WIP: Real-world 3D models derived from mobile mapping for ray launching based propagation loss modeling

Tobias Wahl, Dorit Borrmann,
Michael Bleier, Andreas Nüchter
Robotics and Telematics
Julius Maximilian University of Würzburg
Würzburg, Germany
correspondence: `andreas@nuechti.de`

Thomas Wiemann, Thomas Hänel,
Nils Aschenbruck
Institute of Computer Science
Osnabrück University
Osnabrück, Germany

***Abstract*—This work in progress paper presents an automated approach for network coverage prediction in real-world environments by combining mobile mapping, 3D mesh generation, and a ray launching based network simulator. We identify the challenges and demonstrate the functionality of such a pipeline. We preview an empirical evaluation in a realistic real-world environment.**

***Index Terms*—network simulator, 3D mobile mapping, 3D meshing, network coverage**

I. Introduction

For many years, 3D data processing has been an important aspect of robotics, automotive technology, surveying, and even of computer games. In this paper, we use it for predicting network coverage based on 3D real-world environment data. To this end, a 3D model of the environment is obtained from data acquired with a 3D scanning rig, based on multiple automotive grade LiDARs (Light detection and ranging oder Light imaging, detection and ranging). Using ray launching within this 3D model, the propagation loss model for a discrete-event network simulator is computed.

The entire toolchain consists of several open-source modules: ROS[1] nodes for data acquisition, *3DTK – The 3D Toolkit*[2] for calibration and registration of 3D point cloud data. In the next step, this data is used in the *Las Vegas Surface Reconstruction Toolkit*[3] to create a full-size 3D-mesh of the environment. In the last stage, *RaLaNS*[4] is used to predict network coverage. This scientific tool works with 3D meshes as it was initially built to cope with the lack of accurate simulation models to predict network coverage. We show the potential to get highly precise simulations when using accurate 3D-models at centimeter scale.

The contributions of this short paper are as follows: We show for the first time, that it is possible to predict network coverage on real-world data automatically. To this end, we present an end-to-end pipeline, cf. Figure 1.

The mobile mapping system was built in the project TASTSINN/VR, funded by the Federal Ministry of Education and Research due to an enactment of the German Bundestag under the grant 16SV8159.

[1] https://ros.org/

[2] http://threedtk.de/

[3] https://www.las-vegas.uni-osnabrueck.de/

[4] https://sys.cs.uos.de/ralans/index.shtml

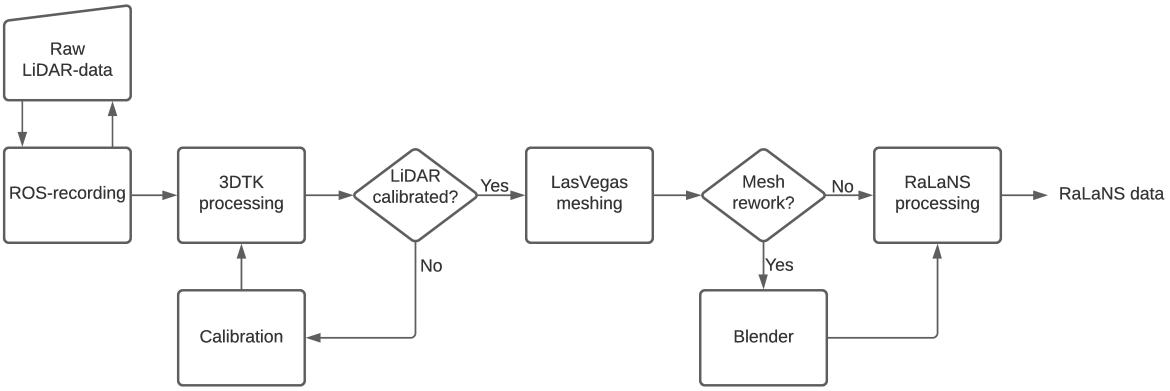

Fig. 1. Several open-source tools are combined for mapping, meshing, and ray launching based propagation loss modeling.

