Received December 18, 2018, accepted January 31, 2019, date of publication February 4, 2019, date of current version February 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2897348

Self-Generated OSM-Based Driving Corridors

JORGE GODOY [1], **ANTONIO ARTUÑEDO** [2], **AND JORGE VILLAGRA** [2]

[1]Centre for Automation and Robotics UPM-CSIC, Universidad Politécnica de Madrid, 28006 Madrid, Spain

[2]Centre for Automation and Robotics UPM-CSIC, Spanish National Research Council, 28500 Madrid, Spain

Corresponding author: Jorge Godoy (jorge.godoy@upm.es)

This work was supported in part by the Spanish Ministry of Economy, Industry and Competitiveness with National Project TCAP-AUTO under Grant RTC-2015-3942-4, in part by the Spanish Ministry of Science, Innovation and Universities with National Project COGDRIVE under Grant DPI2017-86915-C3-1-R, and in part by the European Commission through the Project PRYSTINE under Grant ECSEL-783190-2. The work of J. Godoy was supported by the Juan de la Cierva Fellowship Program through the Spanish Ministry of Economy, Industry and Competitiveness.

ABSTRACT Finding the boundaries of the drivable space is a key to the development of any advanced driver assistance systems with automated driving functions. A common approach found in the literature is to combine the information of digital maps with multiple on-board sensors for building a robust and accurate model of the environment from which to extract the navigable space. In this sense, the digital map is the crucial component for relating the location of the vehicle and identifying the different road features. This paper presents an automatic procedure for generating driving corridors from OpenStreetMap. The proposed method expands the original map representation, replacing polylines by polynomial-based roads, whose sections are defined using cubic Bézier curves. All curves are automatically adjusted from the original road description, thus generating an efficient and accurate road representation without human intervention. Finally, the driving corridors are generated as a concatenation of the modified road sections along a planned route. The proposed approach has been validated in a peri-urban environment, for which corridors where successfully generated in all cases.

INDEX TERMS Path planning, autonomous vehicles, navigation corridors, OpenStreetMaps.

I. INTRODUCTION

A robust and consistent driving environment model is key for any advanced driver assistance system and automated driving function. A digital map is therefore a crucial component that imposes constraints on the navigation solution and enables the systems to relate the positioning information with the physical location of the vehicle in the road. Hence, map-based systems for automated vehicles, where map features are associated with sensor measurements to derive the vehicle localization and its surrounding drivable space, require high precision information. More generally, road models for automated driving have to comply with geometry, implementation and usability requirements [1]. The first is related to the completeness and accuracy of the mapped driving environments, including lane-level information. The second type of requirement aims at defining efficient and compact data structures, so that V2X-based road information can be downloaded, updated and processed via wireless networks. The last condition for these maps to be operational is to have a bounded computational complexity, allowing to compute the road geometry information in a real-time setting [2].

Most of existing digital maps -GDF [3], KiWi [4], Navteq [5]- are based on road level data, which is useful for navigation-like applications, but impractical for automated vehicles. The most widespread practice to obtain lane information is currently the use of annotated spatial maps. These maps provide precise lane information, but they have to be actively annotated by human users, which poses the problem of availability of these maps especially for secondary roads and streets [6]. Although some lane-level mapping techniques have been recently proposed -using data-fusion from COTS on-board sensors data [7], the most widespread solution for precise digital map generation systems has a really high cost as they rely on specialized and expensive sensors, also involve significant manual efforts for data analysis, and often require huge storage and processing capabilities [1].

Different standards have been proposed in the past aiming at fulfil some of these requirements. RNDF is the Road Network Definition File that was designed for DARPA urban

The associate editor coordinating the review of this manuscript and approving it for publication was Bora Onat.

challenge in 2007, in which the basic structure segment-lane-waypoint are included to provide basic information to driverless vehicles [6]. Nowadays, OpenDRIVE and NDS are well-known map standard providers, but their complicated map model and inaccessibility to public cause difficulties for practical uses.

Open street maps (OSM) [8] was born to cope with these constraints, proposing an open data infrastructure to which volunteers, companies or governmental organizations all over the world can contribute. Similarly to RNDF, OSM data is very good at representing topological information as well as positions, but has drawbacks in representing the geometry itself. Lanelets [9] enhance OSM data structure to provide a good trade-off between topology and geometry description, while allowing a reasonable usability and implementation. They model the relevant parts of the environments by means of atomic, interconnected drivable road segments, geometrically represented by a left and right bound. As a result, the road bounds are approximated by a list of points, yielding a polygonal line or a polyline. This approximation is effective in highways, but neither too inaccurate nor complete for urban or peri-urban environments, where roundabouts, crossroads or curvy stretches are often present. Indeed, in roundabouts the linear interpolation between sharp points may deviate from the true curve by more than one meter. In addition, the geometrical description of intersection's internal parts are often omitted or simplified with arc segments [10].

Therefore, a more refined curve representation for various lane shapes is required, taking into consideration the road continuity and the introduction of virtual lanes when required, as proposed in [10] or [11] for specific situations. This paper provides a solution to overcome these difficulties, contributing to the state of the art in the following aspects:

- A novel data structure is proposed to enhance the geometry accuracy of OSM maps, while preserving its topological structure in accordance with the real road network topology, and therefore keeping its efficiency and usability. To that end, a lane-level map will be generated guaranteeing the global continuity both in location and tangent orientation (G^1 continuity) at every node through the use of Bézier curves and a systematic procedure to softly concatenate them. This feature will be applied also to precise virtual lanes in complex roundabouts and/or intersections.
- A novel mechanism to automatically generate driving corridors that can be used in automated vehicles to (i) constrain the perception and localization systems to the surrounding lane(s), and (ii) to make the right decision in complex urban scenarios, and therefore to properly plan the motion to be adopted by the vehicle.

The remainder of the article is organized as follows. Section II gives an overview of the related works on road modelling for automated driving. Section III presents the general schema of the proposed algorithm, including the data structure of the map. The details of each one of the algorithm stages are described in sections IV, V and VI. Section VII presents the results of the algorithm both for map adjustment and driving corridor generation. Finally, Section VIII draws some concluding remarks.

II. RELATED WORKS

An important number of map-related works for autonomous vehicles have been presented in the last decade. They range map-based localisation and navigation [12], [13], acquisition, filtering and optimization techniques for high quality maps [14] or road modelling strategies [15]. However, not many combine smooth road geometry models and map-based advanced driving assistance systems (ADAS), whose state of the art is summarized below.

The simplest but commonly used representation for lanes in digital maps is based on polylines, composed of a sequence of segments, resulting in turn in concatenated polygons [16]. Among this family of mapping data structures, the RDDF was one of the first formats to describe route networks which was mainly used for unpaved desert tracks. It consisted of a simple list of longitudes, latitudes, and corridor widths that define the course boundaries, and a list of associated speed limits.

In RNDF the basic structure segment-lane-waypoint was included to provide basic information to driverless vehicles. Multi-lane stretches of roads could be then modelled as road segments, defined by a set of waypoint, which in addition incorporates metadata to denote relevant elements, such as stop sign, intersection entry or exit, or checkpoints. Although the RNDF specification is fairly easy to understand and sufficient enough to map most road networks, it misses several features and reveals flaws when designing specific road characteristics [17]. Indeed, it is an efficient representation for real-time applications, but do not comply with real roads geometry, because they can exhibit discontinuities and often not describe curved structures in a suitable way. As a matter of fact, for a precise approximation of these structures, a relatively high number of line segments would be 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.

