at the Lane Level," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 48-57, mar 2008.