# **Procedural Tree Generation -** Compressing 3D trees for faster rendering

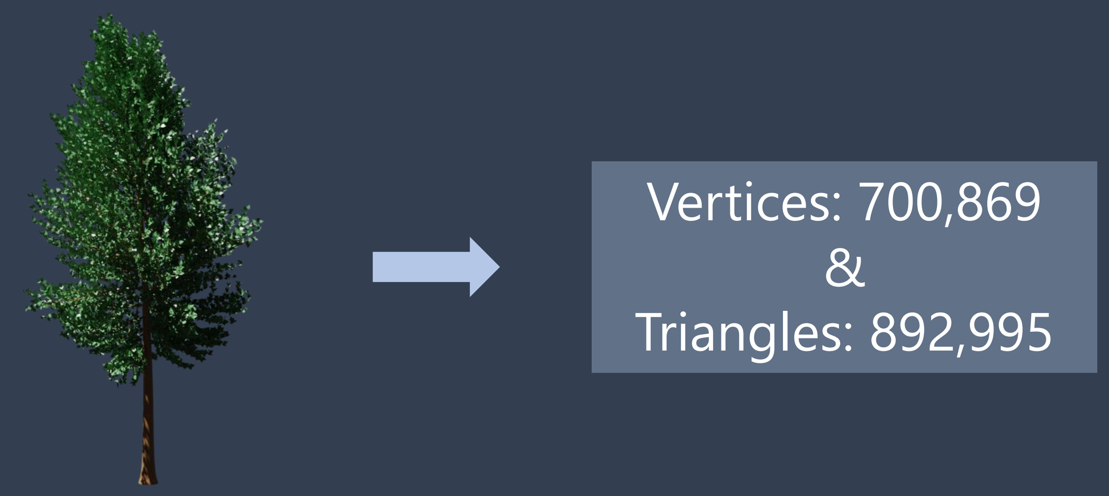
**Sebastian-Alexandru Manda**
s.a.manda-1@student.tudelft.nl

**Supervisor**: Dr. Petr Kellnhofer (p.kellnhofer@tudelft.nl), Lucas Uzolas (l.uzolas@tudelft.nl)
**Responsible Proffesor**: Prof. Dr. Elmar Eisemann (e.eisemann@tudelft.nl)

TU Delft

## 1. Background

Rendering trees has always been computationally expensive as even one tree model can have hundreds of thousands of vertices and triangles as its mesh.




Not all the leaves can be seen at the same time, however, due to refraction, the effects of shading still influence the final render of the model. This becomes even more expensive when rendering more than one tree (a forest).

Methods of compressing trees already exist. We have taken inspiration from the approaches of billboard rendering [1:2] and intend to improve them by using an optimization loop to train a simple model from billboards (upright planes) and attempt to make it resemble the original from all perspectives.

## 2. Research Question

*How to simplify a complex 3D tree into a more compact representation for fast rendering?*

- How does the number of billboards affect the results?
- How do different species of trees affect the results?
- Is L2 loss better than L1 loss as the optimization function?
- How does having different textures on the sides of the billboards instead of a single texture on both sides affect the results?

## References

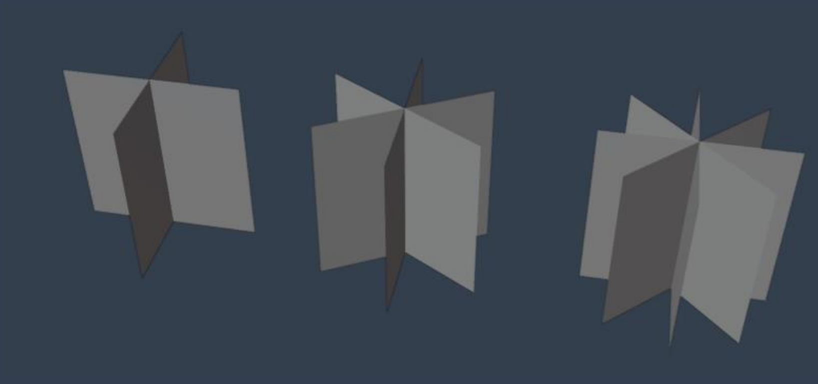
[1] A. Jakulin, "Interactive Vegetation Rendering with Slicing and Blending," in Eurographics 2000 – Short Presentations, Eurographics Association, 2000.

[2] S. Behrendt, C. Colditz, O. Franzke, J. Kopf, and O. Deussen, "Realistic real-time rendering of landscapes using billboard clouds," Computer Graphics Forum, vol. 24, no. 3, p. 507–516, 2005.

[3] S. Laine, J. Hellsten, T. Karras, Y. Seol, J. Lehtinen, and T. Aila, "Modular primitives for high-performance differentiable rendering," ACM Transactions on Graphics, vol. 39, no. 6, 2020.

## 3. Method

Render 360˚ images of the ground truth models. Split images into a training set and a test set.

