MŰEGYETEM 1782

Budapest University of Technology and Economics
Faculty of Civil Engineering
Department of Photogrammetry and Geoinformatics

Geoinformatics support for autonomous vehicles

PhD Thesis

Vivien Potó

Land Surveying and Geoinformatical Engineer

Supervisor:

Dr. Árpád Barsi

Budapest, 2022

1 Introduction and objectives

Self-driving vehicles are equipped with various sensors (these are typically cameras, laser scanners, and radar), which monitor the vehicle environment in complement to the human, and react to events to a larger or smaller level, helping the vehicle to drive safely. In my PhD research, I aim to support research on self-driving vehicles using the tools of geospatial information technology.

The basis of vehicle positioning and navigation is the map, which, when interpreted by the self-driving car, considers different data and features than traditional maps. For example, the machine has no need for environmental representation, which is the main source of information for humans, but it has a greater need for map accuracy and detail. The maps used by self-driving vehicles, HD (High-Definition Map) maps, help with vehicle positioning and navigation, support the vehicle in performing driving tasks and contribute to safe driving. I aim to create high-definition map bases to support driving assistants in lane changing and parking.

For self-driving vehicles, knowing their position is key, and traditional satellite solutions do not provide sufficiently accurate and continuous data (for example, in urban environments, a significant number of satellites are hidden by tall buildings in narrow streets). Therefore, additional positioning methods are needed to redundantly determine the vehicle's position. Using the sensor data of the self-driving vehicle, the vehicle's position can be more accurately determined. My goal is to create an efficient cached environment model to determine the position of the vehicle.

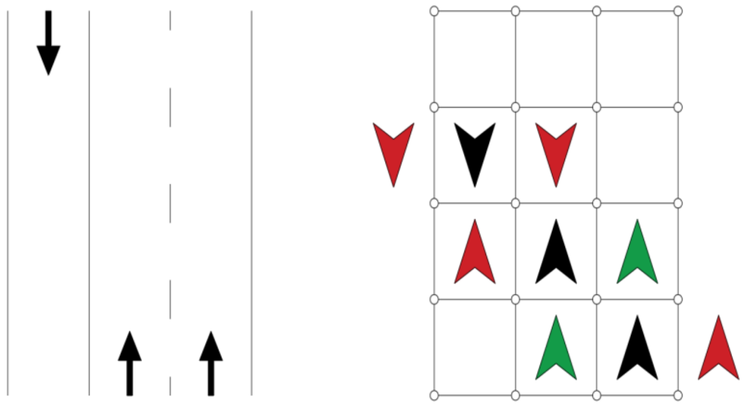
Simulations are an essential part of the vehicle testing process, helping to test innovations safely and cost-effectively. The OpenDRIVE model can be used to describe the geometry of roads and has good compatibility with various simulation software. My aim is to learn about the standard and then develop OpenDRIVE models from various types of ground data.

OpenCRG belongs to the standard family of OpenDRIVE. The model describes the road surface at high resolution in a curved grid structure, where cells usually contain height values. Like OpenDRIVE, the model is well applicable in simulation software to support vehicle dynamics and tire testing. My goal is to produce a road surface model that can represent the road from real measurement data in simulation software.

2 Map support for self-driving

At Tököli airport near Budapest, measurements were taken with a terrestrial laser scanner and a GNSS receiver. After processing the measurements, I created a finite element band model of the sample area, which was used to create the map model. My goal is twofold; the map shows the current position and direction of the vehicle and the possibility of changing lanes.

