2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Automatic 3D Road Surface Reconstruction via Cross-Section Modeling and Interpolation*

Matteo Bellusci and Matteo Matteucci

***Abstract*— Accurate 3D road surfaces are important for the development of detailed and realistic scenarios to validate autonomous driving algorithms. In these scenarios, simulations can be conducted, for instance, to evaluate the response of a safety system under dangerous conditions. In this paper, we propose an approach designed to automatically generate 3D road surfaces from data collected by a vehicle equipped with various sensors, including a LiDAR. These road surfaces are meant to be both accurate and realistic for driving simulations. The proposed approach, after deriving the clothoidal representation of the surface borders, pursues the idea of extracting and interpolating a set of smooth 3D cross-section profiles. The resulting surface provides a 3D representation in analytical form, allowing detailed rendering at the desired resolution. We experimentally evaluate the proposed approach in a real-world scenario to assess its performance in terms of accuracy, scalability, and computing time.**

I. INTRODUCTION

Nowadays, autonomous driving systems are receiving increasing attention from the scientific community and industry. Advanced Driver Assistance Systems (ADAS) and autonomous vehicles are intended to provide, increasingly over time, a higher and higher level of safety for drivers and passengers, being at the same time key elements of future urban mobility. To achieve high levels of reliability, automotive testing procedures are used to subject vehicles and systems to a series of evaluations designed to verify certain safety properties. These procedures are generally conducted first in simulation, then in laboratories, and eventually in the real world. Virtual tests, which are conducted through simulations, commonly require the design of highly realistic navigation environments, i.e., HD maps.

Although it is possible to design such maps manually, this is a very time-consuming process [1], especially when a high level of detail has to be achieved. For instance, this is the case when it is necessary to accurately simulate the contact between the vehicle wheels and the road surface. Automatic 3D road surface modeling can be accomplished by leveraging the data coming from various sensors. Related work mainly involves the use of LiDAR [2], [3], [4], [5] or cameras [6], [7], [8], [9], but the use of other sensors is possible, e.g., infrared-based sensors [10], [11].

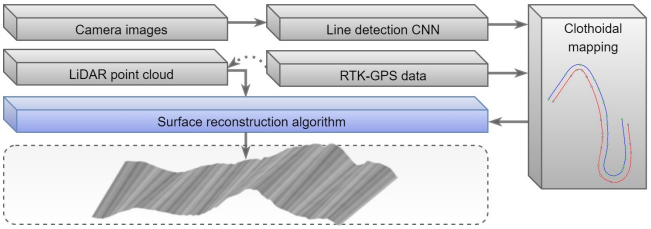


Fig. 1. High-level and simplified overview of the pipeline for reconstructing the road surface from the required sensor data inputs.

A camera-based approach that extracts multiple vanishing points to estimate the slope of the road surface is presented in [12]. Alternative image-based approaches rely on aerial images [13]. Sometimes, the 3D road surface reconstruction relies on the use and integration of data from several sensors [14], [15], combining the potentials of the various data sources, including 3D LiDARs.

Common LiDAR-based approaches for 3D road surface reconstruction, in most cases, retrieve a model in the form of a mesh [2], [4] or a rectangular grid with elevation information [5]. A recent work allows dynamic resolution of the drivable area [15], but typical mesh-based and grid-based approaches do not allow on-demand rendering precision unless the tiles are recomputed. LiDAR-based methods also allow for the calculation of 3D cross-section profiles of the road [16], [17], e.g., to estimate the road slope.

In recent work, a pipeline was presented to map, from vision and precise GPS sensors, the line markings of a road in the form of 2D splines of clothoids [18], ensuring high accuracy in the definition of the borders delimiting the area navigable by the vehicle. In this paper, we propose to go a step further and provide a pipeline designed to derive an analytical form of the whole 3D road surface by exploiting also LiDAR data. An overall high-level overview of the newly proposed pipeline is illustrated in Fig. 1. The analytical form we propose allows virtually infinite rendering resolution. Indeed, differently from mesh and grid-based approaches, the analytical form describing the road surface can be rendered, without any change, at any level of detail. In addition, unlike meshes, our smoother representation: keeps the borders as clothoids, obviates the irregularities caused by triangles, and requires much less disk space for storage, thus resulting in a more scalable and practical solution in this respect.

*This paper was supported by "Sustainable Mobility Center (Centro Nazionale per la Mobilità Sostenibile – CNMS)" project funded by the European Union NextGenerationEU program within the PNRR, Mission 4 Component 2 Investment 1.4. Also, this work was supported by AnteMotion S.r.l., Trento, Italy. Any opinions, findings, conclusions, or recommendations, either expressed or implied, in this material are those of the authors and do not necessarily reflect the views of the sponsoring organizations.

All authors are with the Department of Electronics, Information, and Bioengineering (DEIB), Politecnico di Milano, Milano, Italy, name.surname@polimi.it

The innovative approach presented in this paper extracts an accurate 3D surface of the road via cross-section modeling and interpolation leveraging data from a set of sensors composed of cameras, LiDAR, and RTK-GPS. In Section II, we show how to obtain a clothoidal mapping of road markings. We then present our surface extraction pipeline in Section III, and in Section IV, we evaluate it experimentally in a real-world scenario to assess its performance in terms of accuracy, scalability, and computing time.