As aforementioned, OSM provides a good topological information, but geometry information and accuracy is also poor. Another open and freely available format is OpenDrive [18], which has enhanced features with respect to OSM, as it permits to mark roads being in a tunnel or on a bridge or even tilting and cross fall of roads. However, its deployment is still too limited when compared to the one of OSM, available worldwide with an impressive degree of coverage.

A. SMOOTH ROAD GEOMETRY MODELS

The lanelet model also uses polylines as geometric representation but provides continuous tangents at the junctions of the segments thanks to non-Euclidean point-to-segment projection [9]. This approach provides an interesting real-time solution without discontinuity issues.

TABLE 1. Overview of the main features for the different road geometry mapping models.

Features / Models	Primitive smoothness	Lane Metadata	Topological information	Coverage/ Generation complexity	Travel data independence	Database Query	Centreline available (for path planning)	Margins independent from centreline
RDDF [16]	0	+	0	++	+++	0	0	0
RNDF [17]	0	++	+	++	+++	0	0	0
RNDGraph	++	++	+	++	+++	0	0	0
OSM [8]	0	++	++	+++	+++	0	0	0
OpenDrive [18]	+++	+++	+++	+	+++	0	++	++
Lanelet [9]	+	+++	++	+++	+++	0	+	+
Extended Maps [19]	++	+++	++	++	+++	0	+	++
Akima intrepolation [17]	++	+	+	+	0	0	++	++
Enhanced maps [10]	+++	+	+	+	0	0	+++	+++
ADASIS [5]	+	+++	+++	++	+++	++	+++	0
Self-generated corridors	+++	++	++	+++	+++	+	+++	++

In RNDFGraph the course of the road is mimicked with spline interpolation. By using splines, only the support points for the spline need to be stored, and every point between the support points can be interpolated. The challenge, however, is to place the support points in such a way, so the resulting spline interpolation matches the course of the road. Support points are also the link to the graph. The RNDFGraph provides a road network with additional information for autonomous behaviour such as continuous spline sampling, lane relationships and lane change information, to be used by subsequent path-planning and low-level controller modules.

Extended Maps [19] model the world in terms of interconnected clothoids, line and circle segments, following the Dubins paths structure [20]. Clothoids are used here to provide a smooth steering phase when passing the lane sections since their curvature changes linearly [21]. However, the offset curve of a clothoid is not a clothoid anymore, and it is therefore not possible to easily build corridor using this primitive. Thus, it is reasonable to use alternative curves models, providing smooth continuous and differentiable curvilinear coordinates, such as approximative B-splines [22], Hermite interpolating splines [23], [24], NURBS [25], or arc splines [26].

The Akima interpolation in [17] is a continuously differentiable sub-spline interpolation whose resulting spline is less affected by outliers than cubic spline interpolation. According to [2] piecewise polynomials are more effective in terms of usability than clothoid or B-splines, because the tangent angle and the curvature of the road can be calculated by conducting simple arithmetic operations. However, the procedure proposed by the authors only works with positioning data, from which it determine the number of piecewise polynomials of the overall curve. In this connection, [10] propose an iterative method to automatically generate the lane geometry using fixed and variable control points, which can effectively ensure the accuracy with limited number of control points. Also in these two cases accurate samples of the centreline are required and therefore it is not suitable to enhance OSM maps.

B. MAP-BASED ADAS

ADASIS [5] is a standardization initiative for a data model to represent map data ahead of the vehicle (ADASIS Horizon) using a list of distinct potential future corridors, modelled as road segments. This standard does not store yet additional road features -e.g. traffic lights or signs- neither their position, as OSM-based systems do. However, they are not conceived as a map database and do not need to be queried to obtain the corridors, as happens with lanelets or similar approaches.

ADASIS allows also to obtain the centreline of the corridor as a smooth function -which is very useful for trajectory planning [27]. However, since these corridors are modelled as paths, they do not consider the possibility to characterize the lane margins independently of this centreline, which may be really helpful in situations where lane margins are asymmetrically distributed with respect to the lane center.

Table 1 summarizes the different mapping models and strategies described in this section with regard to the most relevant features to generate driving corridors. Note that a subjective mark ranging from 0 (lower) to $+++$ (higher) is attributed to each of these characteristics. These qualitative comparison is based on the description of each approach in the literature and, in some cases, to our experience as users. It can be observed that although OpenDrive, Lanelets, Extended Maps and ADASIS exhibit a good compromise between the richness of topological information, the precision of its representation, and the suitability of such representation as driving corridors for motion planning, each one of them has still some weakness. More specifically they have room for improvement in terms of coverage, primitive smoothness, availability of centreline, and the existence of margins independent of the centreline, respectively. As a result, the proposed methodology to automatically generate self-driving corridors is, to the best of our knowledge, the best trade-off between map-based ADAS and road geometry models. The strategy introduced in this paper combines the advantages of both approaches, introducing a procedure to automatically generate OSM-based driving corridors with the following three main features:

- It allows to keep the original OSM data structure, preserving its topology and attributes richness, but enhancing its geometry with a G^1 concatenation of Bézier curves.
- It adopts a lane-level approach associating one corridor to every lane, so that corridors will merge or divide where lanes merge or split.

- In every corridor, two smooth profiles for left and right lane margins will be defined. They will be considered either with fixed length or will absorb potential road asymmetries using computer vision systems for lane detection, as in [9].

III. ALGORITHM OVERVIEW

OSM implements a topological data structure, where roads are represented as sets of waypoints or nodes. This means that no geometric information is stored on the database and, despite the number of nodes defining them, any road is defined as a series of straight segments. This format fully meets the requirements for map representation and route generation but it is not accurate enough for representing curve features on roads, especially when the final goal is to define the navigable space for a vehicle.

In this context, we present an automatic procedure that expands the OSM definition, generating a better approximation to the real shape of roads. To that end, the adjacency among the nodes is analysed for a given area, identifying and classifying all road junctions in order to generate an efficient and accurate polynomial based-map representation using Bézier curves.

Bézier curves are defined as

$$B(t) = \sum_{i=0}^{n} b_{i,n}(t) P_i, \quad 0 \le t \le 1 \tag{1}$$

where n is the order of the polynomial, P_i the control points defining the curve, and $b_{i,n}$ the Bernstein coefficients given by:

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i = 0, \ldots, n \tag{2}$$

Despite being firstly implemented in computer graphics, some properties of Bézier curves have encouraged their use on other areas such as path planning [28]–[30]. Those of interest for this work are:

- Curve lies within the area defined by the control points.
- Start and end points are defined by P_0 and P_n, respectively.
- The derivative of a Bézier curve is, in fact, another Bézier curve. This means no derivative discontinuities.
- Tangent direction at the beginning and end of the curve is defined by $\overrightarrow{P_0P_1}$ and $\overrightarrow{P_{n-1}P_n}$, respectively.

Depending on the polynomial degree, it would be possible to define an entire road as one Bézier curve. However, the higher the degree, the higher the complexity of the curve in terms of computation time and control points adjustment. Therefore, the common approach in computer graphics is to define complex paths by concatenating quadratic and cubic curves, taking into consideration the continuity constraints. From a geometrical view point, the continuity at the junction of two curves is expressed in terms of G^k, where k is the number of continuous derivatives at the junction. For example, G^1 refers only to the continuity of the tangent direction, while G^2 includes the continuity of the curvature vectors.

