CEIG - Spanish Computer Graphics Conference (2022)
J. Posada and A. Serrano (Editors)

Guided modeling of natural scenarios: vegetation and terrain

José A. Collado[1], Alfonso López[1], J. Roberto. Jiménez-Pérez[1], Lidia M. Ortega[1], Francisco R. Feito[1], Juan M. Jurado [2]

[1]Department of Computer Science, University of Jaén
[1]Department of Software Engineering, University of Granada

Abstract

The generation of realistic natural scenarios is a longstanding and ongoing challenge in Computer Graphics. LiDAR (Laser Imaging Detection and Ranging) point clouds have been gaining interest for the representation and analysis of real-world scenarios. However, the output of these sensors is conditioned by several parameters, including, but not limited to, distance to scanning target, aperture angle, number of laser beams, as well as systematic and random errors for the acquisition process. Hence, LiDAR point clouds may present inaccuracies and low density, thus hardening their visualization. In this work, we propose reconstructing the surveyed environments to enhance the point cloud density and provide a 3D representation of the scenario. To this end, ground and vegetation layers are detected and parameterized to allow their reconstruction. As a result, point clouds of any required density can be modeled, as well as 3D realistic natural scenarios that may lead to procedural generation through their parameterization.

CCS Concepts

• ***Computing methodologies*** → ***Point-based models; Modeling methodologies;***

1. Introduction

The generation of the Earth digital twins is one of the greatest challenges to mankind. Over the last years, the growth of urban environments in parallel with the role of forests has received increased attention in many applications [MHS*19; ECC*21]. In Computer Graphics, the modeling and rendering of photorealistic scenarios has been a longstanding and ongoing challenge over time. For this purpose, 3D reconstruction techniques have been widely applied for the production of synthetic scenarios with the desired level of detail [KGG*20]. Similarly, other recent research has proposed easy-to-use platforms to facilitate the creation of real-world scenarios through the application of guided and procedural approaches [KGG*20; GGP*19].

On the other hand, the advances in high-resolution cameras and Laser Imaging Detection and Ranging (LiDAR) devices allow us to obtain both natural and artificial models of real-world environments. Therefore, 3D point clouds are generated for the representation of both urban and natural scenarios. Many applications ranging from Earth observation and inventory [COB*18] to visual computing benefit from scanned data to have an accurate understanding of the surrounding environment. Nevertheless, open point cloud datasets may present a low spatial resolution or even occluded areas due to the limitation of the sensor and acquisition conditions.

In this work, we present the first steps for the automatic generation of plausible large-scale natural environments using available point clouds from LiDAR datasets. The realism of the resulting scenario is achieved by following spatial and semantic constraints according to the scanned data. Besides 3D reconstruction, our method enables the procedural-based extension of the boundaries of LiDAR surveys. Moreover, it allows refining the density of sparse LiDAR point clouds.

2. Related work

Previous work has been proposed to generate virtual environments with realistic traits of real landscapes [STBB14]. On the one hand, urban structures are often modeled by procedural approaches based on L-systems. Initially, procedural models of large-scale cities were introduced by Merell and Manocha [MM08]. Other studies used inverse procedural modeling that learns from real-world data and attempts to transfer it to synthetic ones by fitting parameters of procedural models. More recently, trained deep neural networks have been combined with inverse procedural modeling to generate entire real layouts [NPA*22]. Likewise, there is also research focused on plant modeling based on fractals [Opp86], rules and sketches. Alternative approaches attempt to reconstruct plant models automatically either from images, videos [LDS*11], or scanned 3D point clouds [COB*18]. More recent work also focus on dynamic and realistic behaviour of plant models, including growth and the interaction with natural phenomena [NPA*22; PNH*14]. Here we focus on the involved aspects of plants and modeling in natural ecosystems. In any case, generating plausible vegetation and ter-

DOI: 10.2312/ceig.20221144

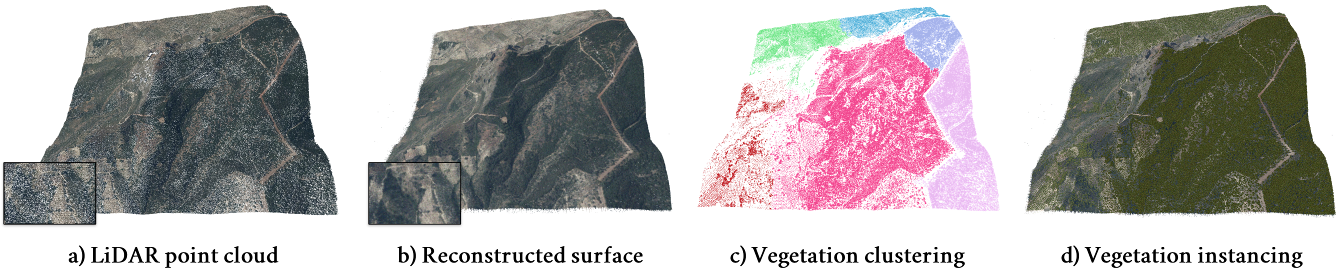

Figure 1: *Comparison of original LiDAR point cloud (a) and our reconstructed point cloud (b). (c) Result of clustering high vegetation and, (d) the reconstructed scenario including surface and high vegetation [CLJ*22].*