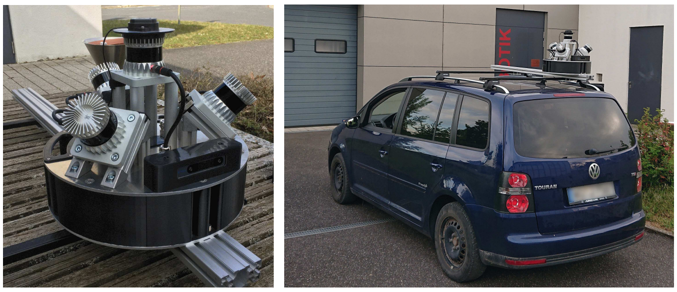

Fig. 2. Left: LiDAR scanning rig with four Ouster laser scanners. Right: Rig mounted on the car of the corresponding author.

Our work is motivated by the need for predicting network coverage in many applications. For example mobile exploration robots distribute network nodes to ensure communication with a base station. One application area that relies on such an extended communication is the exploration of lunar caves, as we recently proposed [1]. Other examples include the communication in underground mines and in factory environments.

II. Mobile Mapping to obtain 3D Point Cloud Data

Laser scanning provides an efficient way to actively acquire accurate and dense 3D point clouds of environments. Mobile scanning is currently used for modeling in architecture, urban and regional planning. Modern systems, like the RIEGL VMY-2 mobile scanning solution or the Optech Lynx Mobile Mapper, work along the same basic principle. They combine a highly accurate Global Navigation Satellite System (GNSS), a high-precision inertial measurement unit (IMU) and the odometry of the vehicle to compute fully timestamped trajectories. Using a process called motion compensation, this trajectory is then used to georeference the laser range measurements that were acquired by a set of 2D laser scanner also mounted on the vehicle. The quality of the resulting point cloud depends on several factors:

- The calibration of the entire system, i.e., the accuracy to which the position and orientation of each individual sensor in relation to the vehicle has been determined.
- The accuracy of the external positioning sensors, i.e., the GNSS, IMU and odometry.
- The accuracy of the laser scanner itself.

As current mobile mapping system are precise, but also very expensive, we have constructed our own mobile mapping system on the basis of four Ouster laser range finders, that are typically used in autonomous driving research. In addition, we have combined a low-end IMU, and an L1/L2 GPS, GLONASS, Galileo receiver (u-blox ZED F9) as shown in Figure 2. The whole scanning rig includes an Intel NUC and is constructed from item profiles, laser cutted aluminum, and 3D printed parts. Data recording and time stamping on the NUC was done with ROS [2].

Calibration is an important issue. To determine the pose, i.e., the geometric position and orientation of each 3D scanner on the rig, we have acquired a few high-precise 3D scans on our campus with a RIEGL VZ400 survey grade laser scanner. Afterwards, we have used the well-known ICP algorithm [3] from 3DTK to match a laser scan of a scanner of the rig into this precise reference scan.

After mounting the system on a car, data was acquired on a path around the computer science building at the University of Würzburg. The point cloud data of the four scanners was combined into a single scan using the calibrated poses. The combined scan was then fed into our SLAM framework, i.e., into 3DTK. We apply the ICP and globally consistent scan matching to create a precise 3D point cloud (cf. [4], [5]).

For evaluating the system, we have compared our obtained 3D point cloud with a reference point cloud, acquired with the survey grade laser scanner (cf. Figure 3, right).

III. From point clouds to 3D meshes

Polygonal environment representations show many benefits and automatic reconstruction of polygonal data has drawn a lot of attention in the robotics community. The polygonal map generation process consists of two steps: Initial mesh generation and mesh optimization. The initial surface reconstruction in Las Vegas is based on Marching Cubes with Hoppes signed distance function [6]. For the evaluation of the distance function within the Marching Cubes algorithm, point normals have to be estimated, which we do using the RANSAC

