Towards Autonomous Self-Assessment of Digital Maps

Oliver Hartmann\textsuperscript{1}, Michael Gabb\textsuperscript{2}, Roland Schweiger\textsuperscript{2} and Klaus Dietmayer\textsuperscript{1}

Abstract—Digital maps are becoming increasingly important for driver assistance systems: providing optimal lighting conditions in night scenarios, presenting the road geometry to the driver, or for usage in autonomous driving tasks. However, recorded digital maps own one drawback: due to road changes and inaccurate recordings, discrepancies between the map and the real world exist. Because these discrepancies can lead to severe application level failures, detection of map errors is essential to ensure overall system integrity.

This work proposes a new approach to online verification of digital maps for automotive usage. In contrast to previous work, the described system is able to detect errors in front of the vehicle. On the basis of a large database of map geometry and sensor information, a neural network is trained to classify the digital map integrity by optimally fusing different information sources depending on their strength and reliability. Although generally applicable, it is shown that a combination of orthogonal measurement principles is greatly beneficial for this decision task. A radar sensor, infra-red imagery and road geometry information estimated from visible light images are employed as input for the neural fusion. Experiments on real-world data verify the proposed concepts.

I. INTRODUCTION

A. Motivation

Several driver assistance systems use digital maps as additional sensor input. Especially, in scenarios where the visibility is not sufficient, at night time or in bad weather conditions. Then the digital map can provide more safety for the driver. For example, augmented reality applications are able to display the road in front the vehicle to the driver [1]. Also, autonomous driving tasks need detailed digital maps as a foundational component of their system [2]. Future lighting systems will use the digital map as additional information to control the headlight. Since 2003 the Mercedes Benz E-class provides a dynamic curve light system based on the steering wheel angle and the lateral acceleration. In some situations, especially in the beginning and at the end of a curve and in the transition of an s-shaped curve, the system can be optimized by using a road course estimation ahead of the vehicle, using a digital map to precisely control the headlight and provide optimal vision to the driver. In particular in such an application, unnoticed map errors may lead to severe consequences, since drivers momentarily lose their vision completely. This work uses the predictive curve light as example application.

B. Related Work

There is a major interest in constructing automated systems to update and detect changes in digital map databases. Various existing and emerging applications require up-to-date, accurate and sufficiently detailed road predictions. A very common approach is the use of data from satellites, aerial images or laser scanners as a source of automated map updating [3]–[6]. The road geometry is extracted from the images and compared to the digital map database. Pink and Stiller [6] developed a special image descriptor for pixel-level classification of aerial images. Afterwards, they extracted lane boundaries on top of the classified landmarks. Finally, the estimated lane boundaries are used to remove falsely detected lane markings from the map. Such approaches are designed to update large map databases at one time since aerial images are readily available for almost any region. But they are not suitable for safety-critical applications due to the age of the recorded images.

One method nearer the real-time capability is the use of traces produced by Global Navigation Satellite Systems (GNSS). Instead of the time consuming and logistically challenging process of driving along roads to register changes, the map provider passively collect characteristics of the roads from in-car navigation systems from their customers [7]–[9]. Ekpenyong et al. [7] use a neural network to collect the user data and decide if the map database should be updated. Even though this strategy is closer to real-time than analysing aerial images it is still not suitable for safety-critical applications.

Zinoune et al. [10], [11] developed a map verification system based on the position track of a single vehicle. Multiple error metrics are defined and used to compare the track to the map database using a statistical test. As this work is designed to reduce the time required for the detection of an error, Page’s trend test is used to perform sequential hypothesis testing for map errors based using the vehicle dynamics and estimation error covariances as input. This approach is detecting the error at the vehicle’s position. Considering the predictive curve light application, mentioned in the introduction, this is too late. The requirement for this light system is at least 25 m in front of the vehicle since the low beam light is controlled in this distance.

To solve this problem, this work presents an approach to assess digital maps in front of the vehicle autonomously while driving. Multiple forward-looking sensors are fused
to compare an image of the current environment vehicle with the digital map geometry. As we consider the predictive curve light application this work focuses on rural roads at night time.

C. Overview

To detect errors at the required distance, a sensor is utilized that gives an image of the environment in front of the vehicle. Comparing only the radar sensor with the digital map, quickly shows that in some situations the former does not supply enough information. For this reason, this work includes multiple sensors to assess the digital map geometry. Figure 1 shows an overview of the framework. Three sensors that are already available in many Mercedes-Benz models are used: a Near Infra-red (NIR) camera, a radar sensor and a RGB camera. The sensor setup is chosen as they own different attributes that are able to support each other. For example the radar sensor has a smaller azimuthal angle than the camera sensors but in contrast it gives precise information about object distances and it is robust against weather and lighting changes. Section II describes the preprocessing of the sensor data and how to extract the features with help of Regression Convolutional Neural Networks (RCNNs).

