Generation of High Precision Digital Maps using Circular Arc Splines

Andreas Schindler, Georg Maier and Florian Janda

Abstract— **Digital maps can provide essential information for many advanced driver assistance systems (ADAS) dedicated to both safety and comfort applications. As the level of detail and global accuracy of state-of-the-art digital maps are not sufficient for a multitude of applications, we present methods and models for the generation of high precision maps. The proposed modeling includes 3D lane level information, road markings, landmarks and additional attributes with benefits for many ADAS. The extensive use of circular arc splines enables both adjustable accuracy and high efficiency as our cartographic methodology guarantees the minimum number of curve segments with respect to a given error threshold.**

I. Introduction

Many driver assistance systems benefit from digital maps as a static source of information on the local environment of a vehicle. In contrast to the maps that are available in today's navigation systems, we focus in this work on highly detailed and precise maps which are required for active safety applications. Advanced lane keeping and navigation systems depend on reliable lane level information including data on the lane curvature and the lane width. In particular, for the situation analysis and the risk estimation the association of dynamic road users to individual lanes plays a decisive role. Knowledge of the drivable direction and a 3D representation of lanes supports assistance systems in areas of dynamically changing elevation. Finally, for map based self-localization purposes the digital map provides the landmarks that are associated with data from the vehicle's environment perception in order to estimate the vehicle's position and orientation. As the requirements analysis of the research project Ko-PER [1] shows, all of the above mentioned data must be available with high global precision. An accuracy of a few centimeters is desirable.

Almost all mentioned applications have in common that distances between points and representations of continuous structures have to be calculated. An example is the association of vehicles to individual lanes, which is referred to the map matching. It requires the computation of the distance between the vehicle's position and the lane candidates. Hence the best-approximating points with respect to the lane candidates have to be calculated. The same requirements arise from the situation analysis and the map based self-localization. Obviously, a representation of continuous structures in form of dense point sequences is not suitable: The amount of data for storage and for the processing on the application side would be immense. Furthermore, determining the best-approximating points would be computationally expensive. Instead, a suitable curve representation is preferable. From a technical point of view, the above mentioned constraints lead, amongst others, to the following criteria concerning the modeling of the digital map:

- Continuous structures like lanes or road markings should be represented as curves instead of point clouds.
- Point to curve distances must be computable in a highly efficient way. This efficiency is determined by the number of segment candidates that must be considered and the complexity of distance calculation between a point and a single segment.
- The global accuracy of the map has to be guaranteed throughout the generation process.
- An efficient data representation is necessary for both storing and processing map information.
- Due to the real time requirements of active safety applications, fast access to map elements within a given region of interest is needed.

II. Related Work - Lane Representation

Analogous to [2] and [3] in many countries a turn of a typical road in rural areas consists, at least constructionally, of a sequence of a straight line, a clothoid, a circular arc, a clothoid and again a straight line segment. Clothoids are used to provide a smooth steering phase when passing the lane sections since their curvature changes linearly. However, clothoids show certain disadvantages: Curve approximation, point to curve distance calculation and even drawing of the curve are computationally expensive as they imply non-linear optimizations. In general, the offset curve of a clothoid is not a clothoid anymore. Thus, it is reasonable to use an alternative curve model.

The most simple way of representing lanes are open polygons, which are often used in digital maps (cf. [4], [5], [6]). In [7] a method for the generation of polygonal maps based on aerial images is proposed. For this purpose, exact geo-referenced undistorted images of high resolution are needed, which is often not available in practice. Furthermore, polygons do not comply with the reality since they cannot describe curved structures in a suitable way. For a precise approximation of these structures, a relatively high number of line segments is needed, which has two negative effects: On the one hand, the amount of data for storage and processing increases. And on the other hand, the association of points to their closest curve segment is hampered as the number of candidate segments is high. Hence the calculation of the point to curve distances is not suited to satisfy the real time requirements defined by the applications. Moreover,