The topology of the model is based on a finite element mesh (FEM). The map lanes are divided into convex rectangular mesh elements. It is known in which lane and in which part of the lane the vehicle is located. Its position is projected to the grid model, and the location of the vehicle in the grid model can be determined by logical tests based on the principle of the minimum closing rectangle. Once the correct cell is found, the program also gives feedback on the possibility of changing lanes based on the adjacencies. Figure 1 shows an example of how the model works. In sub-plot a, the lane splitting of the road is shown with the driving lane indicated by arrows. In sub-plot b, three examples are shown, one below the other, with three possible positions of the vehicle indicated by the black arrows. The possibility of changing lanes to the right or left is indicated by color codes: green arrows indicate that a lane change is possible in the given direction, and red arrows do not.



a) Road lane splitting in the direction of travel with direction sign

b) Possible lane changes (black: vehicle position, green: lane change possible, red: lane change not possible)

Fig. 1: Demonstration of lane change support

On the campus of the Budapest University of Technology and Economics, I created a map that forms the basis of an occupancy grid to support vehicle parking. We performed a terrestrial laser scanning of the area, supplemented by GNSS measurements. With the help of students, a vector map of the area was created, which after topological verification, was used as input for the program code that generates the occupancy grid. As a result of the written procedure, a decimeter resolution occupancy grid of the area was produced (Figure 2). In the figure, yellow indicates the road surface accessible to vehicles, green indicates the potential parking area, red indicates the impassable surfaces, and purple indicates the infill area. The generated occupancy grid was also used by BME KJK researchers for further studies (mainly trajectory planning).

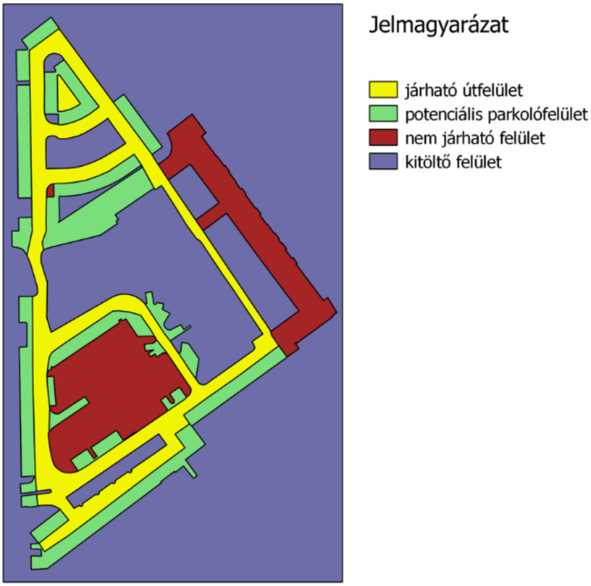


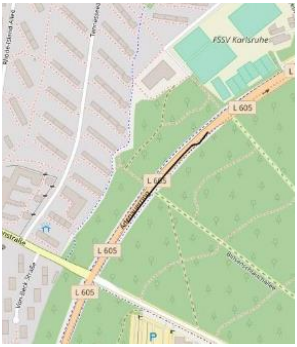
Fig. 2: Occupancy grid with a legend (yellow: accessible road surface, green: potential parking surface, red: non-accessible road surface, infill surface)

Thesis:
I created a high-definition map to support the operation of two self-driving vehicle assistants - lane-changing and parking - by identifying and storing detailed lane descriptor features based on survey data from a terrestrial laser scanner.

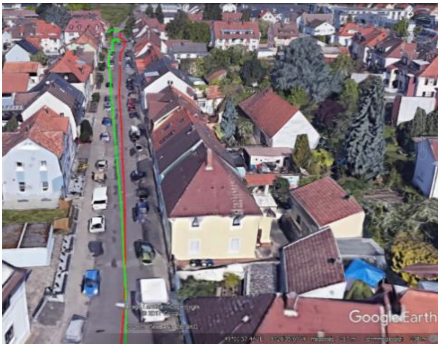
Related publications: [1] [2]

3 Creating an environmental model

Using visual odometry, the position and orientation of the vehicle are determined from camera images. Structure from Motion (SfM) technology is capable of determining the spatial location of surface points of spatial objects by processing images containing parallax due to motion to form two-dimensional image sequences. Interest operators can be used to extract the feature points of an image and link the images together. After alignment, the result is the images' external landmarks in the (local) coordinate system defined by the first image. The coordinates of the projection centers of the images represent the movement of the camera-carrying device, i.e, the trajectory of the vehicle. For the processing, I used the Pix4D software; for the validation, I used the results of GNSS measurements of the dataset and the visualization on different map bases. (Figure 3).