On the other hand the equivalent characteristics are calculated from the digital map. This is explained in Section III. To assess the digital map, the parameters from the sensor stack and the parameters from the digital map are fused using a neural network, as illustrated in Section IV. The result is the probability $p_e$ for an error at the required distance $d_x$. Both, the neural fusion module as well as the RCNNs are trained by a large database with a reference track created from a Real Time Kinematic (RTK) system. Finally this work is evaluated on real world data in Section V.

II. Features

A. Sensors

The first sensor is a Bosch night vision camera recording light in the near infra-red spectrum. A sample picture is shown in Figure 2a. The second sensor is a Continental ARS310 77Ghz long range radar sensor in the front of the vehicle. The third sensor is the multi purpose camera from Continental which provides an optical lane recognition. The lane model is represented as a clothoid. The lateral distance $y_L$ of the left lane marking at the distance $x$ can be approximated by:

$$y_L(x) = b - \delta + \psi x + \frac{1}{2} c_0 x^2 + \frac{1}{6} c_1 x^3.$$

The control unit of the camera computes the lane width $b$, the lateral offset $\delta$, the relative angle $\psi$ between the vehicle and the road, the road curvature $c_0$ at the vehicle position and the change of curvature $c_1$. The curvature $c$ at a certain distance $x$ can be calculated by:

$$c(x) = c_1 x + c_0. \quad (1)$$

An example of the optical lane recognition can be seen in Figure 2c.

Fig. 2: Samples of the sensor data. Figure a shows an image from the NIR camera, Figure b shows the occupancy grid built from the radar measurements and Figure c shows the optical lane recognition.
Fig. 3: Preprocessing step for the nightview camera. Figure a shows the raw data from the NIR camera and Figure b shows the result after applying the inverse perspective transform based on the camera calibration.

B. Preprocessing

The long range radar sensor has a small azimuthal angle. For this reason it is difficult to analyse the street borders in a single measurement. To obtain a meaningful statement about the road geometry an occupancy grid needs to be built. For this, the vehicle odometry is estimated by a Kalman filter that uses the vehicle wheels speeds and the data of a gyroscope as measurements. In the next step moving objects are removed from the measurement based on the Doppler effect. Additionally, the radar measurements are integrated into an occupancy grid. A detailed explanation can be found in the work from Schüle et al. [1]. See Figure 2b for an example. The image shows how the radar sensor gets reflections from obstacles like guardrails, vegetation or reflector posts. The result is a 100×100 px image with 1 px = 1 m.

To receive similar results from the NIR images the first step is an inverse perspective transform of the raw data. The image from the camera is back-projected onto the ground plane, followed by a projection to a virtual camera located above the vehicle. For fast processing a lookup table can be used. The cameras position and focal length is chosen in such way that 1 px corresponds to 0.4 m. For an example see Figure 3b. A detailed description of the inverse perspective transform can be found in [12].

Because the RCNNs require values in the interval [0, 1], all images, from the NIR camera and the radar sensor, have their mean subtracted and are normalized to the requested interval.

C. Regression Convolutional Neural Networks

The relevant features from the radar and the NIR camera images are extracted using a special neural network. The goal of this framework is getting the relevant information based on the error metric described in Section IV-B. Thus, the angle between the vehicle longitudinal axis and the road is requested from the images (see Figure 4). This part is handled by a combination of a Convolutional Neural Network (CNN) [13] with a General Regression Neural Network (GRNN) [14].

Current benchmarks show that CNNs are well suited for classification tasks [13], [15], [16]. The results from Ciresan et al. [17] even compares favourably to a human recognition rate and achieves better results. CNNs proposed by [13] extract localized features from the input images, convolving image with learned filter weights. The responses are then transformed by a non-linear activation function and subsequently sub-sampled. Some configurations repeat this layer a second time with the output from the first pooling module. Afterwards the output is passed to a vanilla feed forward network architecture with one output neuron per class.

The approach of the this work is to advance that of [18], where the road direction is predicted through a CNN with three classes: left, right and straight. However, a regression task is used to get a continuous angle rather than a discrete description of the road geometry. Thus, the proposed architecture is a specific neural network consisting of five layers, where the first layers are two convolutional layers, followed by three feed-forward neuron layers. The output layer of the feed forward network is replaced by a single neuron with a linear activation function. A detailed overview can be seen in Figure 5. This architecture is similar to the estimation of face alignment parameters in the work of Duffner [19]. The commonly used subsampling layers, which, in addition to the reduction of the feature size, provide a translation invariance, have a negative effect on this regression task. As proposed by Simard et al. [20] these layers are omitted. The stride size is increased to reduce the input size before the feed-forward module and the filter size is also increased so that no inputs are missed. The framework is trained by back propagation. Thus, the cost function is changed to the quadratic difference. The result is an angle α m either from

Fig. 4: The CNNs are trained to output the angle between the longitudinal axis of the vehicle and the digital map in 25 m. Additionally, this angle is used as error metric for the neural fusion module.
Fig. 5: The architecture from the RCNNs used to extract the features from the imaging sensors. The feature extraction is done by two consecutive convolution layers. The results are vectorized and transferred to an feed forward multi layer perceptron consisting of an input layer, a hidden layer and an output neuron.

the near infra-red camera input or the angle $\alpha_r$ from the radar sensor input.