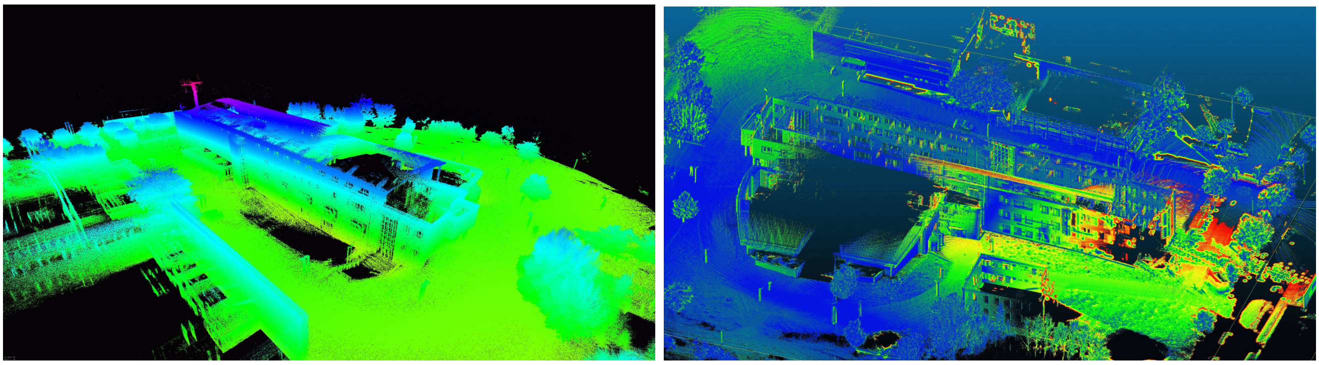

Fig. 3. Left: 3D point cloud obtained from mobile mapping. Right: Comparison with the reference 3D point cloud.

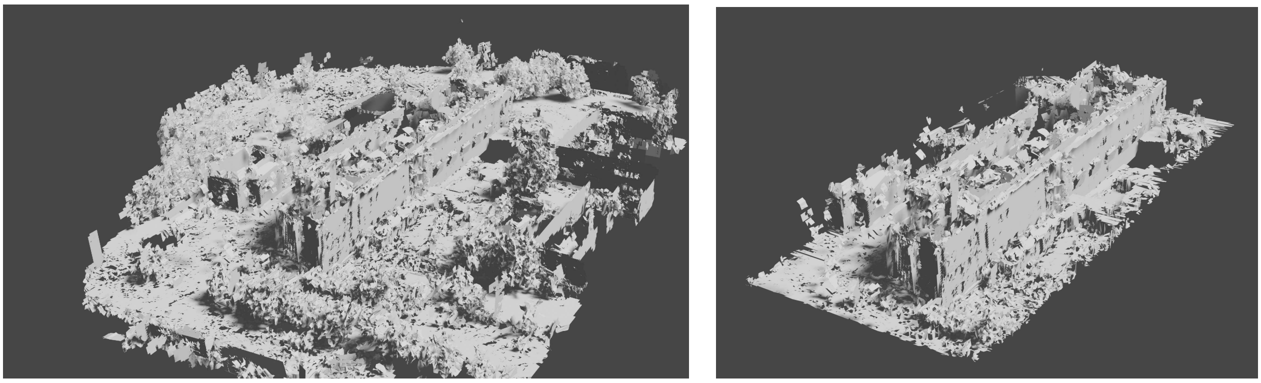

Fig. 4. Left: Generated mesh. Right: Postprocessed and cropped mesh.

algorithm [7] from a k-neighborhood, where we adopt the k automatically according to the local point distribution [8].

For our application, we decided to go for the Marching Tetraeder method and not for the Standard Marching Cubes. Usually, the Marching Cube implementation delivers good results, but the generated meshes can show holes even in dense data. The Marching Tetraeder version delivers more consistent results in such cases, but produces more triangles.

If the point density in the input data is high enough to reconstruct connected surfaces, a *planar clustering* algorithm is applied. This algorithm clusters connected planar regions within the meshes using a grass fire approach that compares the normals of adjacent triangles. The extracted cluster are then optimized by moving all vertices into a common local regression plane. This planar optimization is an efficient noise filter as it compensates fluctuations of the triangle vertices resulting from slightly different signed distance function values coming from sensor noise.

With the optimized mesh from the Las Vegas toolkit we still obtain too many faces for further processing. Thus, we use the CGI-tool Blender to crop the scene to the part of interest, which is given in Figure 4, right.

IV. Application of a Ray Launching Based Propagation Loss Model for ns-3

RaLaNS, a propagation loss model for ns-3 that is based on ray launching, has been presented in [9] and [10]. RaLaNS is divided into two applications: The first contains the actual ray launching and generates a matrix containing the signal strengths for various combinations of transmitter and receiver positions. The resulting matrix is used in the second component which is a PropagationLossModel for ns-3.