All authors are with FORWISS, the Institute for Software Systems in Technical Applications, University of Passau, 94032 Passau, Germany {schindler, gmaier, jandaf}@forwiss.uni-passau.de

polygons do not provide any curvature information which is indeed important for many applications. There are several works proposing cubic Hermit splines [8], B-Splines [9] or NURBS [10] as an approximation of clothoids in order to have a preferably precise curvature information. While some of the polynomial splines guarantee G^2-continuity, i.e. curvature continuity, the calculation of point to curve distances still remains a non-linear optimization problem. Furthermore, the invariance criterion with respect to offset building is not satisfied and its approximation is comparatively expensive to compute.

In contrast, arc splines, i.e. curves composed of circular arcs and line segments, have many advantages compared to the curve systems presented above: They provide exact offset and arc length computation, invariance regarding rotation, translation and scaling and also enable a simple closest point computation (cf. [11]). The mentioned calculation can even be done in a closed form. Arc splines can be expressed in a parameter-free description. In addition, they are compatible with all established geometry and CAD systems. Above all, smoothness is required in order to enable a realistic modeling. In this context smoothness necessarily means G^1-continuity, i.e. tangent continuity. Hence we consider arc splines, where the tangent unit vectors of each two successive segments are equal at the corresponding breakpoint (see Fig. 1). One can show that a smooth arc spline with n segments can be stored with only $2n + 3$ floating-point numbers. In comparison, an open polygon with the same breakpoints needs $2n + 2$ floating-point numbers. Note that, in general, the higher flexibility of arc spline leads to a significantly lower number of segments when approximating a point sequence.

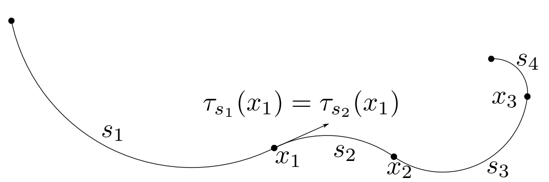

Fig. 1. Smooth arc spline γ with four segments $s_1, \ldots, s_4$ and breakpoints x_1, x_2, x_3. The equality of the tangent unit vectors $\tau_{s_1}(x_1)$ and $\tau_{s_2}(x_1)$ indicates the tangential smoothness at x_1.

A detailed evaluation of the particular curve systems mentioned above regarding precision, calculation of best approximating points, offsetting and resulting data volume can be found in [12]. This paper should be seen as an overview of the methods and models we use for the generation of highly precise maps.

III. Modeling

Our map model contains both geometric and semantic information on different relevant structures of the environment. Currently, this includes landmarks, individual road markings and lanes.

Landmarks are fixed objects with small spatial expansion, that can be robustly detected under various environmental conditions. Examples are traffic signs, reflection posts or trees. They serve as a reference for the map based self-localization. In particular, traffic signs play an important role for the situation analysis. The landmarks are stored as points with additional attributes like height and width in the map.

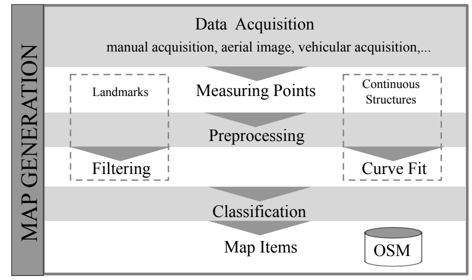

Fig. 2. Overview of the map generation process.

Continuous structures like road markings and lanes are represented as smooth arc splines. The individual segments of dotted road markings are modeled as line segments, which are indeed arc splines with only one segment. The road markings contain information like their width, color and type. A lane is represented by an arc spline as well, which is defined by the centerline of the real lane. Furthermore, a lane contains supplementary data like the elevation profile and the course of the lane width. Again, these profiles are modeled as arc splines. Hence from the resulting smooth 3D curve representation the elevation and thus the inclination can be derived.

