# Pointcloud: Implementation of point clouds as graphs in the 3D plant modeling platform GroIMP

Gaëtan Heidsieck [1*], **Tim Oberländer**[1*], **Thomas Hay**[1], and **Winfried Kurth**[1]

1 Georg-August-Universität Göttingen, Lower Saxony, Germany * These authors contributed equally.

Summary

The graph-based functional structural plant modeling (FSPM) platform GroIMP (Kniemeyer, 2008) was extended by a set of functionalities for managing point clouds on the level of individual points using GroIMP's relational graph grammar. These functionalities include the import of XYZ (simple x y z coordinates in CSV style) and PLY (sets of points, edges and faces) based point clouds as balanced trees to the GroIMP simulation graph and a set of basic point cloud manipulation tools. They are extended to support the possible faces and edges from the PLY as part of the point cloud. We use this new implementation in GroIMP in two examples to showcase its usage: a model validation and a fine-grained light interception on meshes.

Statement of need

In plant science, 3D point clouds are widely used as data structure as they are a straightforward way to represent a 3D object. They bring an efficient and fast approach to collect geometric data from plants/trees, which can be complex structures (Okura, 2022). This data can be used for extracting features (Ghahremani et al., 2021), reconstructing topology (Raumonen et al., 2013), (Preuksakarn et al., 2010) or geometry meshes and volumes (Wang et al., 2023), calibrating models, among others. FSPM represents plants by including fine grain properties and organ interactions in the plant growth representation. The FSPM representation benefits from the plant modeling and reconstruction approaches using point clouds, as they provide accurate analyses for model validation (Perez et al., 2023) or data-driven models.

Most FSPM platforms support importing point clouds as static data structures, yet they do not integrate direct interactions such as simulated growth starting from the measured point cloud, light simulation, or rule-based manipulation of the point cloud. Besides this, FSP models commonly use point clouds in an abstracted form by analyzing them with various platform-independent tools. For instance, Bekkers et al. (Bekkers et al., 2024) present an analysis pipeline for point clouds with the goal of using the derived parameters in a GroIMP FSP Model. Only the FSPM platform OpenAlea (Pradal et al., 2008) provides an integrated tool to do semi-automatic 3D reconstruction from a point cloud (Boudon et al., 2014).

The platform GroIMP with its model-specification XL language provides queries for rewriting or retrieving data from graphs. The platform is well-adapted for plant modeling and has been used extensively for FSPM. In GroIMP, the models and data are represented as graphs. Thus, a graph representation of a point cloud eases its integration into the modeling process. This design also enables to leverage the XL language on the point clouds, allowing the usage of complex queries on both the point cloud and the model.

To showcase the possible benefit of such an approach we consider two use cases: validation of growth rules for a model to fit a measured tree, and integration of organs measured at high

resolution into the model. In the case of validating the accuracy of a growth model simulating a measured tree, integrating the point cloud into the environment of the simulation eases the validation. A scenario where this kind of validation in GroIMP is used is the reconstruction of measured pine trees into a topological graph following a growth model. In the case of integrating measured organs (e.g. leaves) into a model, a suitable data integration of point clouds enables fine-grained analyses and manipulations of the organ. As PLY files can also contain faces, it would be possible to use a raytracer and gather the amount of intercepted energy for each face. In a GroIMP model, this enables to use light models on measured leaves at a very precise resolution, while the leaf is connected to a growth model.

PointCloud plugin in GroIMP

To import and export XYZ and PLY point clouds, new filters were created. Since these filters follow the general structure of importing and exporting in GroIMP, they can be used in any part of the software and in combination with other plugins.

The points of the imported point clouds are added as terminal vertices to a balanced tree in the project graph. Adding the objects only as leaves allowed us to keep the concept of local transformation of GroIMP, which enables the user to manipulate the whole point cloud at once using turtle geometry commands.

If the imported file also includes meshes or lines, these objects are added as separate trees. The meshes and lines are then linked to their corresponding points using refinement edges. This structure of separated trees with only specific relationships allows users to easily manipulate only one scale of the structure while keeping the included knowledge/relationships.

For the visualization, the existing graph visitor that builds the 3D scene has been improved to speed up the overall visitation time. Indeed, by default, GroIMP's display visitors go through every node of the project graph for every user interaction. We added a possible cache of 3D objects that prevents re-visiting the same nodes. It is particularly impactful with point cloud graphs with millions of nodes.

A more in depth description of the point cloud objects in GroIMP is available on the software wiki.

Installation & usage

The code is available on gitlab, and can be used in GroIMP, either downloaded and compiled from the source, or directly from the GroIMP plugin manager. Two example projects are embedded within the plugin and can be opened through the example explorer (File/Example projects). They represent the basic examples of the use cases described above. More documentation on both the algorithms/implementation and on how to set up/use the plugin is available from the wiki.

The example projects are used to highlight the features of the plugin. The first one uses a measured leaf organ in a PLY file (Figure 1). The leaf is transformed and added several times to a branch model. A simple XL rewriting rule is used to simulate leaf growth, and the GPUFlux raytracer (Henke & Buck-Sorlin, 2017) in the light model. The absorbed power per area is used to set the RGBAShader of each mesh in order to create a heatmap-like representation (Figure 1).
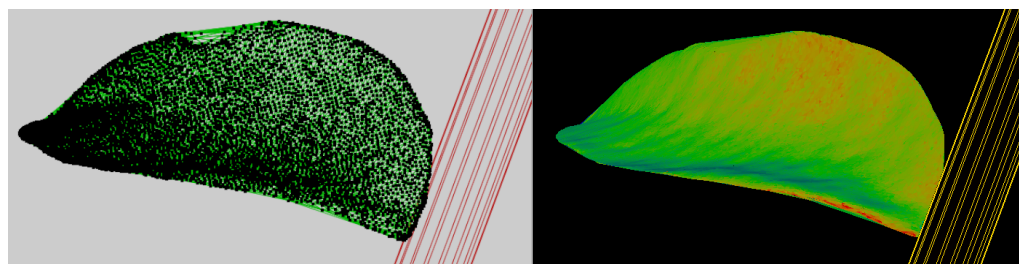
Figure 1: Imported leaf added to a model branch. Results of light modeling. Each mesh is colored depending on its individual amount of light intercepted.