rain models for virtual landscapes faces two major challenges: first, plant placement varies across different characterization of surveyed zones and second, generating a real terrain including elements with specific shapes such as roads [ACV*14] and rivers. To address these challenges, we propose some steps for the guided procedural modeling of natural scenarios by using LiDAR point clouds.

Regarding inverse procedural methods, natural environments can be reconstructed on the basis of real-world data sources, such as point clouds (e.g., photogrammetric or LiDAR), Digital Terrain Models (DTMs) and aerial imagery. Accordingly, Andujar et al. [ACV*14] reconstructed roadside landscapes using the three mentioned data sources. The ground mesh is approached using a DTM enhanced with reconstructed point clouds and fractal algorithms. On the other hand, trees are instanced according to aerial imagery by solely providing a visually consistent distribution, while handling the distance among instances. Also, LiDAR point clouds has been used to reconstruct ground and forest layers [COB*18; QGY18]. Furthermore, the visualization of scenarios with dense vegetation is also a key factor to leverage performance and realism. Hence, the rendering is traditionally accelerated using application programming interfaces, such as OpenGL. Geometry and tessellation shaders have been previously used to control density and level of detail (LOD) of vegetation meshes [ACV*14]. However, scenarios may also be reconstructed as point clouds for point-based applications. In this regard, the efficient rendering of point clouds has been approached through modern extensions of OpenGL's compute shaders [SKW21].

Our study proposes an hybrid solution to enhance LiDAR point clouds and generate highly detailed representations of scenarios including high vegetation. Regarding rendering techniques, our work uses compute shaders to visualize more detailed scenarios composed of a large number of points. The paper is organized as follows: Section 3 details our proposal including the generation of the terrain and the high vegetation, as well as the applied rendering techniques, Section 4 presents the results and, finally, Section 5 concludes the paper.

3. Our method

Our approach is based on a guided procedural modeling of real-world point clouds using scanned data in order to generate synthetic scenarios in natural environments. To this end, open LiDAR data is used to determine spatial and semantic constraints. In this study, we focus on vegetation and ground layers that are identified from semantic labels in LAS (LASer) files. Figure 2 and the following three subsections presents the main steps of the proposed methodology.

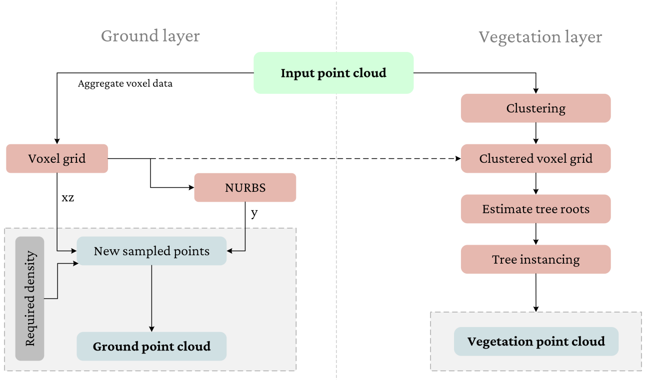

Figure 2: *Overview of the proposed method to process ground and vegetation layers.*

3.1. Ground modeling

As a first step, the ground layer is uniformly split using a regular grid. The voxel dimensions are either defined as the Ground Sampling Distance (GSD) of the point cloud or selected by the user. For each voxel, color and elevation data are aggregated, though other significant data may also be included. Next, a non-uniform rational B-spline (NURBS) surface is automatically built using the prior voxel discretization. Control points are defined by a representative point of each voxel, using the aggregated height. In order to resample the reconstructed ground once the NURBS is created, we use a spatial probability distribution function for each voxel so that better-filled spaces present less probability to generate new points. Hence, voxels are populated with new points as long as the target density is not achieved. This approach guarantees to fill both holes and sparse areas. The color of new points is defined using a bilinear interpolation with neighbouring voxels, although minor randomness is also considered to simulate color noise from natural scenarios. Besides *XZ* coordinates, which are uniformly distributed, the elevation is given by the NURBS. As a result, we have a representation of the ground as a 3D spline (Figure 3) that allows both to build the whole terrain layer as a triangle mesh or increase the original point cloud density with a user-defined rate.