For simulation, the rays are first launched radially symmetrical and an initial energy is equally distributed onto the rays. Then, the ray launcher determines for every ray where it hits an object or gets close to an edge. The direction and energy of the ray is adjusted depending on the effect that occurs. Whenever one of these events occurs, a new ray segment is generated. This is repeated for all the rays until a predefined number of iterations is reached. In this process, the path for each ray is generated and stored as a list of ray segments. The maximum list length is equal to the number of iterations. In a second step, a receiver represented by a sphere is placed in the world. All rays that hit the sphere are collected and the total energy is calculated. This is a simplification of the receiver antenna. Figure 5 bottom shows the result of the simulation with RaLaNS with the signal source used for the simulation at Point-ID 5. The obstruction of the radial signal spread by the building is apparent. The 15 positions marked in the image are used to evaluate the modeling results.

V. Results, Conclusion, and Future Work

To check whether the coverage model is close to the real environment reference measurements were made. A Netgear R7000 router with DD-WRT installed is used as a sender. To be as close as possible to the 2.4 GHz simulation the router uses channel 1 with channel width HT40 and transmission power of 100 mW. Figure 6 shows the setup of router and its position next to the building. We used channel 1 to be as close as possible to the 2.4 GHz simulation and a channel width HT40 and transmission power of 100 mW. Figure 6 shows the setup of router and the building.

For the measurement a Samsung Galaxy S4 (jfltexx) with LineageOS 17 was used. The measurements were made at

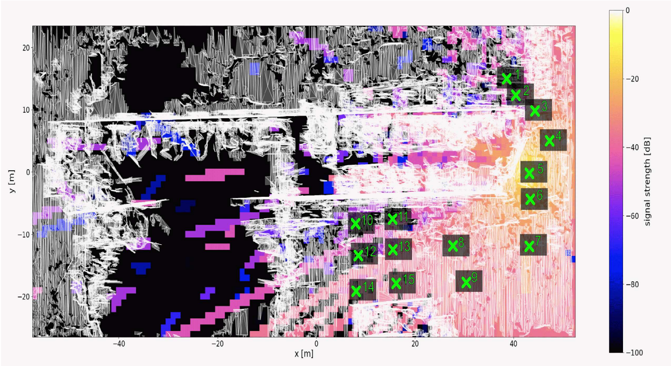

Fig. 5. Mesh processed in the RaLaNS tool in a bird's eye view and the location of the points evaluated in Figure 7. The sending router was positioned at Point-ID 5.

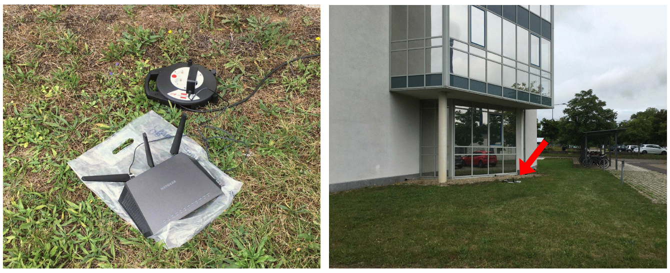

Fig. 6. Router used to perform validation measurements and its position next to the building.

15 positions mainly in the lower right where the edge of the building is expected to lead to diffraction.

The resulting values are presented in Figure 7. Points 1 and 12 were excluded from the statistical evaluation as the number of rays was too low to generate valid simulated values at these positions. The regression line being parallel to the central diagonal clearly indicates that the simulated values differ from the measured values mainly by an offset of 29.7 dB. As outlined in [9], this is the expected behavior because sender and receiver antenna gains are not explicitly simulated. After correcting for this offset, the measured values correlate well with the simulated values with an R^2 of 0.756 and a Root Mean Squared Error (RMSE) of 7.8 dB. Higher errors occur for the points 4, 5, and 6 close to the router which may be attributed to the variability in the near field of the router's antennas and to reflections from the wall because the different material properties were not considered in the simulation. Figure 6 shows that the router was positioned next to glass facades which could have caused these effects. Higher errors also occur for the points far away, especially 12 and 14. These may be attributed to inaccuracies of the model due to the limited number of rays which would lead to wrong diffraction. In the case of point 1, errors mainly stem from signal translucent walls or glass not being simulated well.