In view of all the above, the first task of this algorithm is to modify the map description, replacing the straight segments among nodes by cubic Bézier curves and automatically adjusting the control points for fitting roads. This is done following a two-stage procedure: Node expansion and Bézier adjustment.

On the first stage, the continuity among the road sections traversing each node is classified. To that end, a new property named *raw continuity* is defined for each pair of sections, referring to the hardness of their junction isolated from the rest of the map. For example, on a almost-straight road, as a highway, the road sections joint smoothly, with low changes on path angle and curvature. However, the curvier the road, the sharper are the junctions, reflecting on higher changes between segments and lower continuity. This step also modifies the data structure of the nodes for including information about the neighbours, the related segments and the hardness of all the possible junctions, giving the name to the stage.

The second stage adjusts the control points of Bézier segments according to the fittest junction involving them. To that end, node junctions are sorted from softer to harder following several criteria, adjusting the segments in the same order. As final output, this stage produces a map where roads' centrelines have been automatically modified and better describe the roads.

Finally, once the map is adjusted, the final task is to define the drivable space for the vehicle. To that end, a third and final stage combines a route planner with the polynomial description of the roads, generating the navigable corridor as a concatenation of Bézier segments along a planned route. Both left and right boundaries of the corridor are defined by Bézier curves, generated as displacement of the segments centrelines.

Algorithm 1 shows a general overview of the proposed approach while on the following sections a detailed description of each stage is presented. Moreover, for a better comprehension of the algorithm and methods implemented, the next subsection summarizes the main concepts and notations used along this paper. Part of them are inherited from the OSM data structure for map storage and definition and other have been created for this implementation.

A. DEFINITIONS

- **Tags**: are key-value pairs used to describe specific features about the object they are attached.
- **Nodes**: define single points on the earth surface using WGS84 format. In this work, nodes' data structure has been modified for including further information about neighbour nodes, connecting segments and junctions/links traversing it. Along the next sections, nodes are noted as N_i, where i is the node unique id defined by OSM.
- **Ways**: are polylines representing different map features as ordered list of nodes (N_1, N_2, ..., N_n). The OSM ontology defines several types of features that can be

Algorithm 1 Algorithm Overview

load *Map* ***from*** *OSM*
foreach *Node in Map* **do**
  foreach *Neighbour* ***at*** *Node* **do**
    Extract segment features
  end
  foreach *Link* ***at*** *Node* **do**
    Classify as *Hard* or *Soft*
  end
  foreach *Segment* ***at*** *Node* **do**
    Set and adjust two closest Bézier control points
  end
end
Route=PlanRoute(Origin, Destination)
foreach *Segment* ***in*** *Route* **do**
  Extract lane *Boundaries* from *Segment*
  if *Link to next Segment is Soft* **then**
    Add *Boundaries* to *Corridor*
  end
  else
    Trim *Boundaries*
    Add *Boundaries* to *Corridor*
    Create *Joining Section*
    Add to *Joining Section* to *Corridor*
  end
end
Send *Corridor* to planning module

encoded as ways, but for this work only those marked with the key *highway* are considered.

- **Segments**: is one of the new concepts used on this paper. A segment refers to the road section that joins two adjacent nodes N_a and N_b. Each instance S_j includes inherited way information about the road such as number of lanes, traffic direction and road width. Along this paper, segments are sometimes mentioned as *Straight segments* or *Bézier segments*, referring to the straight sections used by OSM or the polynomial representation defined in this work.
- **Links**: are objects describing the junction among two segments S_a and S_b sharing a common node N_l. Links metadata includes joint features such as junction angle, angle bisector, segments id, hardness and link continuity.

IV. NODE EXPANSION

Having the OSM map for a given area, the goal of this stage is to analyse the adjacency among the nodes in order to estimate the raw continuity of the road segments. To that end, the area is first explored way by way, creating a segment instance S_j for each pair of nodes N_a and N_b connected over a way.

Since no data about the road geometry is stored on OSM database, segments are initialized as straight cubic Bézier curves, where the first and last control points (P_0^j and P_3^j) correspond to N_a and N_b, while the second and third points

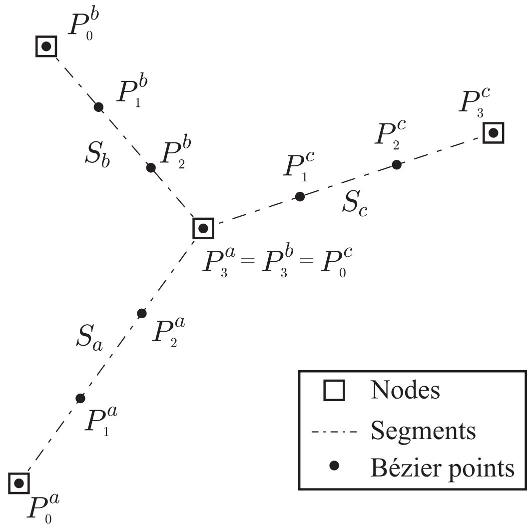

FIGURE 1. Example of Segments with Bézier control points.

(P_1^j and P_2^j) are equally distributed between both nodes (see Fig. 1).

Relevant information such as traffic direction, number of lanes and road width is transferred from ways into segments, thus facilitating future adaptation of road sections individually. Moreover, the nodes' basic data structure defined by OSM is modified for storing the neighbours nodes, connecting segments and raw continuity among traversing segments.

Once ways have been processed and the information has been loaded into nodes, the area is re-explored node by node in order to determine the continuity among road segments. To that end, a link instance L_k is created for each pair of segments traversing a node. Thus, for a node N_i with n_v^i neighbours, the number of possible links is

$$n_l^i = \binom{n_v^i}{2} \tag{3}$$

A hardness binary variable h_k describes the continuity at each junction, being $h_k = soft$ for those links possibly representing a smooth continuous road, while $h_k = hard$ for those that definitively do not. This value is set according to the link shape, taking into account the length, width and joining angle of the segments, as described below.

The selection of these features is the result of analysing the shape of several segment junctions at continuous and discontinuous scenarios such as highways, urban roads, mergings, intersections and roundabouts. From this review it was found that considered continuous roads are usually described by short segments and small changes on road tangents, being the opposite for discontinuous roads.

Let us consider a link L_k joining the segments $S_a = \overline{N_0N_1}$ and $S_b = \overline{N_0N_2}$ at the node N_0 (see Fig. 2). First, the angle α_k is extracted from the vector representation of the segments. If the angle is acute, L_k has a sharp shape and therefore segments are discontinuous at the junction, then $h_k = hard$. On the other hand, if the angle is obtuse the value of h_k