II. ROAD MARKINGS CLOTHOIDAL MAPPING

Our goal is to identify and extract the road surface by exploiting data acquisitions obtained by a vehicle equipped with a set of sensors, including cameras, LiDAR, and RTK-GPS. While the location of road lines is information typically obtained from vision, LiDAR data are better suited for accurately identifying the morphological and roughness details of the traveled road. The proposed procedure exploits the potential of both sensors to operate effectively. The idea is to start from the image stream provided by the cameras to, in the first place, identify road lines and map them (also using RTK-GPS data) in the global 2D reference system; intuitively, the outer (road) borders thus obtained define the drivable area. Then, exploiting LiDAR point cloud data, an extraction algorithm, explained in detail in Section III, derives the 3D profile of the surface by leveraging the elevation of the points belonging to the drivable area.

Mapping road lines is not a trivial step; our system relies on a pipeline that is responsible for identifying the lines in the camera images, extracting a set of points for each frame to then be projected into the absolute 2D reference system, and finally defining the edges as curves in the plane. In the considered context, given a road edge, the curve in the plane identifying it is mathematically represented by a set of clothoids (also known as Euler spirals, Cornu spirals, or spiros). Clothoids are curves whose curvature varies linearly along their length and are proven to be suitable for a better road design [19], [18]. Their adoption is generally related to their ability to increase passenger comfort [20], and they are currently supported by the most common standards for road network specification (e.g., the ASAM OpenDRIVE® standard). They also represent a generalization for different types of curves, including arcs (constant curvature) and straight lines (zero curvature), allowing for more comprehensive mapping capability. We now show how to obtain such curves, starting from the sensor system output.

A. Road Line Markings Mapping

Given the stream of images provided by the vehicle's cameras, the road line markings can be extracted through line detection algorithms. While traditional methods leverage color information or vision cues through image processing [21], [22], [23], recent approaches rely on deep learning [24], [25], [26]. The latter approaches employ the use of Convolutional Neural Networks (CNNs) and represent nowadays a promising resource [24]. Nevertheless, both types of approaches are compatible with our pipeline. In our work, we considered the line detector from [27], which is based on U-Net [28]. Given a front camera image, it generates a mask highlighting the pixels that belong to road lines. Note that, when the vision sensor system is composed of stereo cameras (or a set of multiple cameras), image stitching can be used to produce a single frontal view of the scene [29].

Fig. 2. Example of output resulting from the road line markings extraction pipeline: on the left, the points reprojected (using the homography matrix) onto the image plane; on the right, the same points in BEV space.

The next step is to obtain two sets of approximately equidistant line points, one for each border of the drivable area. For this purpose, we employ an Inverse Perspective Mapping (IPM) [30] to obtain, for each frame, a Bird's-Eye View (BEV), i.e., a rectified view of the scene. In addition to removing perspective effects [31], IPM is useful in that it maps image (and lines mask) pixels into a relative planar coordinate system. We consider the standard IPM model consisting of a 3×3 homography projection matrix with 8 degrees of freedom (please refer to [32] for details). The homography matrix defines a mapping between two planes (frontal view and rectified view), as well as conveying and enclosing information about the intrinsic and extrinsic parameters of the vehicle's camera. Thus, a point (image or mask pixel) in homogeneous coordinates can be projected into the relative 2D reference system using matrix multiplication. Note that in order to achieve an accurate mapping, it is advisable to have an extrinsic camera calibration system to compensate for vehicle motion [33], [34], in addition to proper intrinsic camera calibration that is generally related to manufacturing.

Line masks provide, once the pixels belonging to the lines have been mapped in the BEV space, coarse and noisy information. To filter such information, it is possible to employ line tracking methods like the window-based line following (WLF) algorithm [27], where movable windows, starting from the bottom of the BEV image, follow the curvature of the lines' shapes upwards, collecting points along the lines at a constant distance. The resulting output is composed, in BEV space, of two sets of roughly equidistant line points, which can be eventually reprojected back into the frontal camera view; an example is shown in Fig. 2. To switch from the relative to the absolute reference system, it is sufficient to leverage the vehicle's RTK-GPS position at the time of image acquisition. By doing so, we can accumulate all the detected road line points in a global planar map, i.e., the absolute 2D reference system.

B. Clothoidal Mapping

Once the lane markings are mapped in the 2D global world reference system by exploiting RTK-GPS data, the pipeline

shown in [18] can be used to extract two sets of clothoids. In particular, the line points are first divided and adjusted, using the DBSCAN clustering algorithm [35], in blocks, where they are fitted with a cubic smoothing spline (one for each block). The midpoint between two consecutive splines identifies the reference knot between them; the set of knots allows the set of clothoids to be uniquely identified. Indeed, given two consecutive knots, the corresponding clothoid can be uniquely derived by G^1 interpolation [36], guaranteeing G^1 continuity. Finally, a graph-based clothoid optimization and a greedy clothoid pruning step could be optionally employed to obtain a smoother overall output while maintaining a very low continuity, derivability, and fitting error (please refer to [18] for details). The proposed procedure results in a lower fitting error (i.e., a higher accuracy) and a smaller number of clothoids at the price of a very low continuity and derivability error; such error is generally negligible and bounded.

III. SURFACE EXTRACTION ALGORITHM

We now discuss our pipeline to extract the road surface based on clothoidal mapping information. Without lack of generality, we assume the road consists of a single lane; nevertheless, the proposed methods can be extended to cover the multi-lane road case as well. Concerning the notation of this manuscript, the subscript $k = -1$ is related to the left line, while $k = 1$ is related to the right line. Intuitively, when a variable related to a line is referred with index k (with $k \in \{-1, 1\}$), the corresponding variable related to the opposite line is referred with index $-k$. We denote the sets of clothoids representing the shape of the borders with $\mathbf{C}_k$ and $\mathbf{C}_{-k}$, one for each line. A set of clothoids contains all the clothoids that make up the corresponding border clothoid spline. $\mathbf{C}_k$ ($\forall k$) is composed of N_k clothoids, i.e., $|\mathbf{C}_k| = N_k$. A clothoid $C_{i,k} \in \mathbf{C}_k$ is defined by its initial point $(x^0_{i,k}, y^0_{i,k})$ (defined on the ground plane), its initial heading $\theta^0_{i,k}$, its initial and final curvatures $\kappa^0_{i,k}$ and $\kappa^F_{i,k}$, and its length $L_{i,k}$. A clothoid $C_{i,k} \in \mathbf{C}_k$ is thus defined as a parametric curve ($s \in [0, L_{i,k}]$):