To achieve greatest possible compatibility and interchangeability the approved OpenStreetMap (OSM) database and file scheme is used for data storage. Thereby new elements and attributes can be added easily to the map. New tags have been introduced to enable the storage of arc splines in OSM ways.

IV. Generation

This section deals with the generation of the modeled map elements. Fig. 2 shows an overview of the processing stages. In general, many different data sources can be considered as input for the map. Their common denominator is the possibility of extracting measuring points from the data for the individual map elements, like road markings, lanes or landmarks.

A. Data Sources

Beside the manual topographical survey, this can be achieved by processing orthographic aerial images including elevation maps if available. In our case, we obtain the data using a vehicular perception system. The vehicle is equipped with a real time kinematic (RTK) unit that provides highly accurate estimations of the vehicle's dynamics and pose up to 0.02 m. Having a well-calibrated set of sensors including a monocular gray value camera, laser scanners and a positioning unit, the data acquisition in the desired road area can be performed.

The identification of landmarks is done by an appropriate processing of the laser scanners' data. 3D positions and dimension information of landmarks are output. Subsequently, video based classification techniques like a traffic sign recognition can remove false candidates and helps to specify e.g. the type of a landmark.

In [13], we proposed a video based method for detecting and classifying painted arrow markings on the road surface. It is reasonable to integrate these arrows into the digital map since they are useful for the situation analysis indicating the allowed driving directions. Furthermore, the arrow markings serve as visual landmarks for self-localization purposes.

Continuous structures like road markings or individual lanes are extracted using the visual lane recognition system based on the one presented in [14]. Since points for the lane representation cannot be observed directly, they are determined by the centerline between left and right road markings. In order to enable an accurate global reconstruction of the environment, the vehicle's motion is compensated using the RTK unit. Finally, all extracted measuring points for continuous structures and the landmark candidates are transformed in a global Cartesian coordinate system like UTM or Gauss-Krüger.

B. Segmentation

As mentioned in Section III, the curve representations are initially calculated in a tangential plane of the earth. Afterwards, the corresponding elevation profile is computed, which leads to a 3D curve described in Section V. Hence at first we consider the projected set of measuring points $M \subset \mathbb{R}^2$. Since no order of the points in M for continuous structures is supposed, a segmentation of the point cloud is applied in order to identify the individual structural parts.

Therefore, a Euklidean neighborhood graph $G = (V, E)$ is generated, where the coordinates of the nodes $v \in V$ are identified with the points in M. With d denoting the Euklidean distance, the set of edges E in G is defined as $\{(u, v) \in V \times V | 0 < d(u, v) \leq \delta\}$. The neighborhood distance δ is determined by the density of the measuring points and it is smaller than the minimum distance between two adjacent road marking segments. Let $\mathfrak{C}$ be the set of connected components of G. The set of corresponding points of a connected component $C \in \mathfrak{C}$ is denoted by P_C. Depending on the dimension of C either a line segment or a smooth arc spline is fit to P_C.

C. Dotted Road Markings

For dotted road markings, each $C \in \mathfrak{C}$ is fit with the best-approximating line l, such that $\sum_{p \in P_C} \mathrm{dist}(l, p)^2$ is minimal, where $\mathrm{dist}(l, p) := \min_{x \in l} \ d(x, p)$. The corresponding line segment is the minimal connected subset of l including all projections of P_C.

Probably, outliers appear in the recorded data set due to the physical measuring process. Hence a robust method is needed for their detection. Experiments have shown that using line fitting with RANSAC [15] is suitable to remove most of the outliers successfully. If the lengths of the segments are known by preliminary knowledge, the segments can be used as a start solution for an optimization process with respect to the pose of the line segment. However, this postprocessing should be handled carefully, as the painted road markings in reality often differ from the construction plan.