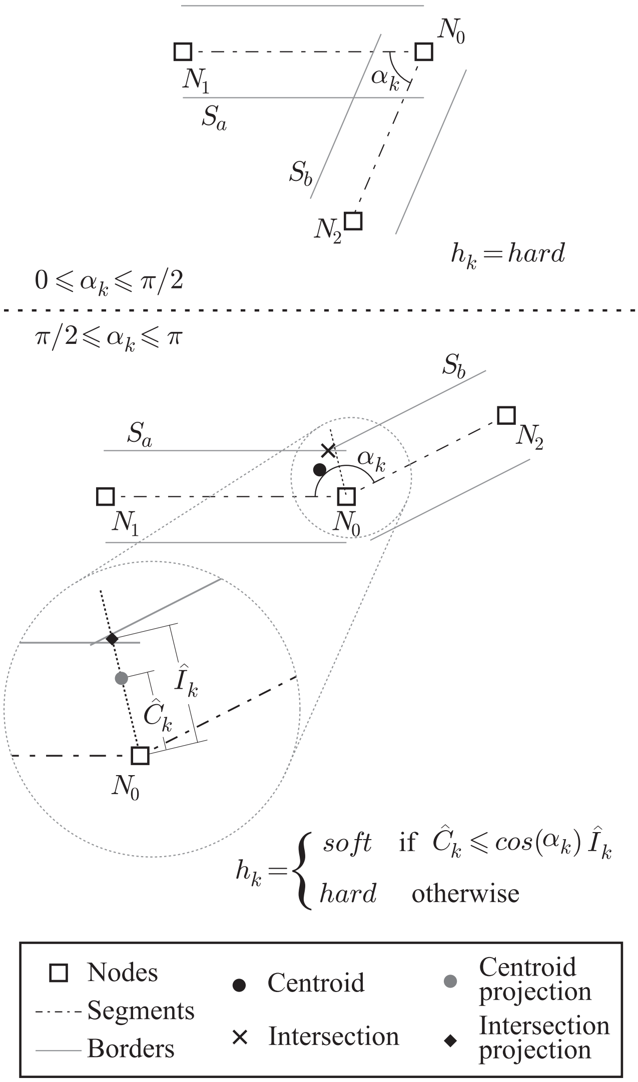

FIGURE 2. **Link hardness calculation.**

depends also on the length and width of the road sections associated to the segments.

As a way to simplify the analysis of all the possible scenarios, the relation among length, width and angle of the segments is expressed by the position of singular points: $\overset{\triangle}{N_0N_1N_2}$ triangle centroid C_k and border intersection point I_k. On the one hand, the triangle centroid relies only on the position of the links nodes and therefore, the segments length and angle, while on the other hand, the intersection on the inner side of the angle relies only on the segments width and angle.

Both the centroid and the intersection point are projected over the angle bisector $\hat{m}_k$, resulting in two points whose distance to N_0 is $\bar{C}_k$ and $\bar{I}_k$, respectively. Finally, for the link to be considered *soft* the following condition must be satisfied:

$$\bar{C}_k \leqslant \kappa * cos(\alpha_k) * \bar{I}_k \tag{4}$$

where κ is implemented as an adjustable threshold for link classification and segment modification. In this work, hardness is calculated using $\kappa = 2$, which was experimentally set after analysing the results for different maps of urban and peri-urban environments.

V. BÉZIER ADJUSTMENT

Once the nodes have been expanded and the raw continuity has been determined for all the possible links, there is enough data for adjusting the shape of the segments so they better fit the real roads. This is done by modifying the Bézier control points of the segments according to the information provided by the soft links at each node. The reason only soft links are considered in this step is because they represent the junctions were, according to our criteria, the road can be continuous.

Node by node, the proposed algorithm iterates over the soft links, selecting a target link L_t at a time for adjusting the segments associated to it. Since links only provide information about the junction of the segments, these are modified by halves, adjusting only their two control points closest to the junction. Moreover, segments are flagged once they have been modified so no further changes can be introduced at that node, thus avoiding possible overwrites among the links.

The criteria for selecting the target link is key for the adjustment result. Given that the continuity of the road is the main priority for this work, target links are selected according to the following priority rules, in order:

1) Both link segments have the same number of lanes and road width and none of them is flagged as processed.
2) Segments does not have the same number of lanes but none of them has been processed.
3) At least one of the segments remains unprocessed.

In case any of these criterion returns more than one link, those with no conflict on the associated segments are processed first. For those sharing a segment S_j, the best option is selected as the link whose angle α_t is closest to the junction angle at the opposite side of S_j. By doing so the changes in road tangent and curvature are minimized. If the segment S_j has not yet been processed at the opposite side, the adjustment at the current node is paused until angle information is available at the neighbour node.

The procedure for adjusting the control points depends on the rule triggered for the target link at the node N_i (See Fig. 3). Given a target link L_t, joining segments S_a and S_b, let us define Q_0^j and Q_1^j as the two control points of S_j closest to N_i, so either

$$Q_0^j = P_0^j, \quad Q_1^j = P_1^j$$

or

$$Q_0^j = P_3^j, \quad Q_1^j = P_2^j$$

and knowing $Q_0^j = N_i$ for $j = a, b$ due to segment initialization.

For the first rule to trigger, both segments must have the same road width and number of lanes, so no modification is done over the end points Q_0^A and Q_0^B since the segments' centrelines meet at the same point. As for Q_1^a and Q_1^b, they

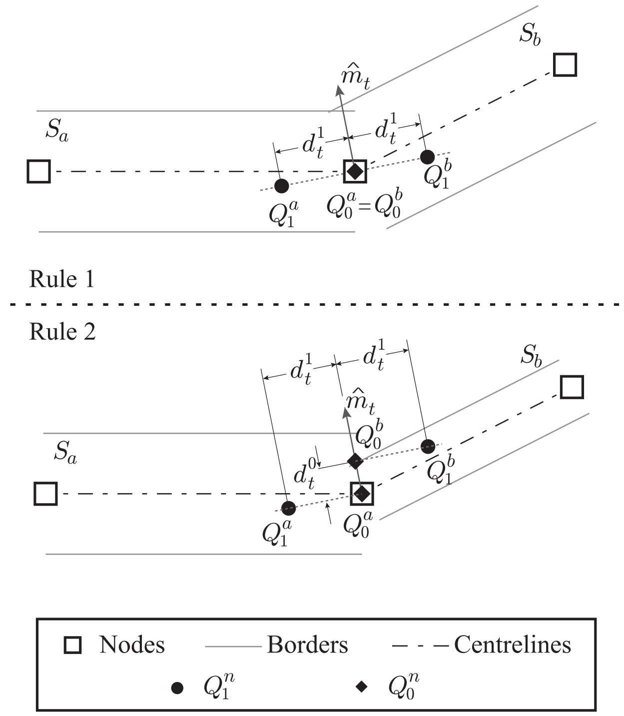

FIGURE 3. Bézier adjustment: rules 1 & 2.

are set over the perpendicular to α_t bisector $\hat{m}_t$, at the same distance d_t^1 from Q_0^a and Q_0^b, respectively. This is done so the resulting path is continuous in G^1. Parameter d_t^1 is calculated as

$$d_t^1 = min(|S_a|, |S_b|)/3 \tag{5}$$

being equivalent to the distance among equally distributed points over the shorter segment.

In case segments do not have the same number of lanes (second criterion), it is necessary to adjust the narrower segment S_n so it meets the wider segment S_w on the correct lane. To that end, Q_0^n is moved along $\hat{m}_t$ a distance d_t^0, which value depends on the target lane. The target lane is extracted comparing the number of lanes of other nodes on the same node. Once Q_0^n is set, Q_1^a and Q_1^b are adjusted by the same procedure previously described for rule 1.

Despite the displacement applied over Q_0^n, which means Q_0^a and Q_0^b are different points, this rule also generates a junction with G^1 continuity. This is because at lane level, the joining point and road tangent are equal for both segments.

Finally, the third rule processes any remaining soft link if at least one of the segments has not been adjusted. In this case, the algorithm calculates the adjustment of Q_0^j and Q_1^j as already described for prior rules, but only modifies the segment that remains unprocessed. As only one of the segments is modified, G^1 property cannot be satisfied for these specific kind of links. However, by adjusting the tangent at the end point, a smoother junction is achieved.