**a)
Dataset validation in an OpenStreetMap environment**



b) Dataset validation in Google Earth environment (result: green color, GNSS measurement: red color)

Fig. 3: Validated data sets

Localization landmarks are land features whose location is fixed and therefore provide a good reference, and they can be used for localization. As a starting data set, I chose a survey using the Leica Pegasus: Two mobile mapping system on the BME campus. With the assistance of students, the point cloud was prepared, and then localization landmarks were created based on the developed method. In the case of horizontal markers, a top view of a cross-section can be used to redraw the typically rectangular or circular object (Figure 4. a). In the case of vertical markers, a planar or three-dimensional marker can be generated by adding several horizontal section markers. (Figure 4. b)

a) Marking of manhole covers   b) Cylinder used as a traffic sign pole

Fig. 4: Using localization landmarks

The positioning can also be improved based on the same points in the environment detected by the vehicle's sensors and the environmental model. I developed an algorithm that generates a voxel model from mobile and terrestrial laser scanned point clouds. For efficient storage of the large-scale model, I improved the two-dimensional sparse matrix storage format of MATLAB. Various analyses were performed to determine the appropriate voxel size. The meter resolution voxel model produced from MLS measurements is shown in Figure 5.

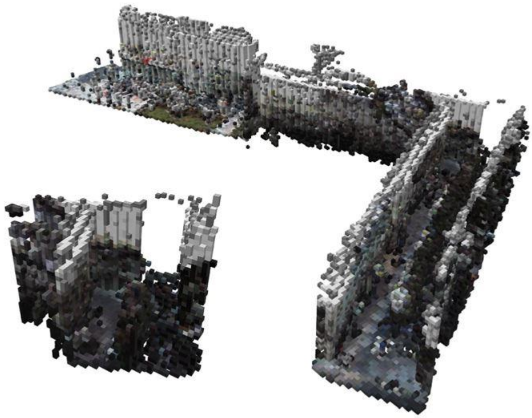


Fig. 5: MLS based voxel model for the whole sample area and a part of it

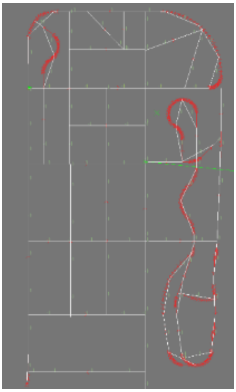
I used a finer resolution voxel model to demonstrate the support for positioning. I placed the vehicle in a small sample area of the model and used points in the city model, loaded with noise, to perform the location analysis. From the model, I produced a centered projection image showing the environment from the car's point of view. The coordinates of the image matrix in the figure represent the location of the object, and the value in it is the distance to the nearest object in that direction. This was then loaded with a normal distribution of ±30 cm noise to obtain the simulated initial data as sensed by the vehicle. Using a planar projection, I generated a space model with distance equations at the points to obtain the vehicle’s coordinates. Thanks to the simulated vehicle data, the obtained result can be easily verified. The generated city model is centimeter-accurate, the location model is decimeter-accurate, and the applicability of the model for location has been demonstrated.

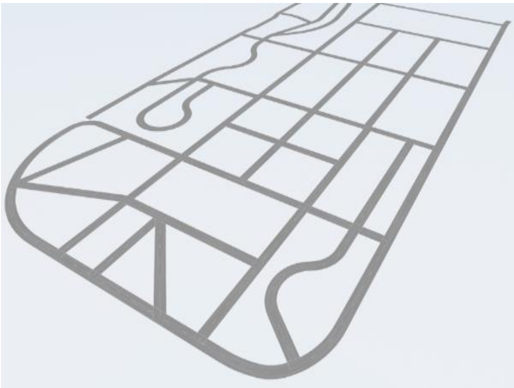
Thesis:
I developed a new voxel model for efficient storage of data obtained by mobile mapping and terrestrial laser scanning and demonstrated the applicability of the method in field tests on sample areas of different sizes. I demonstrated the accuracy of the model through an environmental model-based location task. I developed a visual odometry method based on SfM of in-vehicle camera images and verified the adequacy of my method in deriving vehicle trajectories on KITTI image sequences.