D. Preprocessing of the Arc Spline Fit

Some preprocessing steps are necessary in order to apply the arc spline approximation described in [11]. A so called *tolerance channel* describes the feasible area for the resulting smooth arc spline. For the construction of the channel an ordered and filtered list of points based on the input data is required. To calculate an order of the data points, for large connected components $C \in \mathfrak{C}$ a starting point $s \in P_C$ is selected. The Dijkstra algorithm is used to compute the distances to all points of P_C. The resulting set of points is sorted in ascending order according to their graph theoretical distance to s.

In order to approximate the ordered points by an open polygon, line segments are fitted iteratively. A new segment is started if the line fitting error exceeds a given threshold. However, this leads to a high computational effort. Thus, a divide and conquer algorithm is proposed solving the problem in logarithmic time: Let l be the best approximating line of P_C. The perpendicular bisector of the corresponding line segment of l recursively divides P into two subsets. This procedure is continued until one of the following criteria is not satisfied with respect to preset thresholds anymore:

- the line fitting error (least squares or maximum error) is small
- the number of points per line segment is small
- the line segment is short

Again, RANSAC is used for outlier removal during the line fitting. The result of the bisection process is a sequence of line segments $(l_i)_{1 \leq i \leq n}$. Therefore, the ordered list given by the starting and end points of the line segments l_i can be used for the generation of the tolerance channel.

E. Tolerance Channel and SMAP

As now an ordered list of filtered and denoised input points P is given, it is reasonable to approximate these data up to a user-specified tolerance. For our purposes, a tolerance value between $0.1\mathrm{m}$ and $0.2\mathrm{m}$ turned out to be practicable since a higher accuracy cannot be guaranteed owing to the errors made during the physical acquisition process. The proposed method controls the approximation error by a geometric model: Only solutions staying inside the tolerance channel are taken into account. The width of this channel represents the user-specified maximum tolerance, which can vary locally as well. The canonical shape of a tolerance channel modeling a maximum error ε is given by the set of points which have an Euklidean distance to the open polygon running through P of at most ε. The resulting boundary curve, which is in fact an arc spline, is approximated by a simple closed polygon. In order to ensure that the polygonal offset curve is not further away from the input polygon than ε, we suggest to insert *offset spikes* at

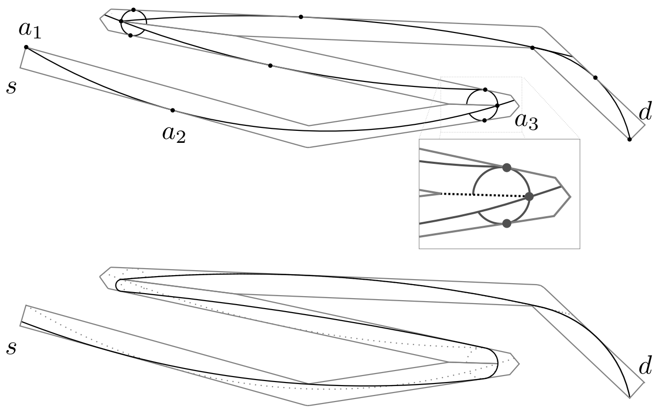

Fig. 3. Visualization of the SMAP algorithm; Forward (top) and backward step (bottom). In the magnification an offset spike is drawn as dotted line.

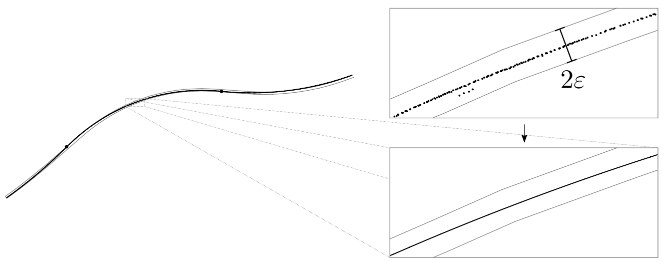

Fig. 4. Illustration of the arc spline approximation for a lane section: Input data points including outliers, ε-tolerance channel together and resulting smooth minimum arc path (SMAP).

sharp corners. This way we obtain geometric constraints, which can be adjusted in a comfortable and intuitive manner. In addition, two disjoint edges of the tolerance polygon, s and d, are fixed and act as start and destination of the channel (see Fig. 3).