As a result of this stage, the straight centrelines defining the roads have been replaced by adjusted Bézier curves that better

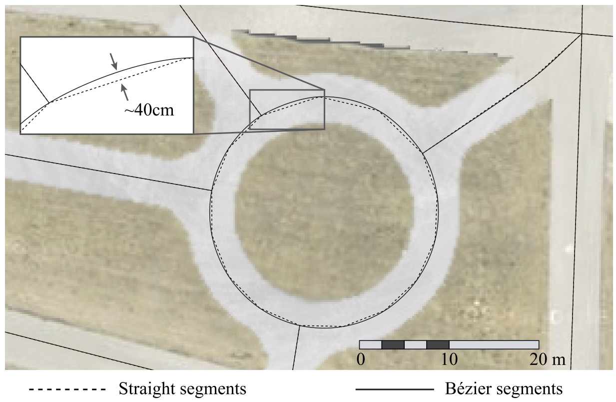

FIGURE 4. Comparison between straight and Bézier segments on a roundabout.

approximate the real paths. The effects of this enhancement on the map structure can be appreciated by comparing both map definitions at curvy roads, as the roundabout shown in Fig. 4. In the illustrated scenario, the difference between segments at middle points is around 40 cm, which is a considerable error when using the map for path planning. For bigger roundabouts, differences up to 70 cm have been found.

VI. CORRIDOR GENERATION

The third and final stage of the proposed algorithm comprises the definition of the drivable space for a vehicle. To that end, the modified Bézier segments are concatenated along a planned route, thus generating the boundaries of a navigable corridor.

For planning the route of the vehicle, a local instance of the Open Source Routing Machine (OSRM) has been implemented as router. This tool is based on contraction hierarchies [31], which reflects on fast routing times and makes it widely accepted by the OSM community. Given the start and end points coordinates, the route is returned as a list of OSM nodes ids, which combined with the expanded structure of the nodes, is translated into a succession of segments to be joined. For simplicity of this work and attending to the European regulation, it is assumed that on the ideal scenario the vehicle travels on the right-most lane on multi-lane roads.

Left and right borders of each segment are obtained by offsetting the Bézier centreline according to its information about traffic direction, number of lanes and road width. As is mentioned in [32], a Bézier curve offset cannot be described by another single Bézier curve. However, it is possible to generate a good approximation by reducing the curve to a collection of subcurves and then offsetting them. This procedure generates a good enough approximation of the displaced curve, while keeping the continuity constrains. The details of this implementation exceeds the scope of this work, but the reader can refer to [32] for more details about this method.

Having the borders of each segment, the navigation corridor is built taking into account the continuity of the links joining the segments. For those having already G^1 continuity at lane level, the borders are concatenated as they are defined. On the other hand, for those links classified as *hard* or without G^1 continuity, a joining section must be added in order to guarantee the G^1 continuity of all the boundaries.

Joining sections are auxiliary Bézier segments built by the algorithm from the tangent information of the segments to join. In this sense, the longest boundary is generated and then the opposite side created as an offset of the first.

Let us consider two discontinuous consecutive segments S_a and S_b, whose left and right boundaries are $B_a^l(t)$, $B_b^l(t)$, $B_a^r(t)$ and $B_b^r(t)$, respectively (See Fig. 5). The first step is to find the boundaries intersection points at each side (I^l and I^r) in terms of the parameter t, such that

$$I^l = B_a^l(t_a^l) = B_b^l(t_b^l) \quad \text{and } I^r = B_a^r(t_a^r) = D_b^r(t_b^r) \tag{6}$$

In case no intersection exists at the side s, then

$$t_a^s = 1 \quad \text{and } t_b^s = 0 \tag{7}$$

considering that parameter t increases in the direction of the route. Once the intersection points are found, the algorithm selects the side w where segment a is more trimmed for setting the reference t parameter for each segment. Otherwise, the generated curve could not be offset for creating the opposite border. This can be formally written as

$$t_a = min(t_a^l,\ t_a^r) \tag{8}$$

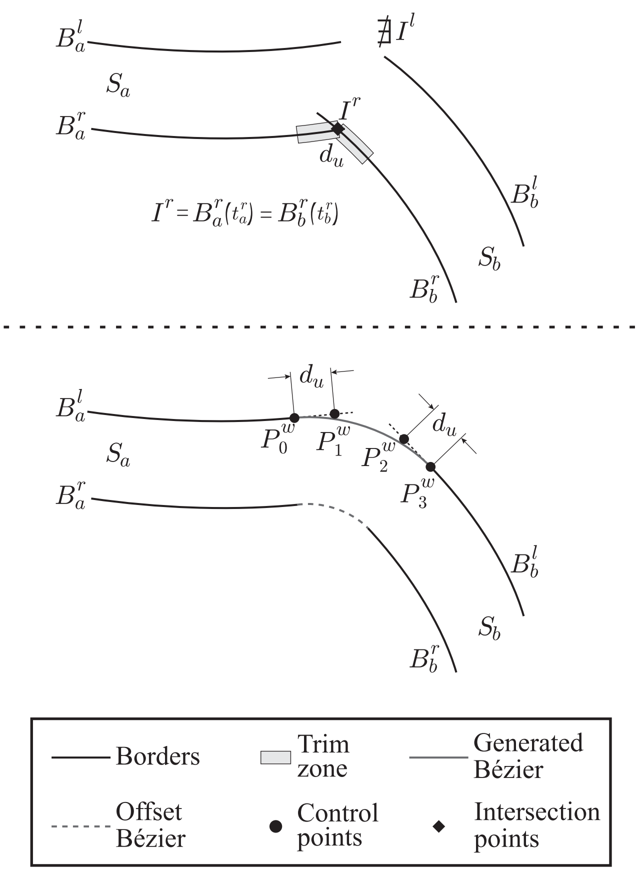

FIGURE 5. Example of Bézier generation for joining sections.

and

$$t_b = \begin{cases} t_b^l, & \text{for } t_a = t_a^l \\ t_b^r, & \text{for } t_a = t_a^r \end{cases} \tag{9}$$

Moreover, in order to reduce the sharpness of the joining section at the smallest side and avoid it collapses into a single point, both S_a and S_b are further trimmed in terms of t_a and t_b a distance d^u from the intersection. This distance is calculated in accordance to the junction angle γ_j such that

$$d_u = max(sin(\gamma_j), 0.3) \tag{10}$$

This last trim is reflected on the values of t_A and t_B as $\hat{t_A}$ and $\hat{t_B}$, respectively. The value of 0.3 was determined experimentally as a threshold for almost parallel segments. Finally, the control points of the auxiliary Bézier B^u are

$$\begin{aligned} P_0^w &= B_a^w(\hat{t_a}) \\ P_1^w &= p_0 + d_u * \dot{B}_a(\hat{t_a}) \\ P_2^w &= p_3 - d_u * \dot{B}_b(\hat{t_b}) \\ P_3^w &= B_b^w(\hat{t_b}) \end{aligned} \tag{11}$$

and the opposite border is generated as an offset curve from this one.

VII. VALIDATION AND RESULTS

The proposed algorithm for map modification and corridor generation has been implemented and applied to the surrounding area of the Centre for Automation and Robotics facilities in Arganda del Rey (Madrid, Spain). The chosen area includes both urban and interurban roads, with numerous intersections, roundabouts and merging roads. The obtained results are presented in two related subsections: (i) Comparison of map representations and (ii) Navigation corridor generation.