Related publications: [3] [4]

4 Creating OpenDRIVE model

We had the planning CAD files from the ZalaZone Hungarian test track, from the Smart City part of which I created the OpenDRIVE model of the area. After extracting the road axes, I performed a topological verification. After further conversions (-> dxf -> shp), the file was loaded into MATLAB software, which I used to create the OpenDRIVE model. After loading the model into the VIRES simulation software, some arcs were not plotted correctly (subfigure 6. a); after manual correction, I get subfigure 6. b.



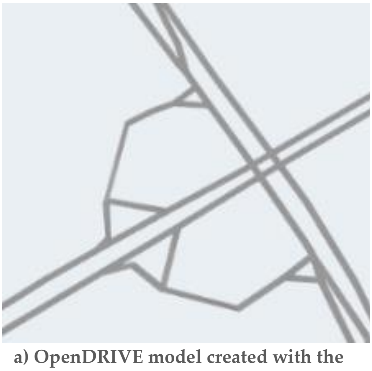


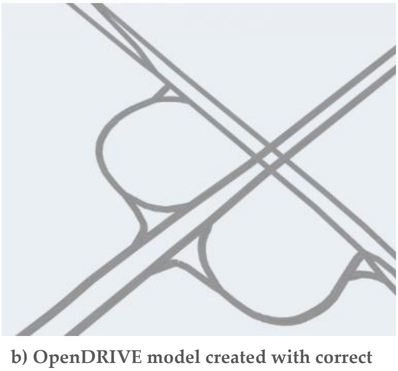
a) The incorrect state after loading

b) The correct state after repair

Fig. 6 Smart City part of the Zalaegerszeg test site in the VIRES environment

After downloading the OpenStreetMap data, the xml format was converted to shape format using tools developed by Geofabrik, which produced the OpenDRIVE model using the above process. This example illustrates the importance of parameter selection. The difference between sub-figures 7. a and 7. b can be seen. I also produced the OpenDRIVE model for a more complex geometric object using an OSM basis.





a) OpenDRIVE model created with the wrong parameterization

b) OpenDRIVE model created with correct parameterization

Fig. 7: OpenDRIVE model of the Buda bridgehead of the Petőfi Bridge in Budapest, generated from OpenStreetMap data in the VIRES environment

After the initial attempts, the next step is to produce a model that more accurately follows the road geometry, realistically representing the different widths of the road. To develop this, a road edge was generated from the point cloud of a survey on the BME campus using a mobile mapping system with the help of students. The road network is composed of straight lines and pure circular arcs, the road axis runs through the middle of the road sections, and the road lanes are shown as polygons. This makes the representation of the road more realistic, but the continuity of the road axes is not yet guaranteed. I displayed the model in OpenDRIVE ODR Viewer (Figure 8) and tested it in MATLAB Driving Scenario Designer, where the vehicle trajectory can be specified separately. Figure 9 also shows the simulated vehicle in the model with a blue rectangle.

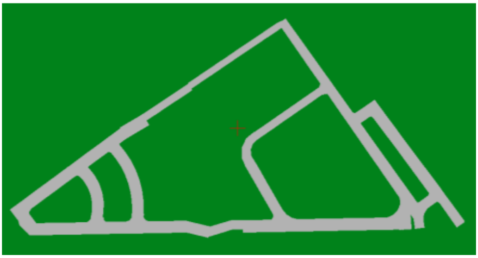


Fig. 8: Display of the MLS point cloud model in OpenDRIVE ODR Viewer

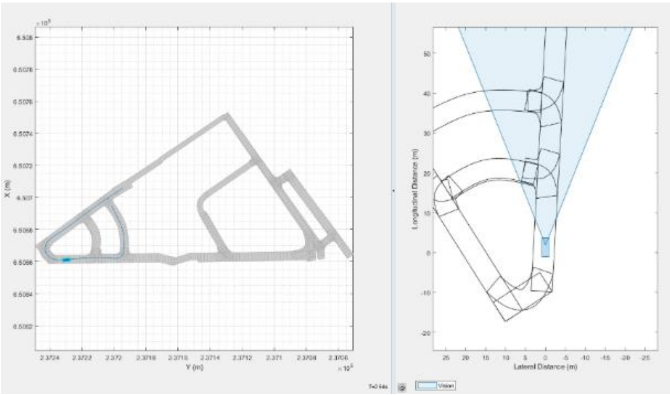


Fig. 9: Display of the model generated from the MLS point cloud in MATLAB Driving Scenario Designer