$$x_{i,k}(s) = x^0_{i,k} + \int_0^s \cos\left(\frac{1}{2}\kappa'_{i,k}\tau^2 + \kappa^0_{i,k}\tau + \theta^0_{i,k}\right) \mathrm{d}\tau \quad (1)$$

$$y_{i,k}(s) = y^0_{i,k} + \int_0^s \sin\left(\frac{1}{2}\kappa'_{i,k}\tau^2 + \kappa^0_{i,k}\tau + \theta^0_{i,k}\right) \mathrm{d}\tau \quad (2)$$

where s is the curvilinear abscissa and $\kappa'_{i,k} = \frac{\kappa^F_{i,k} - \kappa^0_{i,k}}{L_{i,k}}$. Note that, for each value of s, the curve orientation can be computed as ($s \in [0, L_{i,k}]$):

$$\theta_{i,k}(s) = \frac{1}{2}\kappa'_{i,k}s^2 + \kappa^0_{i,k}s + \theta^0_{i,k} \quad (3)$$

while the clothoid curvature is defined as ($s \in [0, L_{i,k}]$):

$$\kappa_{i,k}(s) = \frac{\partial\theta_{i,k}(s)}{\partial s} = \kappa'_{i,k}s + \kappa^0_{i,k}. \quad (4)$$

As the clothoids in $\mathbf{C}_k$ ($\forall k$) describe a road border profile, when considered collectively as a single curve, they form a global well-defined curve that is continuous and differentiable at any point. Specifically, when merging two adjacent clothoids $C_{i,k}$ and $C_{i+1,k}$, $\forall i, k$, the resulting curve is continuous and differentiable at any point. We refer to the just mentioned global curve as $C^*_k = C^*_k(s)$, defined for $s \in [0, L^*_k]$, where $L^*_k = \sum_i L_{i,k}$, and to its orientation as $\theta^*_k = \theta^*_k(s)$ (with $s \in [0, L^*_k]$). Intuitively, if k is the index of the reference line, then the reference line is represented by C^*_k.

A. 2D Road Cross-Section Profiles

The underlying idea of our method is to define a series of road cross-section profiles. To accomplish this task, we sample $N > 1$ reference line points equidistant along the s-coordinate of the reference line. We define these point coordinates from $\mathbf{S}_k = \{s_0, ..., s_{N-1}\}$, where $s_i = i \cdot \frac{L^*_k}{N-1}$ and k is the index of the reference line, obtaining the following samples: $\{C^*_k(s_0), ..., C^*_k(s_{N-1})\}$. Each sample represents the starting point of a road cross-section profile. To determine the length of a cross-section profile π_i, with $i \in \{0, ..., N-1\}$, we intersect the ray $\hat{\pi}_i$ growing in the direction perpendicular to the reference line at point $C^*_k(s_i)$ with C^*_{-k}. The ray, in parametric form, is defined as:

$$\hat{\pi}_i(t) = C^*_k(s_i) + \eta_k(\theta^*_k(s_i)) \cdot t \quad (5)$$

where $t \in [0, +\infty)$ is the parameter and $\eta_k(\cdot)$ indicates the growing direction:

$$\eta_k(u) = \left(\cos\left(u + \frac{k\pi}{2}\right), \sin\left(u + \frac{k\pi}{2}\right)\right). \quad (6)$$

Without any optimization, computing N times the intersection between a ray and a list of clothoids can be time-consuming; therefore, we apply the following verification step. Consider a clothoid $C_{j,-k} \in \mathbf{C}_{-k}$ and a ray's starting point $C^*_k(s_i)$, if:

$$||C^*_k(s_i) - C_{j,-k}(0)||^2_2 > (L_{j,-k} + L_{\text{max}})^2 \quad (7)$$

where L_{max} is the (theoretical) maximum road width, then there is no need to compute the intersection between $C_{j,-k}$ and the ray $\hat{\pi}_i$ because, if it exists, the intersection is too far away from $C^*_k(s_i)$ to be used to compute the cross-section profile's length (i.e., the local road width).

Let $\mathbf{\Omega}_i = \{(t, s)_j\}$ be the set of intersections between ray $\hat{\pi}_i(t)$ (with $t \in [0, +\infty)$) and $C^*_{-k}(s)$ (with $s \in [0, L^*_{-k}]$), then $\forall i \in \{0, ..., N-1\}$, we assume it exists s'_i such that:

$$(t'_i, s'_i) = \underset{(t,s)\in\mathbf{\Omega}_i}{\operatorname{argmin}}\, t \quad (8)$$

where $C^*_{-k}(s'_i)$ represents the projection of $C^*_k(s_i)$ onto C^*_{-k}. We assume that $s'_i > s'_{i-1}$, $\forall i \in \{1, ..., N-1\}$. A cross-section profile π_i, with $i \in \{0, ..., N-1\}$, is defined with a new parametric form ($t \in [0, 1]$):

$$\pi_i(t) = C^*_k(s_i)(1-t) + C^*_{-k}(s'_i)t \quad (9)$$

where its (planar) length is $l_i = ||C^*_k(s_i) - C^*_{-k}(s'_i)||_2$, which represents an estimate of the local road width. Intuitively, the segment π_i is the portion of the ray $\hat{\pi}_i$ that is inside the area limited by the lane's borders.