A. MAP REPRESENTATION

In order to demonstrate the feasibility of the proposed Bézier-based map, a comparison to the original map is presented for three different scenarios: (i) a three-way intersection, (ii) entrance and exit to a wide roundabout and (iii) small urban roundabout with all its entrances. Fig. 6 shows the superposition of map representations over an aerial image of the real roads, where the images on the left correspond to the OSM-based segments and the images on the right to the ones obtained with Bézier segments. For all images, X and Y axes represent East and North UTM coordinates, respectively, with the zero position located at the CAR facilities for better readability.

As can be seen on these figures, the proposed road representation based on Bézier curves results in a map that is visually more accurate when compared to the aerial image. This is clearly appreciated on the roundabouts (Figs. 6d and 6f) where, in contrast to the original OSM, a smooth and continuous circular shape is achieved. Moreover, as was previously

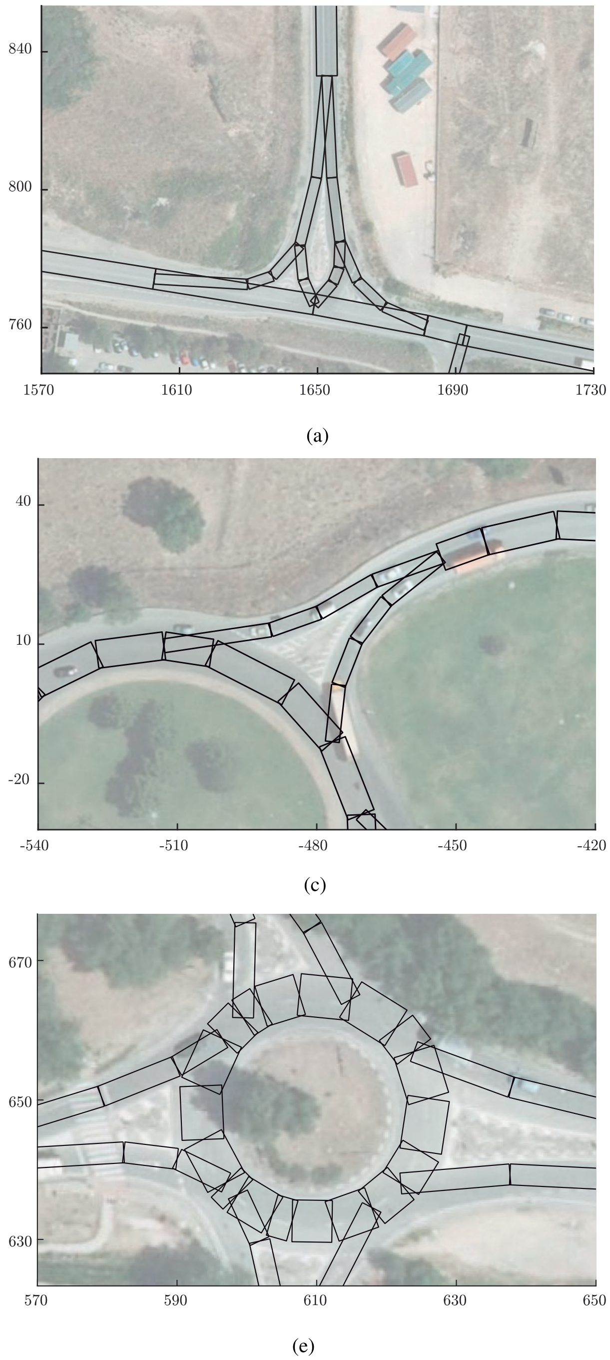

FIGURE 6. **Comparison between straight and Bézier maps. (a) Intersection - straight segments. (b) Intersection - Bézier segments. (c) Roundabout entrance & exit - straight segments. (d) Roundabout entrance & exit - Bézier segments. (e) Roundabout - straight segments. (f) Roundabout - Bézier segments.**

commented in section V, the proposed map reduces the representation error on for curvy roads, where differences up to 70 cm have been found in the explored area.

In addition to Bézier curves, the endpoint adjustment made on narrow segments is also key for map improvement. Thanks to this modification, narrow sections meet wider ones on the right point, thus reducing the lane discontinuities and overlaps found with the original map (Figs. 6b and 6f).

B. NAVIGATION CORRIDOR

Once the map of the area has been successfully adjusted and in order to validate the final stage of the algorithm, several planned routes near CAR's facilities have been introduced as input for corridor generation.

Fig. 7 shows the details about the resulting corridor for the first route, which traverses two interurban roundabouts. Notice the aerial image 7a has been rotated 45 degrees

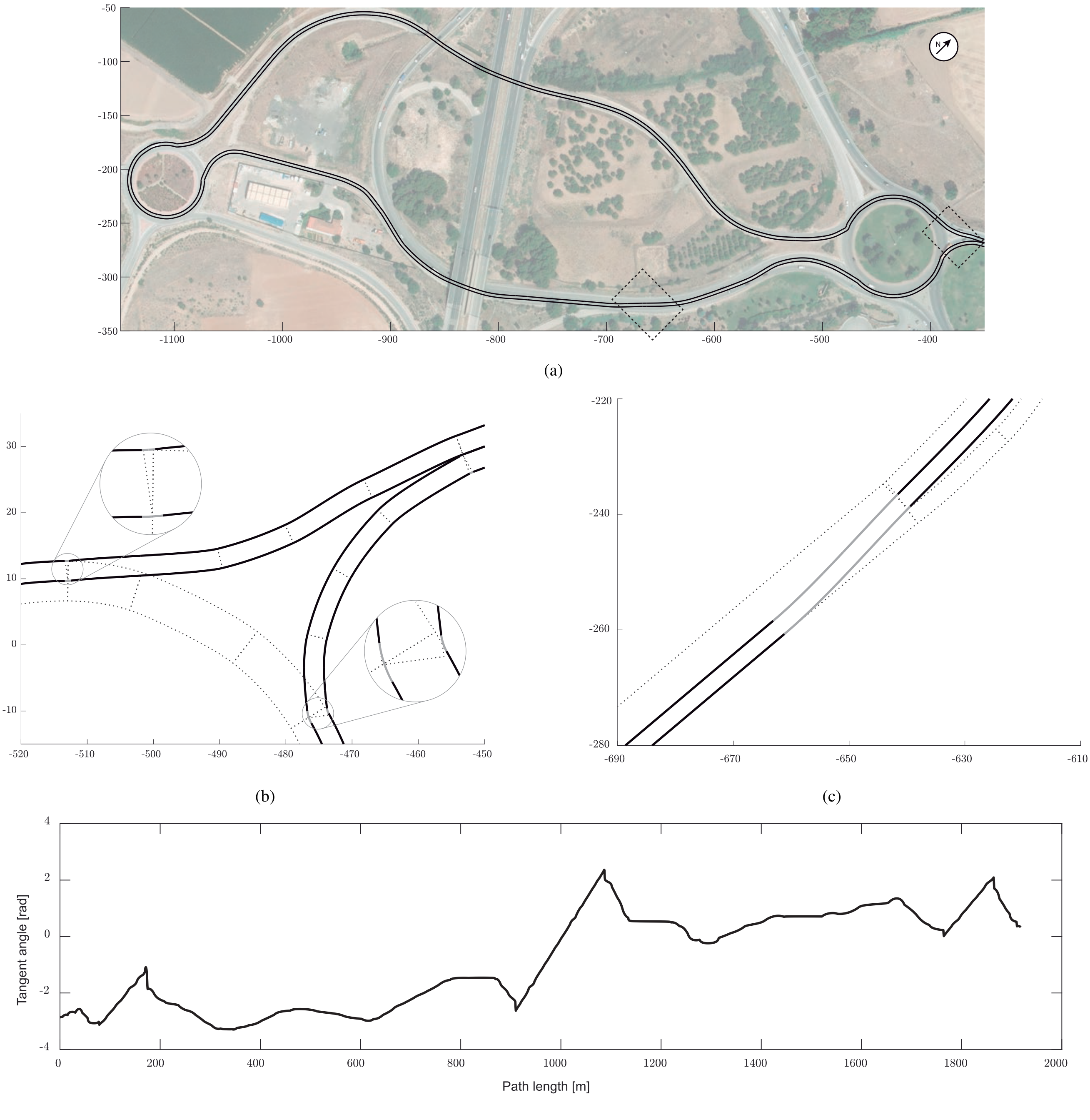

FIGURE 7. Navigation corridor for first route. (a) Navigation corridor and map correspondence. (b) Entrance and exit to roundabout. (c) Generated section on multilane segment. (d) Evolution of the corridor tangent along the planned path.