In the previous case, the representation of the models was sufficiently detailed, but the geometric connection was not realized in the lanes. I downloaded St. Gellert Square and its surroundings from the Budapest 11th district spatial database. With a team of students, the file was manually over-digitized in RoadRunner, which allowed to connect the lane polygons and to choose the road markings (Figure 10).

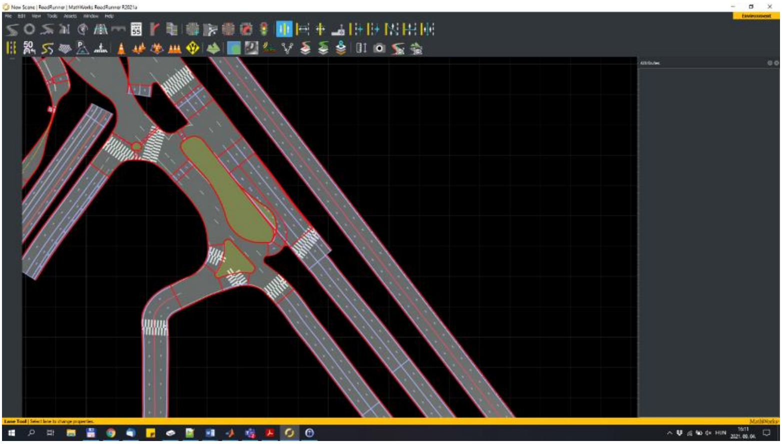


Fig. 10: OpenDRIVE model created from orthophoto in RoadRunner software Tézis:

Thesis:
I developed a methodology for processing data from four types of field surveys to upload a standard OpenDRIVE file format containing input environment descriptions for simulators used in automotive development. These data sources are CAD model, OpenStreetMap dataset, mobile laser scanning, and orthophoto. By properly processing the data sources, I have provided the description of the environment with the tools required by the standard. The resulting environment descriptions were tested in simulation environments, and their usability was verified.

Related publications: [5] [6]

5 Creating an OpenCRG model

Using MATLAB software, I wrote a procedure to produce a CRG grid of any resolution for the sample area. The input parameters of the function are the point cloud, the resolution of the grid, and the coordinates of the road axis. The procedure performs the generation, parameterization, and data loading of the

CRG grid. I tested the procedure on several sample plots, checking that the road features are reflected in the models. Figure 11 shows the source data, a point cloud segment of the Csiky street detail in CloudCompare software. The generated CRG model is shown in Figure 12. The produced model also shows the road features, the road pattern, and the manhole and missing stone fragment indicated by ellipses.