This paper presents a pipeline that includes mobile mapping, 3D modelling and network coverage estimation aiming at combining and automating these steps. We have shown a 3D model obtained from sensor data can form an input for a network simulator.

Needless to say, a lot of work remains to be done. In future work, we aim at elaborating the presented pipeline such that the data is processed automatically. Furthermore, similar experiments are planned for indoor environments, especially for factory environments.

While processing point cloud data into a mesh and then into a network simulator, we are wondering, if one could potentially skip the meshing part. Simulating networks in raw point clouds should be possible, given current progress in the area of 3D point cloud classification. With a point classification as *surface point*, *corner point*, or *edge points* ray launching simulations with reflection, diffraction, and scattering at building should be possible. Also the influence of reflective surfaces [11] could be analyzed and incorporated into the pipeline.

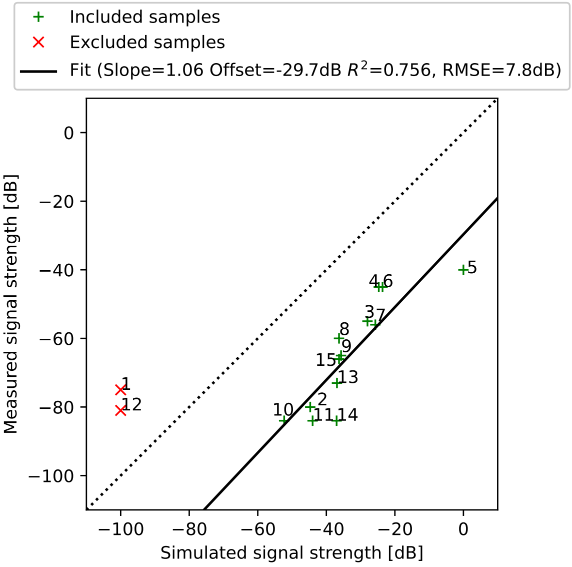

Fig. 7. Measured signal strength vs. simulated.

References

[1] A. P. Rossi, et al. "Daedalus – descent and exploration in deep autonomy of lava underground structures," Uni Wuerzburg Research Notes in Robotics and Telematics, Julius Maximilian University of Würzburg, March 2021.

[2] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA '09) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.

[3] P. Besl and N. McKay, "A method for Registration of 3-D Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, vol. 14, no. 2, pp. 239–256, 1992.

[4] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg, "Globally consistent 3d mapping with scan matching," *Journal Robotics and Autonomous Systems (JRAS)*, vol. 56, no. 2, pp. 130–142, 2008.

[5] J. Elseberg, D. Borrmann, and A. Nüchter, "Algorithmic solutions for computing accurate maximum likelihood 3D point clouds from mobile laser scanning platforms," *Remote Sensing*, vol. 5, no. 11, pp. 5871–5906, November 2013.

[6] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *Computer Graphics*, vol. 26, no. 2, pp. 71–78, 1992.

[7] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, pp. 381–395, 1981.

[8] T. Wiemann, H. Annuth, K. Lingemann, and J. Hertzberg, "An extended evaluation of open source surface reconstruction software for robotic applications," *Journal of Intelligent & Robotic Systems*, vol. 77, no. 1, pp. 149–170, 2015.

[9] T. Hänel, A. Bothe, and N. Aschenbruck, "Ralans – a ray launching based propagation loss model for ns-3," in *2015 International Conference and Workshops on Networked Systems (NetSys)*, 2015, pp. 1–7.

[10] T. Hänel, M. Schwamborn, A. Bothe, and N. Aschenbruck, "On the map accuracy required for network simulations based on ray launching," in *2015 IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM '15)*, 2015, pp. 1–8.

[11] R. Koch, S. May, P. Murmann, and A. Nüchter, "Identification of Transparent and Specular Reflective Material in Laser Scans to Discriminate Affected Measurements for Faultless Robotic SLAM," *Journal of Robotics and Autonomous Systems (JRAS)*, vol. 87, pp. 296–312, January 2017.