We are interested in a curve that approximates not only the extracted points up to unavoidable fitting errors but also describes them effectively, i.e. with minimal complexity. Such a characterization allows coping with tasks and applications motivated in Section I. Any smooth, i.e. G^1-continuous, arc spline staying inside the tolerance channel and connecting s and d with a minimum number of segments solves the problem. Such a curve is called *Smooth Minimum Arc Path (SMAP)*. Note that the breakpoints are not required to be part of P but they are to be determined automatically. This has considerably positive effects on the resulting number of segments. In contrast, all other currently-known methods using arc splines have no theoretical bounds concerning the number of segments (e.g. [16]).

In [11] an efficient algorithm for generating a SMAP is presented. The proposed method is a two step greedy algorithm traversing the tolerance channel K from s to d in a forward step and back again from d to s in a backward step: In a forward step, so-called *windows* are constructed successively. These are arcs inside K having three *alternating restrictions*. Alternating restrictions are points where

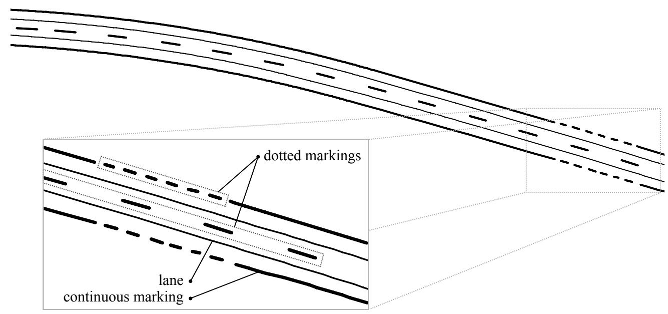

Fig. 5. Reconstructed road with all road markings (thick) and lanes (thin).

an arc is alternately touched by the boundary from the left and from the right. In Fig. 3 an example is illustrated with restriction points a_1, a_2, a_3 (left – right – left). While the first window simply has to start at s, the second and the following windows ω_i have to satisfy a certain continuation condition. Either they touch ω_{i-1} and have two alternating restrictions or they have three alternates. The procedure is stopped when d has been reached. Though the windows do not represent a smooth arc spline, they can be used to compute a SMAP: In a backward step the lastly calculated window ω_k represents the last segment of the resulting SMAP. In particular, the minimum segment number is k. The predecessor segments are then determined by touching their successor and by two alternating restrictions. The whole procedure is finished when s has been reached. In the second row of Fig. 3 the backward step of the SMAP algorithm is visualized as well. Fig. 4 shows the reconstruction result of a real lane section.

Though generating a SMAP guarantees a smooth arc spline with the least possible number of segments with respect to any given accuracy, the approximation algorithm doesn't satisfy real time requirements. The best known implementation of this algorithmic approach has a quadratic worst case complexity regarding the number of input points. However, the computational time doesn't play an important role since the map is generated offline. In fact, it is more important to provide a preferably small number of curve segments. This way, the closest point computations in the online applications can be solved in an efficient manner.

V. 3D Profile

Once an arc spline representation γ of a lane with arc length $l > 0$ is available, the corresponding elevation profile can be computed. For this purpose, let t_i be the run length parameter of the input data points p_i with respect to their projection on γ. The elevation information h_i of the measuring points, which is in fact its Z-coordinate, is used in order to create the input data (t_i, h_i) for the elevation profile. Next, the SMAP algorithm is applied to these points resulting in a smooth arc spline η. In case of realistic inclinations, for every run length $t \in [0, l]$ we obtain exactly one height value. Thus, we have a function $h : [0, l] \to \mathbb{R}$ describing the elevation profile with respect to the arc length l of γ.