III. DIGITAL MAP

The implemented map provider is a map database designed for automotive applications. It delivers information about the road geometry, the number of lanes, speed limits, the functional class etc. This data is provided by an industry standard interface called Advanced Driver Assistance Systems Interface Specifications (ADASIS) [21] that offers efficient transmission of the map data. The geometry information is transmitted as discrete points describing the shape of the road. In order to be able to work with these shape points, they are transformed from the WGS84 coordinate system to the UTM coordinate system. A coarse vehicle position is transmitted by the ADASIS protocol. It is used as the initial point for a precise localization.

The localization module is based on a radar environment map already described in Section II-B. The positioning is solved by matching the local environment map with the digital map geometry provided by the ADASIS protocol. For details see [1].

A. Digital Map Feature Extraction

To be able to compare the features from the sensor part with the digital map, the corresponding parameters from the map geometry need to be extracted. The first parameter is the angle $\alpha_n$ from the NIR RCNN or the angle $\alpha_r$ from the radar RCNN. This can be done analog to Figure 4. The next parameters are the curvature $c_0$ at the vehicle position and the curvature variation $c_1$ from the optical lane recognition. To get these parameters from the digital map, the curvatures and longitudinal distances are formed:

$$ T_i = (c_i, x_i). $$

To determine the coefficients $c_0$ and $c_1$ from (1), a least-squares fit to the linear function is performed for the set of tuples $T$ (see Figure 6). The coefficients $\psi$ and $\delta$ can read directly from the localization module. As this value are sourced from the digital map, they are indexed by $m$.

IV. NEURAL FUSION

Up to this point, different features from different sensors of varying reliability have been retained. Common approaches, like Majority Vote, Decision Templates, Naive Bayes Combination or Probabilistic Approximations, either statistically determine the strength of these features or do not consider their reliability. In contrast, in this work a trainable neural fusion approach is chosen which is capable of learning the strength of every feature. Additionally, it can build up complex relations between the input data, which are hard to model in common approaches.

A. Architecture

The neural fusion module is a feed-forward neural network with an architecture seen in Figure 7. It is composed of an input layer, a hidden layer and an output layer. The number of input neurons in the input layer equals the size of the input vector. The hidden layer consists of 100 neurons with a hyperbolic tangent activation function. A softmax function is used for the output, so every class is represented by its posterior probability. Two classes are defined: an error in the digital map and a correct digital map. The input vector $v_f$ for the input layer consists of the features from the sensor data $v_s$:

$$ v_s = (\alpha_r, \alpha_n, c_0, c_1, \psi, \delta)^T $$

$$ v_f = (\alpha_r, \alpha_n, c_0, c_1, \psi, \delta)^T $$

Fig. 6: Clothoid approximation from a digital map. The curvature $c$ in every shape point is calculated. Afterwards a least square fit for (1) is applied to the tuples of distance and curvature.
In the present work, the digital map assessment is an important module for the light application. For this reason it is useful to adapt the error metric to this system. Thus, the error $e$ is defined by:

$$e = \left| \arctan \left( \frac{y_m}{d_x} \right) - \arctan \left( \frac{y_r}{d_x} \right) \right|$$

where $d_x$ is the distance to observe in front of the vehicle and $y_m$ is the lateral offset in this distance based on the digital map (see Figure 4). $y_r$ is similar with the reference track, in this case $d = 25$ m. To get the label $l$, a hard threshold needs to be defined whether the error $e$ is acceptable for the application or not:

$$l = \begin{cases} 
0, & e \leq 4^\circ \\
1, & e > 4^\circ 
\end{cases}$$

The error function and the threshold are chosen appropriately for the supported system. In this case the values are chosen for a predictive curve light system that controls the low beam in a distance of 25 m. A difference of $4^\circ$ is the maximum acceptable difference to the reference. The error metric and the thresholds are learned implicitly by the neural network. For other applications the error metric and thresholds need to be adapted and the neural fusion module needs to be trained again.