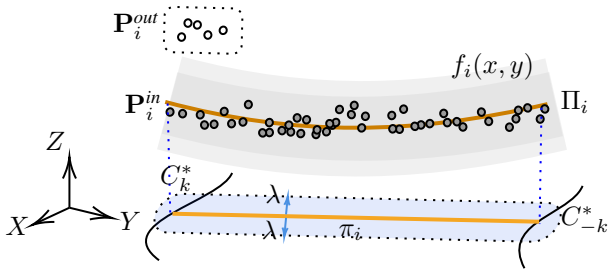


Fig. 3. The light gray surface, represented by $f_i(x, y)$, is the result of the local fitting, using the LS algorithm, of the points in $\mathbf{P}_i^{in}$. These are those points that, when projected onto the 2D plane, lie within the light blue area delimited by π_i and λ, and that are not classified as outliers.

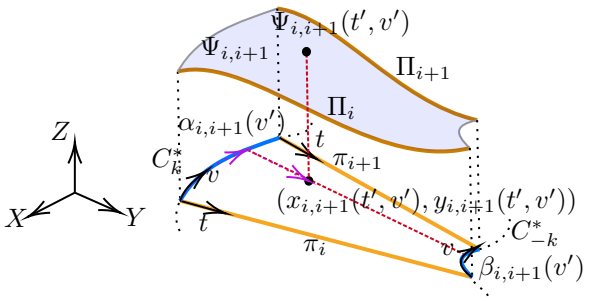


Fig. 4. Representation of the road surface section $\Psi_{i,i+1}$ between the cross-section profiles Π_i and Π_{i+1}, with k as reference line index. A point from the domain, relative to (t', v'), and its elevation are shown.

B. 3D Road Cross-Section Profiles

The segment $\pi_i(t)$, with $i \in \{0, ..., N-1\}$ and $t \in [0, 1]$, defines one of the N planar road cross-section profiles. We now aim to assign a 3D shape to $\pi_i(t)$, $\forall i$. We first define $\mathbf{P}_i \subseteq \mathbf{P}$ as the points in the point cloud that, when projected to the plane $z = 0$, lie in the surroundings of π_i:

$$\mathbf{P}_i = \{(x_j, y_j, z_j) \in \mathbf{P} \mid \min_t ||(x_j, y_j) - \pi_i(t)||_2 < \lambda\} \tag{10}$$

where $\lambda > 0$ is a reasonably small number. Note that the shortest distance $\min_t ||(x_j, y_j) - \pi_i(t)||_2$ can be computed in $O(1)$ time, $\forall i, j$. We also perform a further step to reduce possible noise in the point cloud data by removing points from $\mathbf{P}_i$ that potentially do not belong to the ground surface. Assuming no obstacles are present on the road and that the point elevations follow a normal distribution, we apply the method of removing outliers using Z-score [37]. As is commonly done, the mean and standard deviation of the target distribution are estimated from the sample (of size $|\mathbf{P}_i|$) using the sample mean and standard deviation, respectively. Thus, the cleaned set $\mathbf{P}_i^{in}$, with outliers removed, is defined as:

$$\mathbf{P}_i^{in} = \left\{ (x_j, y_j, z_j) \in \mathbf{P}_i \;\middle|\; \frac{|z_j - \mu(\mathbf{Z}_i)|}{\sigma(\mathbf{Z}_i)} < \delta \right\} \tag{11}$$

where $\mathbf{Z}_i$ is the vector of all z-values of points in $\mathbf{P}_i$, $\delta > 0$ represents a positive threshold, $\mu(\cdot)$ and $\sigma(\cdot)$ are the mean and standard deviation functions, respectively. Intuitively, the set of outliers is defined as $\mathbf{P}_i^{out} = \mathbf{P}_i - \mathbf{P}_i^{in}$, where $\mathbf{P}_i^{in}$ is called the set of inliers and is assumed to have a cardinality of M_i.

The Least-Squares (LS) algorithm is used to determine the elevation profile of each road cross-section profile. In particular, given π_i and $\mathbf{P}_i^{in}$, we fit a polynomial $f_i(x, y)$ of degree q. For simplicity, we consider $q = 2$ in the following notation, but the formulas can be trivially extended to the $q > 2$ case. Using the LS closed-form solution, we obtain $f_i(x, y) = w_{0,i} + w_{1,i}x + w_{2,i}y + w_{3,i}x^2 + w_{4,i}xy + w_{5,i}y^2$:

$$\mathbf{w}_i = (\mathbf{\Theta}_i^T \mathbf{\Theta}_i)^{-1} \mathbf{\Theta}_i^T \mathbf{Z}_i \tag{12}$$

where $\mathbf{w}_i = \begin{bmatrix} w_{0,i} & \dots & w_{5,i} \end{bmatrix}^T$ and

$$\mathbf{\Theta}_i = \begin{bmatrix} 1 & x_{1,i} & y_{1,i} & x_{1,i}^2 & x_{1,i}y_{1,i} & y_{1,i}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{M_i,i} & y_{M_i,i} & x_{M_i,i}^2 & x_{M_i,i}y_{M_i,i} & y_{M_i,i}^2 \end{bmatrix}. \tag{13}$$

The 3D road cross-section profile Π_i is then represented by the following parametric form (t is the parameter):

$$\Pi_i(t) = \begin{cases} z_i(t) = f_i(x_i(t), y_i(t)) \\ (x_i(t), y_i(t)) = C_k^*(s_i)(1-t) + C_{-k}^*(s_i')t \\ t \in [0, 1] \end{cases}. \tag{14}$$

The derivation of Π_i is illustrated in Fig. 3.