clockwise in order to reduce the space occupied by the figure. As can be seen, the resulting corridor fits almost perfectly with the aerial view of the road, being the differences mainly caused by the image distortion and shift of the aerial pictures.

Two detailed views of the corridor are shown in Figs. 7b and 7c, corresponding to the areas marked by dotted-line rectangles in Fig. 7a. On these images, Bézier segments are represented by dotted lines and the corridor boundaries by the continuous black and grey lines. Those sections of the Bézier segments used as defined for the corridor are marked by the black lines, while the grey ones are the auto-generated sections joining discontinuous segments.

For both views, it can be appreciated the joining sections are successfully generated in all cases, maintaining the continuity of the corridor and its tangent. Indeed, Fig. 7d, shows the evolution of corridor tangent along all the planned path,

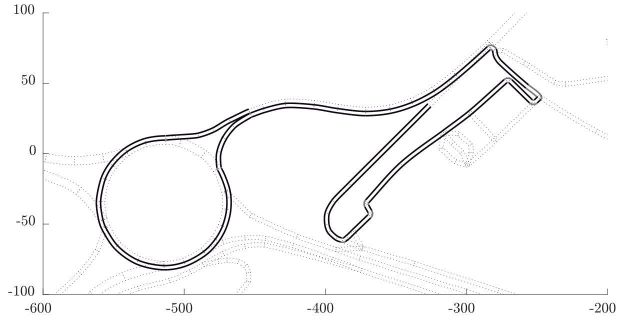

FIGURE 8. Overview of the navigation corridor for second route.

which being continuous guarantees the G^1 continuity of the corridor.

Figs. 8 and 9 show an overview of the navigation corridors obtained for the second and third route. On the second case, the planned path starts at the entrance of a wide roundabout and goes through a small parking lot. On the other hand, the third route goes back and forth along a peri-urban road. For both scenarios, it can be appreciated the navigation corridor is successfully generated, coping with the singularities of each path such as sharp junctions and three-way intersections.

Finally, for the longest two routes (cases 1 & 3), the generated Bézier corridors were compared with paths calculated from raw OSM and real GNSS traces. Due to the slight difference in location found between GNSS and OSM, paths are compared in terms of tangent evolution at the centrelines. Both orientation and difference of tangent vectors along the route are important indicators to take into account when considering a path as input for autonomous navigation, since they directly reflect the path smoothness and influence the control of the vehicle.

The OSM paths were extracted directly from map definition, displacing the points towards the right lane as it is done for Bézier adjustment (section V). GNSS traces were recorded using a receiver installed on a Citron DS3 automated vehicle with Real Time Kinematics capabilities, having an accuracy up to 2 cm in position resolution. On each case, three different drivers drove along the planned routes three time each for a total of 9 passes.

Table 2 summarizes the results for each route. Tangent values were calculated and compared at the closest points to the OSM nodes. The quality of the generated paths is quantified by the Mean Average Error (MAE) and the Mean Square Error (MSE) in regard to the tangent orientation calculated from vehicle traces. As can be seen, in both cases

TABLE 2. Comparison of tangent evolution along the planned routes.

Route		Tangent orientation		Tangent difference	
		MAE	MSE	MEAN	STD
1	GNSS Trace	N/A	N/A	1.537e-02	1.219e-02
	OSM	9.289e-02	1.538e-02	1.842e-01	1.316e-01
	Bézier	4.182e-02	5.616e-03	3.104e-02	6.317e-02
3	GNSS Trace	N/A	N/A	8.883e-03	1.150e-02
	OSM	5.549e-02	7.281e-03	1.158e-01	1.355e-01
	Bézier	2.118e-02	2.044e-03	2.214e-02	5.839e-02

FIGURE 9. Overview of the navigation corridor for third route.

the paths extracted from Bézier corridors have lower errors than the OSM paths. Likewise, the statistical analysis of tangent differences shows the evolution of tangent vectors along vehicles traces is more similar to the evolution found on Bézier paths than to the one observed on OSM paths.

VIII. CONCLUSION AND FUTURE WORKS

This paper proposes an automated procedure for generating navigation corridors using OpenStreetMap. To that end, the underlying algorithm first analyses the raw map information and deduces the geometric shape of the roads defined by adjacent nodes. Thereafter, a polynomial-based map representation is generated using third order Bézier curves. Finally, the adjusted map is combined with a third party route planner for defining the boundaries of the navigable space along the route.

As was shown in results, when compared to the aerial view of the area the proposed map representation is visually more accurate than the original OSM map based on straight segments. This improvement is due two main reasons: (i) by using third order Bézier curves two degrees of freedom are added to segment definition, allowing to better fit the road shape and (ii) the endpoint adjustment made on the narrow segments corrects the displacement error inducted by the difference in road width, joining the segments on the right lanes.

Once the polynomial map has been created, the definition of the navigable corridor is almost straightforward. Thus, the problem is reduced to the concatenation of segments along a route, automatically generating joining sections when necessary.

The viability of this algorithm has been validated with an experimental offline implementation, but future activities will extend this work by creating a C++ API to be integrated on the control architecture of prototype vehicles and used for trajectory planning. In this connection, the algorithm will be extended for taking into consideration on-board sensors information in the corridor generation stage. This, the vehicle will be able to provide feedback to the map, adjusting road segments position and curvature using real-time vision-based lane detection systems. In addition to that, the estimated uncertainty from GNSS-based localization will be used to assign probabilistic limits to the driving corridors. These two features will allow low-fidelity maps structure to be used in autonomous driving systems and reduce therefore the dependency on HD maps that most intelligent cars have to navigate.

REFERENCES

[1] K. Jo and M. Sunwoo, "Generation of a precise roadway map for autonomous cars," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 3, pp. 925–937, Jun. 2014.

[2] G.-P. Gwon, W.-S. Hur, S.-W. Kim, and S.-W. Seo, "Generation of a precise and efficient lane-level road map for intelligent vehicle systems," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 4517–4533, Jun. 2016.

[3] B. Williams, *Intelligent Transport Systems Standards*. Norwood, MA, USA: Artech House, 2008.

[4] *Input for ISO Physical Storage Format*, KW Consortium, Japan, Mar. 2000.

[5] S. Durekovic and N. Smith, "Architectures of map-supported ADAS," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 207–211.

[6] E. Casapietra, T. H. Weisswange, C. Goerick, and F. Kummert, "Enriching a spatial road representation with lanes and driving directions," in *Proc. 19th IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1579–1585.