Figure 3: *Control points of the NURBS.*

3.2. Vegetation modeling

Points labeled as high vegetation are then processed to reconstruct forestry areas. To this end, these points are clustered to differentiate areas of different tree specimens. We resolved this using the region growing clustering method based on color from Point Cloud Library (PCL) (Figure 4).

Once the clusters are built, a regular grid is generated for each one. In order to reconstruct the high vegetation layer, we use some predefined tree point clouds that match the specimens living in the study area. To instance trees, we exploit the knowledge of LiDAR scans. Although LiDAR is capable of acquiring both ground and vegetation layers, voxels concerning the latter class present higher density than ground-labeled voxels. Thus, the vertical position (y) of each tree can be either computed from the NURBS surface, by sampling xz coordinates, or using the mean voxel height. On the other hand, xz is determined by the voxel centre. The overlapping of tree instances is avoided considering the bounding box of previously instanced trees, thus preventing the addition of new models in neighbouring voxels that are already covered. Also, the overall number of tree instances depends on the previous voxelization of the terrain (Figure 5).

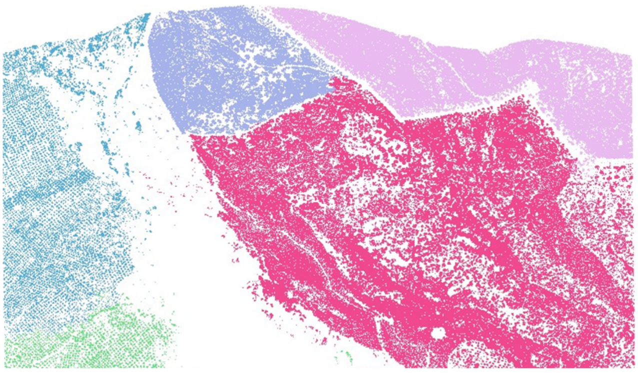

Figure 4: *Clustering of a LiDAR point cloud according to the canopy color, aimed at differentiating tree specimens.*

Figure 5: *Comparison of vegetation instancing, using grids of a) 500x500 and b) 1100x1100 subdivisions.*

3.3. Rendering improvements

Although this work is focused on the density enhancement of point clouds, the rendering methodology can be modified to achieve better performance and quality of the resulting image, as described by Schütz et al. [SKW21]. The first step is based on sorting points according to their Morton codes, which are subsequently shuffled in batches to prevent unbalanced processing in the GPU threads. Then, the point cloud rendering is implemented in compute shaders that aggregate colors from points projected in the same pixel. To avoid noisy renderings, as a consequence of point cloud errors and variable depth, several points are considered per pixel using a depth interval instead of solely taking into account the first visible one. With these improvements, we can render up to 200+ millions points in real-time using an NVIDIA RTX 2070 Super.

4. Results

The proposed methodology is implemented in C++17 along with OpenGL (Open Graphics Library). Most of the processing work is parallelized in CPU using OpenMP, whereas the rendering algorithms have been developed in GLSL with OpenGL's Compute Shaders. Our approach has been evaluated over publicly available point clouds datasets from PNOA. This repository covers the Spanish country with subdivisions of two squared kilometers, with density of 0.5 points/m^2 and altimetry precision of 20cm. Also, point clouds are segmented in different layers according to the LAS standard. In this work, we focus on an archaeological site located in Campillo de Arenas (Spain) since it is composed of several tree specimens (e.g., olive groves and pine) and presents a steep surface. Response time measurements were performed on a PC with AMD Ryzen 5 1600 3.2 GHz, 16 GB RAM, GTX 1060 GPU with 6 GB VRAM (Pascal architecture) and Windows 10 OS.

To assess the performance of the proposed methodology, we have evaluated both the resulting point clouds and the response time of the pipeline. First, we generate point clouds with increased point density. Figure 6 compares the input and the resulting point cloud with x100 more ground density. The first image barely shows tree canopy returns, whereas our solution instance tree models and sample them to compose a dense point cloud concerning the vegetation layer. Also, our result fills the ground areas not reached by LiDAR due to dense vegetation, while also increasing the overall ground density. Figure 7 shows this comparison using an aerial point of view.

a)

b)