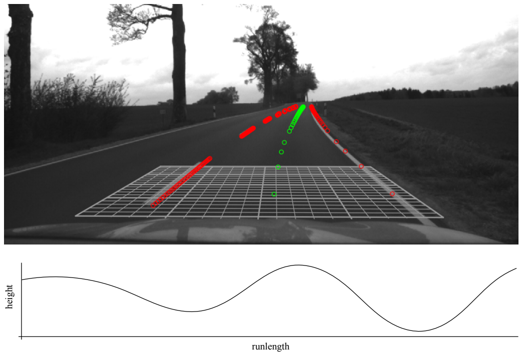

Fig. 6. Projection of a lane with corresponding road markings and the elevation profile.

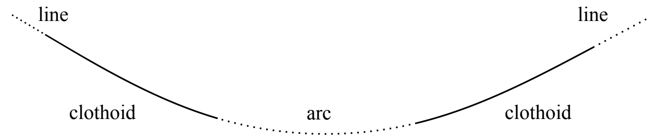

Fig. 7. A road consisting of: line segment – clothoid segment – arc segment – clothoid segment – line segment

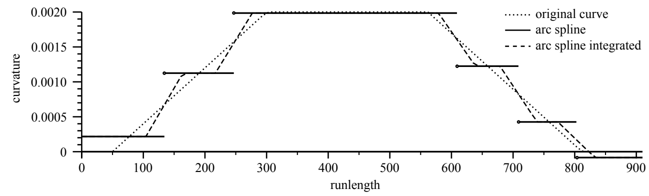

Fig. 8. Curvature of the original curve κ_{ref}, the arc spline curvature κ and the linearized curvature $\tilde{\kappa}$.

Denoting by $g : [0, l] \to \mathbb{R}^2$ the arc length parametrization of γ, the smooth function

$$f : [0, l] \to \mathbb{R}^3,\ t \mapsto \begin{pmatrix} g(t) \\ h(t) \end{pmatrix},$$

represents the 3D course of a lane. It is not hard to prove that the length of f is equal to the length of η. Therefore, distances along the 3D curve can be computed efficiently as η is composed of a small number of circular arcs and line segments. For reconstructing the course of the road width we can proceed analogously. Fig. 5 and 6 show a section of the generated map and the corresponding elevation profile.

VI. Applications and Map Access

In order to satisfy the real time requirements, it is important to gain access efficiently to the stored map data. A common task is to provide map information within a region of interest around a vehicle's position. For fast access to the stored data, caching strategies and index structures like quad-trees, R-trees or a Voronoi decomposition are useful.

In addition to the applications mentioned in the introduction, we present some more examples for the usage of the generated map in the following.

A. Curvature

Especially for autonomous driving applications a precise reconstruction of the original curvature is essential. In particular, the linear curvature characteristics of the clothoid parts should be approximated in a preferably exact manner. However, the curvature of an arc spline is a step function and therefore not continuous. Though there are approaches for G^2-smoothing of arc splines, we consider the following method which results in a piecewise linear curvature characteristic: Let γ be a SMAP with arc length parametrization $g : [0, l] \to \mathbb{R}^2$ and run length knots $t_0 := 0 < t_1 < \cdots < t_m := l$. Hence the corresponding curvature function is

$$\kappa : [0, l] \to \mathbb{R},\ \kappa = \sum_{j=1}^{m} \kappa_j \mathbf{1}_{[t_{j-1}, t_j[},$$

where $\mathbf{1}_{[x,y[}$ is the characteristic function of the interval $[x, y[$ and κ_j are the curvature values of the particular segments. We then define an approximation $\tilde{\kappa} : [0, l] \to \mathbb{R}$ by

$$\tilde{\kappa}(t) := \frac{1}{2\Delta t} \int_{t-\Delta t}^{t+\Delta t} \kappa(s) ds,$$

where $\Delta t > 0$ is a user-specified parameter controlling how strong the influence of the curvature values of the adjacent segments are. For arc splines $\tilde{\kappa}$ is a sum, which can be calculated very efficiently.