The second example project shows how the point cloud can be integrated in model validation. The plant model is simulated in the same scene as the point cloud. The point's coordinates are tested for whether they fit in the volume of the simulated plant. The result of the fitting can be used in the growth rules (Figure 2).
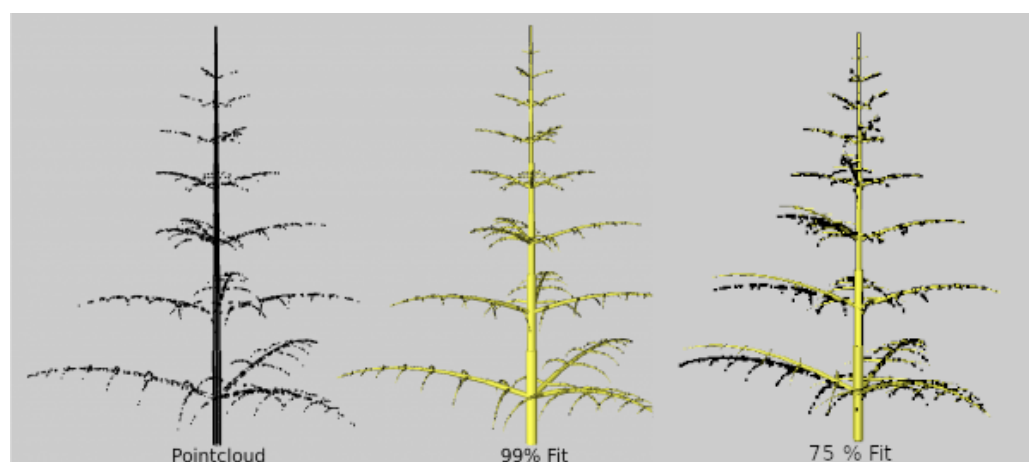
Figure 2: Point cloud of a tree, and reconstruction models that matches the point cloud at 99 and 75 percents.

Acknowledgements

References

Bekkers, V., Evers, J., & Lau, A. (2024). Improving the 3D representation of plant architecture and parameterization efficiency of functional–structural tree models using terrestrial LiDAR data. *AoB PLANTS*, *17*(2), plae071. https://doi.org/10.1093/aobpla/plae071

Boudon, F., Preuksakarn, C., Ferraro, P., Diener, J., Nacry, P., Nikinmaa, E., & Godin, C. (2014). Quantitative assessment of automatic reconstructions of branching systems obtained from laser scanning. *Annals of Botany*, *114*(4), 853–862. https://doi.org/10.1093/aob/mcu062

Ghahremani, M., Williams, K., Corke, F., Tiddeman, B., Liu, Y., Wang, X., & Doonan, J. H. (2021). Direct and accurate feature extraction from 3D point clouds of plants using RANSAC. *Computers and Electronics in Agriculture*, *187*, 106240. https://doi.org/10.1016/j.compag.2021.106240

Henke, M., & Buck-Sorlin, G. (2017). Using a full spectral raytracer for calculating light microclimate in functional-structural plant modelling. *Computing and Informatics*, *36*, 1492–1522. https://doi.org/10.4149/cai_2017_6_1492

Kniemeyer, O. (2008). *Design and implementation of a graph grammar based language for functional-structural plant modelling* [PhD thesis]. BTU Cottbus-Senftenberg.

Okura, F. (2022). 3D modeling and reconstruction of plants and trees: A cross-cutting review across computer graphics, vision, and plant phenotyping. *Breeding Science*, *72*(1), 31–47. https://doi.org/10.1270/jsbbs.21074

Perez, R., Torrelli, V., Roques, S., Devidal, S., Piel, C., Landais, D., Ramel, M., Arsouze, T., Caliman, J.-P., & Vezy, R. (2023). *Testing the capacity of an oil palm FSPM to simulate changes in water and carbon dioxide fluxes under a range of climatic conditions.*

Pradal, C., Dufour-Kowalski, S., Boudon, F., Fournier, C., & Godin, C. (2008). OpenAlea: A visual programming and component-based software platform for plant modelling. *Functional Plant Biology*, *35*(10), 751–760. https://doi.org/10.1071/FP08084

Preuksakarn, C., Boudon, F., Ferraro, P., Durand, J.-B., Nikinmaa, E., & Godin, C. (2010). Reconstructing plant architecture from 3D laser scanner data. *6th International Workshop on Functional-Structural Plant Models*, 12–17.

Raumonen, P., Kaasalainen, M., Åkerblom, M., Kaasalainen, S., Kaartinen, H., Vastaranta, M., Holopainen, M., Disney, M., & Lewis, P. (2013). Fast automatic precision tree models from terrestrial laser scanner data. *Remote Sensing*, *5*(2), 491–520. https://doi.org/10.3390/rs5020491

Wang, W., Li, Y., Huang, H., Hong, L., Du, S., Xie, L., Li, X., Guo, R., & Tang, S. (2023). Branching the limits: Robust 3D tree reconstruction from incomplete laser point clouds. *International Journal of Applied Earth Observation and Geoinformation*, *125*, 103557. https://doi.org/10.1016/j.jag.2023.103557