V. EXPERIMENTS

Both neural network parts, the RCNNs and the neural fusion module, are trained in supervised mode. The neurons in the Multi Layer Perceptron (MLP) as well as the filter kernel in the RCNNs are initialized randomly. They are trained through stochastic gradient descent by minimizing the error $E$ from the outputs compared to the labelled data. LeCun [13] showed that stochastic online learning is superior to batch mode as it is faster and results in a better solution. Since it is hardly possible to implement the in parallel, a compromise between both modes is chosen. In this work the weights are updated after a small batch of 64 samples.

A. Sensor Decision

The features from both image sensors, the radar and the NIR camera, are extracted by a RCNN framework. The general architecture for both tasks is equal: two convolution layers, and one MLP with an input layer, a hidden layer and an output layer. To achieve the best results from the RCNNs, a grid search over different parameter sets was performed. Different numbers of filters, filter sizes, stride sizes and the number of hidden neurons were evaluated. In consequence, some parameters do not differ for both input databases, the NIR camera images and the radar images. The best results were achieved with 16 filters and a stride size of 5 px in both convolution layers. To prevent missing any inputs the filter size is 9 px in the first convolution layer and 5 px in the second convolution layer. In contrast, the number of neurons in the hidden layer is different for both databases: the NIR task is more demanding concerning the capacity of the net. So the radar architecture uses only 20 neurons in the hidden layer.
Fig. 8: An example part of a sequence out of 22 test sequences. The label $l$ is marked in orange. The prediction $p_e$ for an error in the digital map from the neural fusion module is plotted in blue.

layer and the NIR needs 200 hidden neurons to achieve the best results. The best result using the radar database with 19k training and 9k test samples is a mean error of 3.21° between the output and the label. When it comes to the NIR data with 32.7k training and 15.3k test samples, the mean error on the best parameter set is 4.79°. The database sizes differ due to the different cycle rates of the sensors.

B. Neural Fusion

The architecture of the neural fusion module is depicted in Figure 7. Two modes of building the input vector (see IV) are considered: append mode and the distance mode. So the number of input neurons is 12 in the append mode and 6 in the distance mode. Different numbers of hidden neurons (10, 20, 50, 100, 200, 400) were evaluated for the neural fusion module. In the end, the best results were achieved by using 100 hidden neurons for both modes. The result is the probability for an error $p_e$ and the probability for no error $p_c$. Thus, it is:

$$p_e + p_c = 1.$$ 

Formally, two class problems are mapped to a positive and a negative class label. In this case an error $e$ is labelled as the positive case. Normally the decision for one class is done by choosing the class with the highest probability. In a two class problem, an error is chosen if $p_e > t$ and $t = 0.5$. The neural fusion module was trained with 72k training and 37k test samples. These samples are extracted from recordings of rural roads in night time. The complete database has a length of 3.5h.

The map database is provided by Navteq with the last update in the year 2010. Though the recordings are already four years old the ratio between correct map samples and map errors is unbalanced. Due to the low occurrence of digital map errors in the database, artificial errors are created to adjust the ratio between correct and incorrect digital map geometry in the database. Four different types of manual errors are implemented: roundabouts, lane changes, gaussian distributions and intersections. See Figure 9 for examples. Every error is defined by different parameters like width, length etc. These parameters are set randomly over the database.

Regarding the accuracy the append mode is slightly better with 96.64 % than the distance mode with 95.70 %. Varying the threshold $t$ from 0 to 1 creates a ROC graph, which can be seen in Figure 10. The append mode clearly outperforms the distance mode when the FP rate is under 0.055. For higher FP rates the distance mode is slightly better. Figure 8 shows a plot of a test sequence. In the background the time intervals for an error are marked in orange. The prediction of the neural fusion module, the probability $p_e$, is plotted in blue. Note that both, the label and the prediction plots, are positioned 25 m in front of the vehicle. Thus, our system not only detects errors with great reliability, but also is able to assess the map quality upfront, before dangerous situations occur.

VI. Conclusion

This paper presented a novel approach for an online system to assess digital maps regarding the current environment of the vehicle. In contrast to common systems mentioned in Section I-B, this work is able to detect errors 25 m in front of the vehicle, primarily to support a map-based curve light application. A neural network fused features from a NIR

![Fig. 8: An example part of a sequence out of 22 test sequences. The label l is marked in orange. The prediction p_e for an error in the digital map from the neural fusion module is plotted in blue.](image-url)
camera, a radar sensor and an optical lane recognition. An accuracy value of 96.64\% was attained, which is superior to previous approacher since this system not only detects map errors with great reliability. It does so in an upfront manner, detecting early and before they lead to severe application level failures.

For future work it is planned to extend the framework with a temporal filter after the neural fusion module to increase the performance of the single frame classifier. Also additional features can increase the accuracy and can easily be adopted and included in the system.

**REFERENCES**