[7] C. Guo, K. Kidono, J. Meguro, Y. Kojima, M. Ogawa, and T. Naito, "A low-cost solution for automatic lane-level map generation using conventional in-car sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2355–2366, Aug. 2016.

[8] F. Ramm, J. Topf, and S. Chilton, *OpenStreetMap: Using and Enhancing the Free Map of the World*. Cambridge, MA, USA: UIT Cambridge, 2011.

[9] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2014, pp. 420–425.

[10] J. Liu, B. Cai, Y. Wang, and J. Wang, "Generating enhanced intersection maps for lane level vehicle positioning based applications," *Procedia-Social Behav. Sci.*, vol. 96, pp. 2395–2403, Nov. 2013.

[11] C. Guo, J.-I. Meguro, Y. Kojima, and T. Naito, "Automatic lane-level map generation for advanced driver assistance systems using low-cost sensors," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 3975–3982.

[12] J. Rabe, M. Meinke, M. Necker, and C. Stiller, "Lane-level map-matching based on optimization," in *Proc. 19th IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1155–1160.

[13] C. Liu, K. Jiang, D. Yang, and Z. Xiao, "Design of a multi-layer lane-level map for vehicle route planning," in *Proc. MATEC Web Conf.*, vol. 124, 2017, p. 03001.

[14] S. Brummer, F. Janda, G. Maier, and A. Schindler, "Evaluation of a mapping strategy based on smooth arc splines for different road types," in *Proc. 16th IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 160–165.

[15] L. Liu, T. Wu, Y. Fang, T. Hu, and J. Song, "A smart map representation for autonomous vehicle navigation," in *Proc. 12th Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD)*, Aug. 2015, pp. 2308–2313.

[16] Y. Xu, V. Sasse, and K. Harms, "The European digital road map mutimap and its applications," *Int. Arch. Photogram. Remote Sens.*, vol. 31, pp. 982–987, Mar. 1996.

[17] P. Czerwionka, M. Wang, and F. Wiesel, "Optimized route network graph as map reference for autonomous cars operating on german autobahn," in *Proc. 5th Int. Conf. Autom., Robot. Appl. (ICARA)*, Dec. 2011, pp. 78–83.

[18] M. Dupuis, "Opendrive format specification," VIRES Simul. Technol., Bad Aibling, Germany, Tech. Rep., Aug. 2010.

[19] D. Bétaille and R. Toledo-Moreo, "Creating enhanced maps for lane-level vehicle navigation," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 4, pp. 786–798, Dec. 2010.

[20] J. Villagra, V. Milanés, J. Pérez, and J. Godoy, "Smooth path and speed planning for an automated public transport vehicle," *Robot. Auto. Syst.*, vol. 60, no. 2, pp. 252–265, 2012.

[21] D. J. Walton and D. S. Meek, "A controlled clothoid spline," *Comput. Graph.*, vol. 29, no. 3, pp. 353–363, 2005.

[22] G. Koutaki, K. Uchimura, and Z. Hu, "Network active shape model for updating road map from aerial images," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2006, pp. 325–330.

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

[24] E. Héry, S. Masi, P. Xu, and P. Bonnifait, "Map-based curvilinear coordinates for autonomous vehicles," in *Proc. 20th IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2017, pp. 1–7.

[25] L. 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," *Comput.-Aided Des.*, vol. 33, no. 14, pp. 1049–1058, 2001.

[26] D. S. Meek and D. J. Walton, "An arc spline approximation to a clothoid," *J. Comput. Appl. Math.*, vol. 170, no. 1, pp. 59–77, 2004.

[27] A. Artuñedo, J. Godoy, and J. Villagra, "Smooth path planning for urban autonomous driving using OpenStreetMaps," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 837–842.

[28] J.-W. Choi, R. E. Curry, and G. H. Elkaim, "Continuous curvature path generation based on Bézier curves for autonomous vehicles," *Int. J. Appl. Math.*, vol. 40, no. 2, pp. 1–12, 2010.

[29] L. Han, H. Yashiro, H. T. N. Nejad, Q. H. Do, and S. Mita, "Bezier curve based path planning for autonomous vehicle in urban environment," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2010, pp. 1036–1042.

[30] A. Artuñedo, J. Godoy, and J. Villagra, "A primitive comparison for traffic-free path planning," *IEEE Access*, vol. 6, pp. 28801–28817, 2018.

[31] D. Luxen and C. Vetter, "Real-time routing with OpenStreetMap data," in *Proc. 19th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst. (GIS)*, 2011, pp. 513–516. [Online]. Available: http://doi.acm.org/10.1145/2093973.2094062

[32] M. Kamermans. (Dec. 2016). *A Primer on Bézier Curves*. [Online]. Available: https://pomax.github.io/bezierinfo/

JORGE GODOY was born in Maracay, Venezuela, in 1986. He received the degree in electronic engineering from the Universidad Simón Bolívar, in 2008, and the M.E. and Ph.D. degrees in automation and robotics from the Universidad Politécnica de Madrid, in 2011 and 2013, respectively. In 2009, he was granted with a Predoctoral JAE Fellowship from CSIC for researching on autonomous vehicles at the Centre of Automation and Robotics UPM-CSIC. From 2013 to 2017, he was the Technical Coordinator of the AUTOPÍA Program, funded by research contracts from National and European research projects. In 2017, he was granted with a Juan de la Cierva Fellowship for Postdoctoral Research at the Universidad Politécnica de Madrid. His research interests include intelligent transportation systems, autonomous driving, path planning, and embedded AI-based control for autonomous vehicles.

ANTONIO ARTUÑEDO received the B.Sc. degree in electrical engineering from the Universidad de Castilla–La Mancha, Spain, in 2011, and the M.Sc. degree in industrial engineering from the Universidad Carlos III de Madrid, in 2014. He is currently pursuing the Ph.D. degree in automation and robotics with the Centre for Automation and Robotics UPM-CSIC, Spain, in the AUTOPÍA Program. His research interests include system modeling and simulation, intelligent control, and planning and decision making in autonomous vehicles.

JORGE VILLAGRA graduated in industrial engineering from the Universidad Politécnica of Madrid, in 2002. He received the Ph.D. degree in real-time computer science, robotics and automatic control from the École des Mines de Paris, France, in 2006. From 2007 to 2009, he was a Visiting Professor with the University Carlos III de Madrid, Spain. From 2013 to 2016, he led the Department of ADAS and Highly Automated Driving Systems, Ixion Industry & Aerospace SL, where he also coordinated all the activities in the EU R&D Funding Programmes. He has been leading the AUTOPÍA Program at the Centre for Automation and Robotics UPM-CSIC, since 2016. He then received a three year JAEDoc Fellowship at the AUTOPÍA Program, Center for Automation and Robotics (UPM-CSIC), Spain, where he spent one additional year funded by a research contract. He was first granted with a three years CIFRE Program in PSA-Peugeot-Citroën and then with a Postdoctoral Fellowship at a Joint Research Unit, INRIA-Mines ParisTech, France. He has developed his research activity in six different entities with a very intense activity in project setup and management, through over 30 international and national R&D projects, where he is or has been IP of ten of these projects. He has published over 85 papers in international journals and conferences on autonomous driving, intelligent transportation systems, dmodel-free control, and new probabilistic approaches for embedded components in autonomous vehicles. The results of the Ph.D. were granted with the Prize for the Best dissertation in automatic control in France, in 2006.