C. Road Surface Sections

The road surface section between two consecutive 3D road cross-section profiles, Π_i and Π_{i+1}, is referred to as $\Psi_{i,i+1}$. Such road surface section $\Psi_{i,i+1}(t, v) = (x_{i,i+1}(t, v), y_{i,i+1}(t, v), z_{i,i+1}(t, v))$ is defined for $(t, v) \in [0, 1] \times [0, 1]$, where the parameter t allows us to move from the reference line (with index k) to the opposite line (with index $-k$), while the parameter v is used to move forward along consecutive road cross-section profiles (with index i and $i + 1$, respectively), as shown in Fig. 4.

Given values for t, v, and i, the corresponding elevation $z_{i,i+1}(t, v)$ can be computed as a linear combination of the road cross-section profiles' elevations along the v-axis:

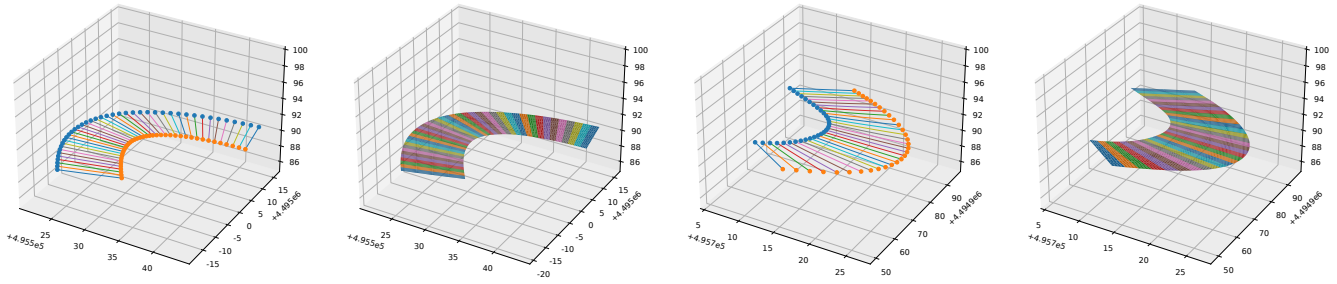
$$z_{i,i+1}(t, v) = z_i(t)(1 - v) + z_{i+1}(t)v. \tag{15}$$

To have a smoother profile, $z_{i,i+1}(t, v)$ can also be computed via cubic (i.e., 3rd order) Hermite interpolation [38] as:

$$z_i(t)h_{00}(v) + z_{i+1}(t)h_{01}(v) + m_i(t)h_{10}(v) + m_{i+1}(t)h_{11}(v) \tag{16}$$

where $m_i(t)$ represents the chosen tangent for i at t along the v-axis, e.g., $m_i(t) = \frac{z_{i+1}(t) - z_{i-1}(t)}{2}$, and $h_{kh}(\cdot)$ are the Hermite basis functions.

Along the v-axis, the presence of clothoids requires appropriate arrangements to maintain a drivable area bounded by these curves. Our approach allows bent roads to be modeled while preserving their curvatures. To achieve this, we define $\alpha_{i,i+1}(v)$ and $\beta_{i,i+1}(v)$ as the planar projections of the



(a) Cross-sections of the first curve (b) Surface of the first curve (c) Cross-sections of the second curve (d) Surface of the second curve

Fig. 5. Example of qualitative results related to two different curved sections of the circuit. Images (a) and (c) show the 3D cross-sections, of the first and second curves, respectively, calculated by our method, while (b) and (d) show the corresponding surfaces at the end of the pipeline execution.

point $(x_{i,i+1}(t,v), y_{i,i+1}(t,v))$ onto the reference line and the opposite line, respectively. Hence:

$$\alpha_{i,i+1}(v) = C^*_k(s_i) + \\ + \left(\int_{s_i}^{\gamma_{i,i+1}(v)} \cos(\theta^*_k(\tau))d\tau, \int_{s_i}^{\gamma_{i,i+1}(v)} \sin(\theta^*_k(\tau))d\tau \right), \tag{17}$$

$$\beta_{i,i+1}(v) = C^*_{-k}(s'_i) + \\ + \left(\int_{s'_i}^{\gamma'_{i,i+1}(v)} \cos(\theta^*_{-k}(\tau))d\tau, \int_{s'_i}^{\gamma'_{i,i+1}(v)} \sin(\theta^*_{-k}(\tau))d\tau \right), \tag{18}$$

where $\gamma_{i,i+1}(v) = s_i(1-v) + s_{i+1}v$ and $\gamma'_{i,i+1}(v) = s'_i(1-v) + s'_{i+1}v$. Given t, v, and i values, we are now able to retrieve the associated point projected onto the ground plane:

$$(x_{i,i+1}(t,v), y_{i,i+1}(t,v)) = \alpha_{i,i+1}(v)(1-t) + \beta_{i,i+1}(v)t. \tag{19}$$

An intuitive view of all variables is shown in Fig. 4.

IV. EXPERIMENTS

All the experiments have been conducted on a computer equipped with an AMD Ryzen™ 5 1500X quad-core processor at 3.79 GHz and an NVIDIA® GeForce® GTX 1050 GPU. Our algorithms have been written in Python and they employ the clothoid-related MATLAB library based on [36]. To evaluate the performance of the proposed method, we used real data coming from an acquisition that took place in Battipaglia, Italy, specifically at the Sele Race Circuit Track; the considered portion of the circuit (outer loop for car races) has a length of about 1.7km. The experimental vehicle was equipped with a SITECO Road-Scanner C mobile mapping system, including a FLIR® LadyBug®5 spherical camera system. The mapping system features a FARO® Lidar3D; the algorithm was fed with a point cloud representing the entire circuit. As the map point cloud is very dense, we downsampled it maintaining a total of over 11.1M points. We compare and showcase the benefits of our reconstruction over standard analytical and mesh-based ones.