In Fig. 7 a synthetic example of a lane course is shown. It is composed of a line segment with length 50m, a clothoid segment of length 250m, an arc segment with radius 500m and length of approximately 260m, again a clothoid segment with 250m and a line segment of 100m.

Fig. 8 shows the original curvature κ_{ref}, the curvature of the corresponding SMAP κ for a tolerance of $\varepsilon = 0.1\text{m}$ and the approximated curvature $\tilde{\kappa}$ for $\Delta t = 30$. The resulting number of segments is six. We obtain the error measures

$$E := l^{-1} \int_0^l |\kappa(t) - \kappa_{\text{ref}}(t)|\, dt = 13.9 \cdot 10^{-5},$$

$$\tilde{E} := l^{-1} \int_0^l |\tilde{\kappa}(t) - \kappa_{\text{ref}}(t)|\, dt = 7.33 \cdot 10^{-5},$$

where l is the arc length of the SMAP. This relates to a relative error of roughly 4.1% for the clothoid and arc parts. In comparison to that, a SMAP for $\varepsilon = 0.02\text{m}$ needs nine segments and yields errors $E = 8.05 \cdot 10^{-5}$ and $\tilde{E} = 1.36 \cdot 10^{-5}$ which means a relative error of about 0.57%. Since this example yields promising results, a verification on large data sets will be done in our future work.

B. Map based Self-Localization

The precisely reconstructed landmarks and road markings in the map enables map based self-localization approaches in order to provide an accurate estimation of a vehicle's pose in situations when GPS signals are not available or erroneous.

On hilly grounds the measurements of the vehicle's local perception differ from map data due to the inclination of the road. However, the elevation profile of lanes allows the correction of these measurements and thus to cope with this problem.

C. Lane Level Information

Due to the individual lane representations in the map, it is now possible to provide detailed information on the vehicle's surrounding. This includes

- the most likely lane a vehicle is driving on,
- the current number of lanes of the road,
- the number of lanes in the vehicle's direction,
- the number of lanes in the opposite direction,
- the comparison of lane indices in order to check whether two vehicles are driving on the same lane,
- the distance between a stopping line and a vehicle approaching to a intersection,
- the prediction of a vehicle's position on a lane in a certain distance.

Furthermore, the elevation profile of a lane allows the detection of cambers, which, moreover, enables assistance systems to provide occlusion warnings.

To extract this information new topological levels can be added to the map, which provide the relationships between the individual lanes.

However, all of the above mentioned data can also be calculated online as distances and intersections of points, lines and arcs can be computed very efficiently.

VII. Conclusion and Future Work

We have presented methods and models for the generation of high precision digital maps providing important information for many different ADAS dedicated to safety applications. The advantageous properties of the arc spline model enable fast information processing for the applications. In addition, our approach generates an efficient data representation as the least possible number of curve segments is guaranteed.

On behalf of the Ko-PER project, a map was generated for selected road sections with $\varepsilon = 0.1\mathrm{m}$. Its accuracy has been validated independently by a third-party using a high precision reference system: Owing to measuring errors during the acquisition process, an accuracy of $0.18\mathrm{m}$ in average has been assessed at randomly chosen reference points. Fig. 9 illustrates a map section with classified landmarks and an aerial image.

In our future work we are going to develop a map based self-localization strategy and extend the automatization of the map generation process.

VIII. Acknowledgments

This work results from the joint project Ko-PER, which is part of the project initiative Ko-FAS, and has been funded by the German Bundesministerium für Wirtschaft und Technologie (Federal Department of Commerce and Technology) under grant number 19 S 9022E.