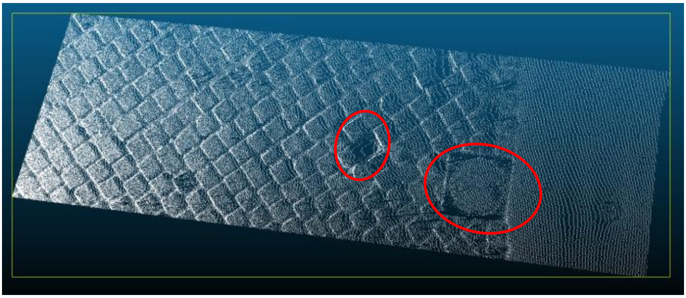


Fig. 11: Detail of the point cloud of Csiky street

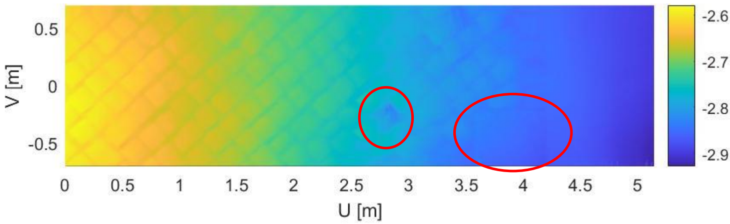


Fig. 12: 1 cm resolution CRG model of Csiky street

Thesis:
Instead of using synthetic (theoretical) data to produce a micro-level surface model of the road’s surface, I developed a practical methodology to generate a surface model using real field survey data and then produced standard OpenCRG models using point clouds from terrestrial and mobile laser scanning. Using the measured elevation data, I demonstrated that regular and irregular patterns of the road surface and its various irregularities can be represented by measurements from laser scanning, thus demonstrating the applicability of the standard description-based methodology to include and support the development of real surfaces.

Related publications: [7] [8]

6 Summary

My PhD research was aimed to support the development of self-driving vehicles. To support self-driving functions, I developed a map that gives the position of the vehicle and shows the possibility of changing lanes. I produced an occupancy grid that supports the vehicle by showing possible parking spaces during the parking process. Using a combination of visual odometry and SfM, localization landmarks can be used to pinpoint the vehicle's position based on its (sensor-sensitive) environment. By generating a road environment voxel model, the vehicle's environment can be generated, and efficiently stored, and the positioning performed using this model can help to refine the vehicle's position. I have created OpenDRIVE models from different input formats (CAD file, OSM, MLS point cloud, orthophoto) and demonstrated their applicability in several simulation software. I developed an algorithm to create an OpenCRG model with field data for different sample areas.

References

[1] Vivien Potó, József Árpád Somogyi, Tamás Lovas, Árpád Barsi, Viktor Tihanyi, and Zsolt Szalay, “Creating HD map for autonomous vehicles - a pilot study,” in *34th International Colloquium on Advanced Manufacturing and Repairing Technologies in Vehicle Industry*, 2017, no. January 2018, pp. 127–130.

[2] Árpád Barsi, Vivien Potó, Árpád Somogyi, Tamás Lovas, Viktor Tihanyi, and Z Szalay, “Supporting autonomous vehicles by creating HD maps,” *Prod. Eng. Arch.*, vol. 16, pp. 43–46, 2017, doi: https://doi.org/10.30657/pea.2017.16.09.

[3] Vivien Potó and Árpád Barsi, “Applying Structure-from-Motion technique for visual odometry,” in *AIS 2017 - 12th International Symposium on Applied Informatics and Related Areas*, 2017, pp. 145–149.

[4] Vivien Potó, József Árpád Somogyi, Tamás Lovas, and Árpád Barsi, “Laser scanned point clouds to support autonomous vehicles,” *Transp. Res. Procedia*, vol. 27, pp. 531–537, 2017, doi: 10.1016/j.trpro.2017.12.130.

[5] Vivien Potó, Nikol Krausz, János Máté Lógó, and Árpád Barsi, “OpenDRIVE Standard for Road Description,” in *AIS 2019 14th International Symposium on Applied Informatics and Related Areas organized in the frame of Hungarian Science Festival 2019 by Óbuda University Proceedings*, 2019, pp. 134–138.

[6] Árpád Barsi, Vivien Potó, János Máté Lógó, and Nikol Krausz, “Creating an OpenDRIVE Model of the Campus of the Budapest University of Technology and Economics for Automotive Simulations,” *Period. Polytech. Civ. Eng.*, vol. 64, no. 4, pp. 1269–1274, Sep. 2020, doi: 10.3311/PPci.16768.

[7] Árpád Barsi, Vivien Potó, and Viktor Tihanyi, “Creating OpenCRG Road Surface Model from Terrestrial Laser Scanning Data for Autonomous Vehicles,” in *Lecture Notes in Mechanical Engineering*, no. 9783319756769, 2018, pp. 361–369.

[8] Vivien Potó, András Csepinszky, and Árpád Barsi, “Representing road related laserscanned data in curved regular grid: A support to autonomous vehicles,” in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 2018, vol. 42, no. 2, pp. 917–921, doi: 10.5194/isprs-archives-XLII-2-917-2018.