To determine the coordinates of road line points, the original pipeline shown in [27], courtesy of the authors, was employed. The line mask extraction CNN, based on U-Net [28], was trained using the BDD100k dataset [39],

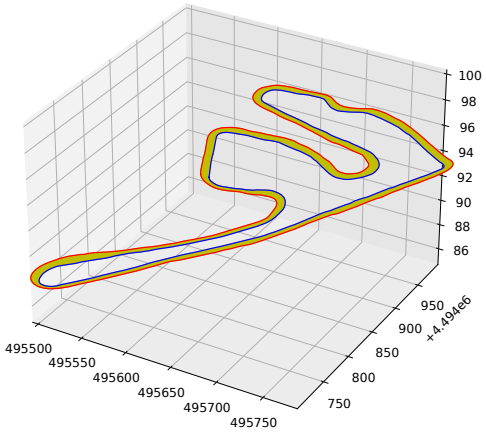


Fig. 6. Full Sele Race circuit track reconstruction.

TABLE I

FITTING ERRORS, AVERAGED OVER THE NUMBER OF CROSS-SECTIONS, AND TIME REQUIRED TO COMPUTE THE ELEVATIONS.

	$q=2$	$q=3$	$q=4$
Average fitting RMSE [mm]	4.508	4.486	**4.485**
Average fitting MAE [mm]	3.561	3.543	**3.542**
Total required time [s]	**28.6**	29.1	29.2

and the window-based line following (WLF) algorithm was used to extract the line point positions in BEV space (please refer to [27] for details). We fed their algorithm with input images obtained by performing image stitching to merge the front images from the two most relevant lenses. Leveraging RTK-GPS data, line points have been projected onto a planar global map. Then, we used the clothoidal mapping technique illustrated in [18] to derive the borders' clothoid splines of the circuit lane (please refer to Section II-B for details). Each border was described using 69 clothoids.

For the reconstruction, we sampled $N = 1500$ reference line points so as to have a distance between them (along the reference line) of slightly more than a meter (about 1.1m); the procedure to find them and corresponding rays $\hat{\pi}_i$ ($\forall i \in \{0, ..., N-1\}$) took less than 7s. Instead, it took 109s to

find the intersections between these rays and the inner border clothoids. Nevertheless, without the proposed optimization (i.e., the verification step), we measured that it would have taken about 10 times longer. The time required to compute the elevation of the cross-sections, i.e., the parameters of Π_i, $\forall i$, having the LS fitting algorithm set to operate with degree of $q \in \{2, 3, 4\}$, are reported in Table I. Computationally speaking, all the aforementioned steps took less than 3 minutes in total.

Table I reports also the average (over the N cross-sections) of the LS fitting errors. To measure accuracy errors, including LS fitting errors, we consider the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE). The results indicate that using a slightly higher q requires slightly more computation time but may allow achieving better accuracy, although as q varies (for reasonably low values of q) performance changes minimally in this case. Indeed, regardless of the value of $q \in \{2, 3, 4\}$, on average the fitting error relative to $f_i(\cdot)$ $(\forall i)$ remained below half a centimeter. Qualitative results are shown in Fig. 5, which highlights a couple of critical corner sections, and a reconstruction of the entire circuit is shown in Fig. 6.

To measure the overall average error along the entire circuit, a metric must be defined. In fact, since the surface model is defined in analytical form, the error does not depend on a resolution as in the case of classical surfaces obtained through a mesh or regular grid. Therefore, to evaluate our method, we propose two new different metrics. The first one needs to define an abstract cross-section $\Pi^{abstr}_{i,i+1}(t) = (\pi^{abstr}_{i,i+1}(t), z^{abstr}_{i,i+1}(t))$ $(t \in [0, 1])$ between every two consecutive cross-sections Π_i and Π_{i+1} as $\Pi^{abstr}_{i,i+1}(t) = \Psi_{i,i+1}(t, \frac{1}{2})$. Then, the points $\mathbf{P}^{abstr,\ in}_{i,i+1}$ of the point cloud in its surroundings (excluding outliers) are found and fitted using the LS algorithm, obtaining a different elevation $z^{abstr,\ LS}_{i,i+1}(t)$ (with respect to $z^{abstr}_{i,i+1}(t)$) for $\Pi^{abstr}_{i,i+1}(t)$. To compute the difference between the two elevation profiles, a natural way is to compute the difference between them as $\sqrt{\int_0^1 \left(z^{abstr,\ LS}_{i,i+1}(t) - z^{abstr}_{i,i+1}(t)\right)^2 \mathrm{d}t}$. A variant of this metric is to take the $\mathbf{P}^{abstr,\ in}_{i,i+1}$ points and project (planar orthogonal projection) them into $\pi^{abstr}_{i,i+1}$, keeping their elevation value unchanged; then, a fitting error can be calculated between them and $z^{abstr}_{i,i+1}(t)$. This variant has the advantage of relying on the points' elevations instead of relying on their LS fitting. Note that, for both variants, by selecting the abstract cross-sections exactly halfway between two consecutive cross-sections, we expect to evaluate, on average, the fitting error at the point where it is highest since the elevation is given by the exact average of the two consecutive cross-sections' elevations. We refer to the first variant of the metric as *Integral Abstract Error* and the second variant as *Point Abstract RMSE* (or MAE, depending on the considered fitting error), and report the results of the experiments in Table II. As expected, the fitting errors of abstract cross-sections are larger than those for cross-sections shown in Table I, but the average error is similar and less than 10mm.