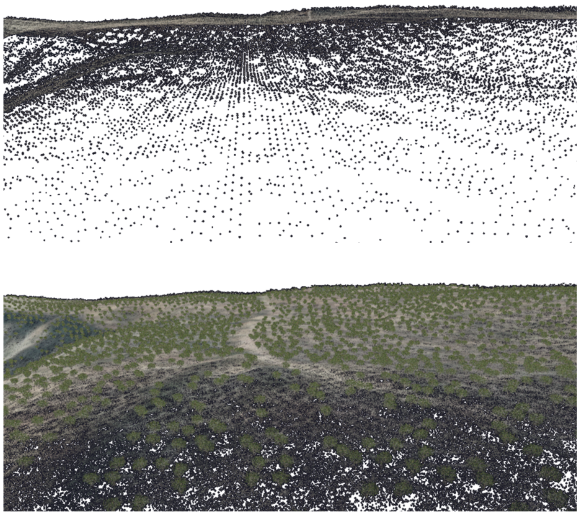

Figure 6: *a) Sparse LiDAR point cloud from an open dataset and b) the reconstruction result.*

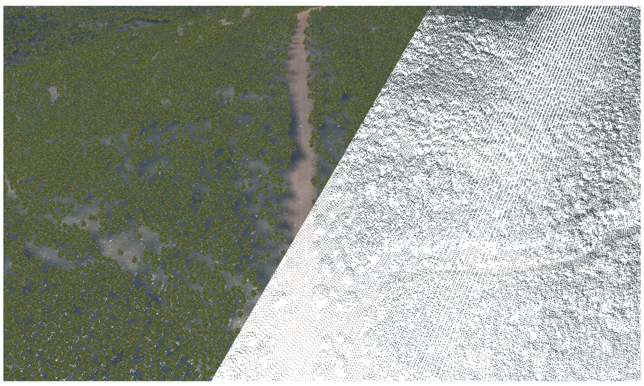

Figure 7: *The top left image shows the reconstructed scenario with x100 density, concerning ground and vegetation, while the bottom right image shows the original point cloud.*

Regarding response time depicted in Figure 8, the reconstruction and enhancement of point cloud density represent a time-consuming task, though minor increments are resolved in a few seconds while also providing a significantly improved rendering. Furthermore, most of the response time of the pipeline corresponds to the definition of the NURBs surface. On the other hand, the instancing and sampling of tree models present a constant and low response time, similarly to the point cloud voxelization and aggregation, which are handled by our parallel implementation. Note that the clustering of large point clouds is also time-consuming. Thus, this stage was not reported in the response time chart since it is computed once and stored.

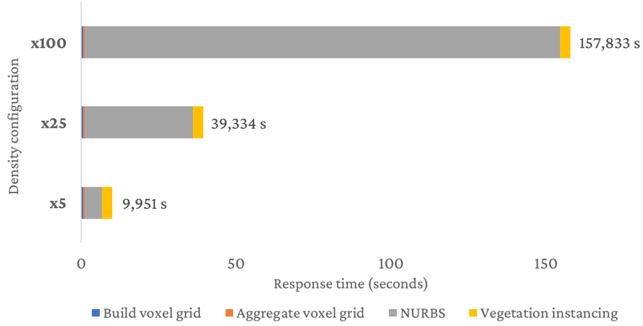

Figure 8: *Stacked response time of the reconstruction methodology in seconds, considering four of five described stages. Vegetation clustering is excluded as it is performed only once per point cloud.*

5. Conclusions

We have presented a preliminary study regarding the reconstruction of real-world environments, to output either realistic scenarios or more dense point clouds. Our results suggested that it is possible to reconstruct ground and vegetation to generate a virtual scene similar to the original source. However, we aim to extend this solution with other scene layers, such as low-vegetation, and to acquire lighting details to homogenize the rendering of all the layers. Also, this method can be implemented as part of an out-of-core solution to extend the dataset boundaries with procedural forest, following patterns similar to those observed in the original dataset.

6. Acknowledgments

This result has been partially funded through the research project 1381202-GEU, PYC20-RE-005-UJA, IEG-2021, which are cofinanced with the Junta de Andalucía, Instituto de Estudios Giennenses and the European Union FEDER funds, as well as by the Spanish Ministry of Science, Innovation and Universities via a doctoral grant to the second author (FPU19/00100).

References

[ACV*14] Andújar, C., Chica, A., Vico, M. A., et al. "Inexpensive Reconstruction and Rendering of Realistic Roadside Landscapes". *Computer Graphics Forum* 33.6 (2014), 101–117. DOI: https://doi.org/10.1111/cgf.12281 2.