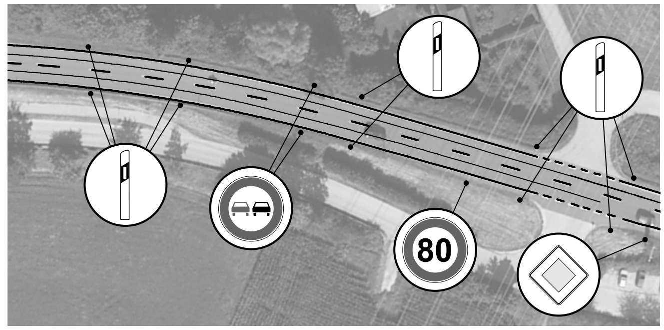

Fig. 9. An aerial image by the Bayerische Vermessungsverwaltung is used as overlay to verify the reconstruction result.

We would like to thank BMW Group Research and Technology (by name A. Rauch, K. Vogel and F. Klanner) and SICK AG (by name R. Krzikalla) for the cooperation in this project and for providing the vehicular platform and sensor data of various test scenarios.

References

[1] "Ko-FAS," http://www.ko-fas.de.

[2] Bundesministerium für Verkehr, "Richtlinien für die Markierung von Straßen," 1993.

[3] *Straßen- und Tiefbau*. Teubner, 2008.

[4] N. Mattern, R. Schubert, and G. Wanielik, "High-accurate vehicle localization using digital maps and coherency images," in *IEEE Intelligent Vehicles Symposium (IV)*, 2010, pp. 462–469.

[5] K. Gerlach and M. Meyer zu Horste, "A precise digital map for GALILEO-based train positioning systems," in *9th International Conference on Intelligent Transport Systems Telecommunications (ITST)*, 2009, pp. 343–347.

[6] U. Noyer, J. Schomerus, H. Mosebach, J. Gacnik, C. Loper, and K. Lemmer, "Generating high precision maps for advanced guidance support," in *IEEE Intelligent Vehicles Symposium*, 2008, pp. 871–876.

[7] O. Pink and C. Stiller, "Automated map generation from aerial images for precise vehicle localization," in *13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2010, pp. 1517–1522.

[8] A. Chen, A. Ramanandan, and J. A. Farrell, "High-precision lane-level road map building for vehicle navigation," in *IEEE/ION Position Location and Navigation Symposium (PLANS)*, 2010, pp. 1035–1042.

[9] G. Koutaki, K. Uchimura, and Z. Hu, "Network Active Shape Model for Updating Road Map from Aerial Images," in *IEEE Intelligent Vehicles Symposium*, 2006, pp. 325–330.

[10] L. Z. Wang, K. T. Miura, E. Nakamae, T. Yamamoto, and T. J. Wang, "An approximation approach of the clothoid curve defined in the interval [0, [pi]/2] and its offset by free-form curves," *Computer-Aided Design*, vol. 33, no. 14, pp. 1049–1058, 2001.

[11] G. Maier, *Smooth Minimum Arc Paths. Contour Approximation with Smooth Arc Splines*. Shaker, 2010.

[12] A. Schindler, G. Maier, and S. Pangerl, "Exploiting arc splines for digital maps," in *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 1–6.

[13] G. Maier, S. Pangerl, and A. Schindler, "Real-time detection and classification of arrow markings using curve-based prototype fitting," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 442–447.

[14] T. Tatschke, *Methoden und Modelle der frühen Sensordatenfusion zur Umgebungserfassung für Fahrerassistenzsysteme*. Shaker, 2011.

[15] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography." *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[16] H. Weigel, H. Cramer, G. Wanielik, A. Polychronopoulos, and A. Saroldi, "Accurate road geometry estimation for a safe speed application," in *IEEE Intelligent Vehicles Symposium*, 2006, pp. 516–521.