Our second proposed metric takes into account the fact

TABLE II

CUSTOM METRICS, AVERAGED OVER THE NUMBER OF CROSS-SECTIONS.

First proposed metrics	$q = 2$
Average Integral Abstract Error [mm]	1.608
Average Point Abstract RMSE [mm]	5.966
Average Point Abstract MAE [mm]	4.385

TABLE III

CUSTOM METRICS, "POLY" REFERS TO THE GLOBAL FITTING WITH LS.

Second proposed metrics	Ours $q = 2$	Poly $q = 2$	Poly $q = 3$	Uniform $\mu(\mathbf{Z}^{surf})$
Full fitting RMSE [mm]	**5.245**	89.58	76.26	969.9
Full fitting MAE [mm]	**3.656**	73.28	60.78	791.8

that, by approximating the clothoids with dense linear splines, a sequence of polygons approximating the drivable area can be derived. Then, once outliers are removed using Z-scores, a point set $\mathbf{P}^{surf}$ representing the points that belong to the road surface can be obtained. Moreover, for each point $(x^{surf}_j, y^{surf}_j, z^{surf}_j) \in \mathbf{P}^{surf}$, it is possible to know in which polygon it falls, thus allowing us to estimate the t and v parameters within the corresponding domain, from which we can derive the corresponding elevation z^{proj}_j given by the 3D road surface. From these parameters, the considered error metrics (RMSE and MAE) can be calculated. Results are reported in Table III, confirming the considerable accuracy of our pipeline. We compared our method with standard analytical reconstructions: a surface with a uniform height $h = \mu(\mathbf{Z}^{surf})$, and a surface obtained via a global polynomial fitting of all points. Unlike these methods, our pipeline, by interpolating cross-sections that follow the road slope as it evolves, was able to accurately incorporate the road shape within the analytical form of the output surface.

We also reconstructed the surface as a mesh by linking the points in $\mathbf{P}^{surf}$ using the Ball-Pivoting algorithm [40]. As expected, the mesh presented irregularities, due to the use of triangles (Fig. 7), which are not present in the real world, nor in our reconstruction. The mesh is unable to maintain the borders as clothoids, thus losing the properties discussed in Section II and appearing inaccurate. Lastly, in terms of memory, saving the mesh to file required 3.96MB of space, while our analytical surface required only 136KB, showing evidence that our method is much more scalable in this sense.

V. CONCLUSIONS

In this work, we presented a novel approach designed to automatically extract and reconstruct the 3D surface of the road using a set of sensors installed on a vehicle. Starting from a 2D clothoidal map of the road line markings, obtained

Fig. 7. Portion of track mesh reconstructed via Ball-Pivoting [40]. Triangles represent irregularities not present in reality and borders are imprecise.

from vision and precise GPS sensors, the algorithm processes the LiDAR point cloud data and derives 3D cross-sections at regular intervals along the reference line. These cross-sections are then interpolated in a parametric space to generate the global surface of the drivable road while preserving the road's curvature properties. Moreover, unlike mesh-based reconstructions, the proposed smoother representation: keeps the borders as clothoids (ensuring high precision), obviates the irregularities caused by triangles, allows virtually infinite resolution rendering, and requires much less disk space for storage, thus resulting in a more scalable and practical solution in this respect. Our experimental results showed that the pipeline is fast and accurate, making it very effective for tasks such as automatic map reconstruction.

REFERENCES

[1] G. Chen, G. Esch, P. Wonka, P. Müller, and E. Zhang, "Interactive procedural street modeling," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 1–10, 2008.

[2] D. Craciun, J.-E. Deschaud, and F. Goulette, "Automatic ground surface reconstruction from mobile laser systems for driving simulation engines," *Simulation*, vol. 93, no. 3, pp. 201–211, 2017.

[3] D.-L. Chen and Y.-Y. Lu, "3D road surface reconstruction based on point clouds data assimilation algorithm," in *Proc. of International Conference on Mechanical, Electronic and Information Technology Engineering*, 2017, pp. 1–5.

[4] D. Chen and X. He, "Fast automatic three-dimensional road model reconstruction based on mobile laser scanning system," *Optik*, vol. 126, no. 7-8, pp. 725–730, 2015.

[5] M. Montemerlo and S. Thrun, "A multi-resolution pyramid for outdoor robot terrain perception," in *Proc. of AAAI*, 2004, pp. 464–469.

[6] A. Wedel, H. Badino, C. Rabe, H. Loose, U. Franke, and D. Cremers, "B-spline modeling of road surfaces with an application to free-space estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 572–583, 2009.

[7] H. Brunken and C. Gühmann, "Road surface reconstruction by stereo vision," *Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 88, no. 6, pp. 433–448, 2020.

[8] S. Sugimoto, T. Kato, K. Motooka, and M. Okutomi, "Direct ground surface reconstruction from stereo images," *Information and Media Technologies*, vol. 8, no. 4, pp. 1051–1055, 2013.

[9] D. Charnley and R. Blissett, "Surface reconstruction from outdoor image sequences," *Image and Vision Computing*, vol. 7, no. 1, pp. 10–16, 1989.

[10] A. Mahmoudzadeh, S. F. Yeganeh, S. Arezoumand, and A. Golroo, "3D pavement surface reconstruction using an RGB-D sensor," in *Proc. of International Electronic Conference on Sensors and Applications*, 2019, pp. 1–6.

[11] Y. Zhang, C. Chen, Q. Wu, Q. Lu, S. Zhang, G. Zhang, and Y. Yang, "A kinect-based approach for 3D pavement surface reconstruction and cracking recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3935–3946, 2018.