[CLJ*22] Collado, José Antonio, López, Alfonso, Jiménez-Pérez, J. Roberto, et al. "Modeling and Enhancement of LiDAR Point Clouds from Natural Scenarios". *Eurographics 2022 - Posters*. Ed. by Sauvage, Basile and Hasic-Telalovic, Jasminka. The Eurographics Association, 2022. ISBN: 978-3-03868-171-7. DOI: 10.2312/egp.20221016 2.

[COB*18] CALDERS, KIM, ORIGO, NIALL, BURT, ANDREW, et al. "Realistic Forest Stand Reconstruction from Terrestrial LiDAR for Radiative Transfer Modelling". *Remote Sensing* 10.6 (2018). ISSN: 2072-4292 1, 2.

[ECC*21] ECORMIER-NOCCA, PIERRE, CORDONNIER, GUILLAUME, CARREZ, PHILIPPE, et al. "Authoring Consistent Landscapes with Flora and Fauna". *ACM Trans. Graph.* 40.4 (July 2021). ISSN: 0730-0301. DOI: 10.1145/3450626.3459952 1.

[GGP*19] GALIN, ERIC, GUÉRIN, ERIC, PEYTAVIE, ADRIEN, et al. "A Review of Digital Terrain Modeling". *Computer Graphics Forum* (2019). ISSN: 1467-8659. DOI: 10.1111/cgf.13657 1.

[KGG*20] KAPP, KONRAD, GAIN, JAMES, GUÉRIN, ERIC, et al. "Data-Driven Authoring of Large-Scale Ecosystems". *ACM Trans. Graph.* 39.6 (Nov. 2020). ISSN: 0730-0301. DOI: 10.1145/3414685.3417848. URL: https://doi.org/10.1145/3414685.3417848 1.

[LDS*11] LI, CHUAN, DEUSSEN, OLIVER, SONG, YI-ZHE, et al. "Modeling and Generating Moving Trees from Video". *ACM Trans. Graph.* 30.6 (Dec. 2011), 1–12. ISSN: 0730-0301. DOI: 10.1145/2070781.2024161 1.

[MHS*19] MAKOWSKI, MIŁOSZ, HÄDRICH, TORSTEN, SCHEFFCZYK, JAN, et al. "Synthetic Silviculture: Multi-Scale Modeling of Plant Ecosystems". *ACM Trans. Graph.* 38.4 (July 2019). ISSN: 0730-0301. DOI: 10.1145/3306346.3323039 1.

[MM08] MERRELL, PAUL and MANOCHA, DINESH. "Continuous Model Synthesis". *ACM Trans. Graph.* 27.5 (Dec. 2008). ISSN: 0730-0301. DOI: 10.1145/1409060.1409111. URL: https://doi.org/10.1145/1409060.1409111 1.

[NPA*22] NIESE, TILL, PIRK, SÖREN, ALBRECHT, MATTHIAS, et al. "Procedural Urban Forestry". *ACM Trans. Graph.* 41.2 (Mar. 2022). ISSN: 0730-0301. DOI: 10.1145/3502220 1.

[Opp86] OPPENHEIMER, PETER E. "Real Time Design and Animation of Fractal Plants and Trees". 20.4 (Aug. 1986), 55–64. ISSN: 0097-8930. DOI: 10.1145/15886.15892 1.

[PNH*14] PIRK, SÖREN, NIESE, TILL, HÄDRICH, TORSTEN, et al. "Windy Trees: Computing Stress Response for Developmental Tree Models". 33.6 (Nov. 2014). ISSN: 0730-0301. DOI: 10.1145/2661229.2661252 1.

[QGY18] QI, JIANBO, GASTELLU-ETCHEGORRY, J.P., and YIN, TIANGANG. "Reconstruction of 3D Forest Mock-Ups from Airborne LiDAR Data for Multispectral Image Simulation Using DART Model". *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*. 2018, 3975–3978. DOI: 10.1109/IGARSS.2018.8517627 2.

[SKW21] SCHÜTZ, MARKUS, KERBL, BERNHARD, and WIMMER, MICHAEL. "Rendering Point Clouds with Compute Shaders and Vertex Order Optimization". *Computer Graphics Forum* 40.4 (July 2021), 115–126. ISSN: 1467-8659. DOI: 10.1111/cgf.14345. URL: https://www.cg.tuwien.ac.at/research/publications/2021/SCHUETZ-2021-PCC/ 2, 3.

[STBB14] SMELIK, RUBEN M., TUTENEL, TIM, BIDARRA, RAFAEL, and BENES, BEDRICH. "A Survey on Procedural Modelling for Virtual Worlds". *Comput. Graph. Forum* 33.6 (Sept. 2014), 31–50. ISSN: 0167-7055. DOI: 10.1111/cgf.12276 1.