[12] B. Li, Y. Guo, J. Zhou, Y. Cai, J. Xiao, and W. Zeng, "Lane detection and road surface reconstruction based on multiple vanishing point & symposia," in *Proc. of IEEE Intelligent Vehicles Symposium*, 2018, pp. 209–214.

[13] C. Zhang, "Towards an operational system for automated updating of road databases by integration of imagery and geodata," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 58, no. 3-4, pp. 166–186, 2004.

[14] S.-J. Yu, S. R. Sukumar, A. F. Koschan, D. L. Page, and M. A. Abidi, "3D reconstruction of road surfaces using an integrated multi-sensory approach," *Optics and Lasers in Engineering*, vol. 45, no. 7, pp. 808–818, 2007.

[15] B. Forkel and H.-J. Wuensche, "Dynamic resolution terrain estimation for autonomous (dirt) road driving fusing LiDAR and vision," in *Proc. of IEEE Intelligent Vehicles Symposium*, 2022, pp. 1181–1187.

[16] S. A. Gargoum, K. El-Basyouny, K. Froese, and A. Gadowski, "A fully automated approach to extract and assess road cross sections from mobile LiDAR data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 11, pp. 3507–3516, 2018.

[17] Y. Tsai, C. Ai, Z. Wang, and E. Pitts, "Mobile cross-slope measurement method using lidar technology," *Transportation Research Record*, vol. 2367, no. 1, pp. 53–59, 2013.

[18] B. Gallazzi, P. Cudrano, M. Frosi, S. Mentasti, and M. Matteucci, "Clothoidal mapping of road line markings for autonomous driving high-definition maps," in *Proc. of IEEE Intelligent Vehicles Symposium*, 2022, pp. 1631–1638.

[19] H. Marzbani, M. Simic, M. Fard, and R. N. Jazar, "Better road design for autonomous vehicles using clothoids," in *Intelligent Interactive Multimedia Systems and Services*, 2015, pp. 265–278.

[20] E. Lambert, R. Romano, and D. Watling, "Optimal path planning with clothoid curves for passenger comfort," in *Proc. of International Conference on Vehicle Technology and Intelligent Transport Systems*, 2019, pp. 609–615.

[21] G. Kaur and D. Kumar, "Lane detection techniques: A review," *International Journal of Computer Applications*, vol. 112, no. 10, pp. 4–8, 2015.

[22] B. Yu and A. K. Jain, "Lane boundary detection using a multiresolution Hough transform," in *Proc. of International Conference on Image Processing*, vol. 2, 1997, pp. 748–751.

[23] T.-T. Tran, C.-S. Bae, Y.-N. Kim, H.-M. Cho, and S.-B. Cho, "An adaptive method for lane marking detection based on HSI color model," in *Proc. of International Conference on Intelligent Computing*, 2010, pp. 304–311.

[24] J. Tang, S. Li, and P. Liu, "A review of lane detection methods based on deep learning," *Pattern Recognition*, vol. 111, p. 107623, 2021.

[25] Z. Qin, H. Wang, and X. Li, "Ultra fast structure-aware deep lane detection," in *Proc. of European Conference on Computer Vision*, 2020, pp. 276–291.

[26] D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool, "Towards end-to-end lane detection: An instance segmentation approach," in *Proc. of IEEE Intelligent Vehicles Symposium*, 2018, pp. 286–291.

[27] P. Cudrano, S. Mentasti, M. Matteucci, M. Bersani, S. Arrigoni, and F. Cheli, "Advances in centerline estimation for autonomous lateral control," in *Proc. of IEEE Intelligent Vehicles Symposium*, 2020, pp. 1415–1422.

[28] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. of International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.

[29] C.-Y. Chen and R. Klette, "Image stitching–Comparisons and new techniques," in *Proc. of International Conference on Computer Analysis of Images and Patterns*, 1999, pp. 615–622.

[30] H. A. Mallot, H. H. Bülthoff, J. J. Little, and S. Bohrer, "Inverse perspective mapping simplifies optical flow computation and obstacle detection," *Biological Cybernetics*, vol. 64, no. 3, pp. 177–185, 1991.

[31] M. Bertozzi, A. Broggi, and A. Fascioli, "Stereo inverse perspective mapping: Theory and applications," *Image and Vision Computing*, vol. 16, no. 8, pp. 585–590, 1998.

[32] Y. Chen, Z. Xiang, and W. Du, "Improving lane detection with adaptive homography prediction," *The Visual Computer*, pp. 1–15, 2022.

[33] J. Jeong and A. Kim, "Adaptive inverse perspective mapping for lane map generation with SLAM," in *Proc. of International Conference on Ubiquitous Robots and Ambient Intelligence*, 2016, pp. 38–41.

[34] M. Bellusci and M. Matteucci, "Advances in real-time online vehicle camera calibration via road line markings parallelism enforcement," in *Proc. of IEEE Intelligent Vehicles Symposium*, 2022, pp. 1511–1516.

[35] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Proc. of International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[36] E. Bertolazzi and M. Frego, "G^1 fitting with clothoids," *Mathematical Methods in the Applied Sciences*, vol. 38, no. 5, pp. 881–897, 2015.

[37] I. Ben-Gal, "Outlier detection," in *Data Mining and Knowledge Discovery Handbook*, 2005, pp. 131–146.

[38] E. Kreyszig, K. Stroud, and G. Stephenson, "Advanced engineering mathematics," *Integration*, vol. 9, no. 4, 2008.

[39] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving video database with scalable annotation tooling," *arXiv preprint arXiv:1805.04687*, 2018.

[40] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, pp. 349–359, 1999.