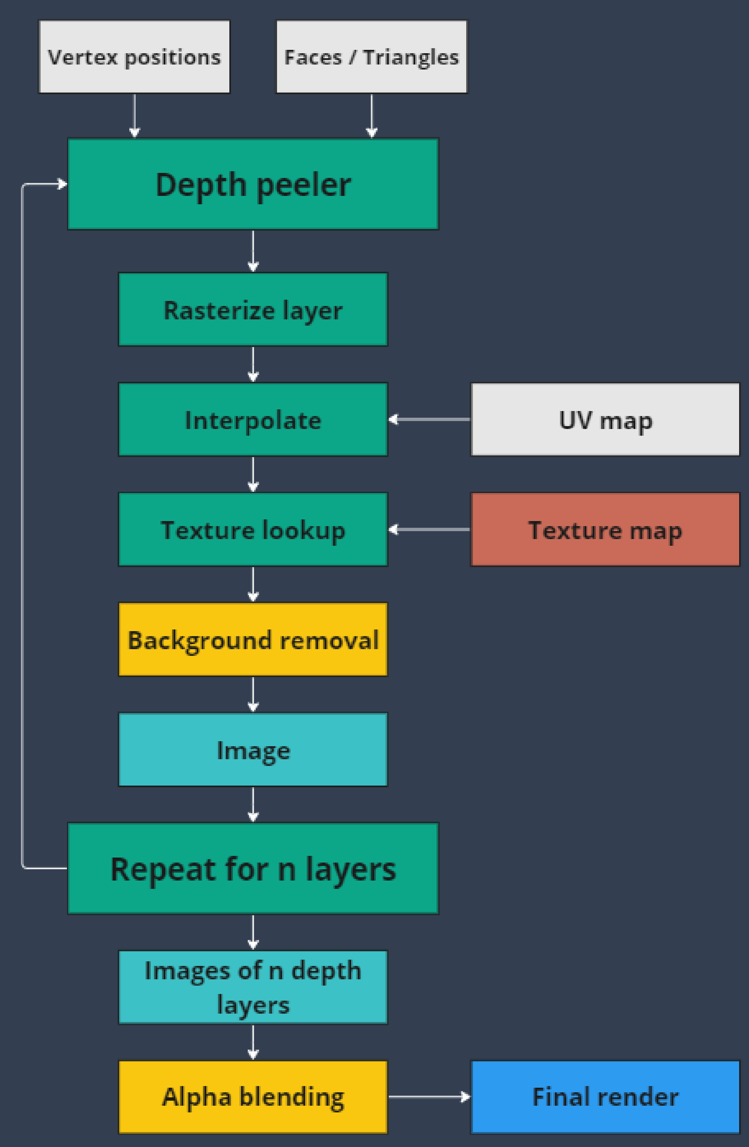
Create base models onto which we will optimize the texture. These models are composed of upright planes at uniform rotations around their center vertical axis.

We sample a random angle θ from the training set and render our optimized model from that perspective.

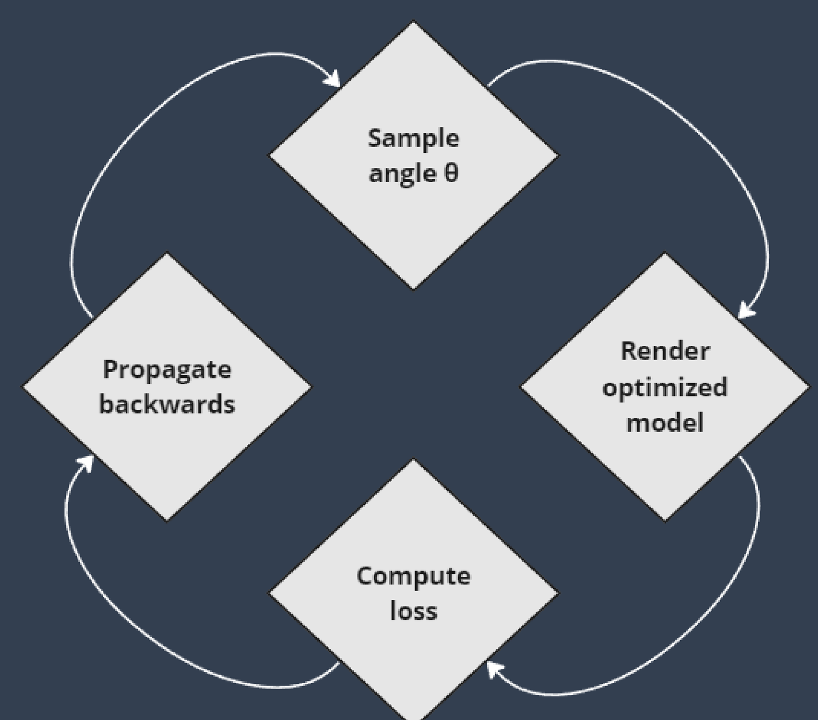
We use NVdiffRec [3] computer graphics primitives to rasterize, interpolate, and texture all depth layers of the model.

Images of the depth layers are combined through alpha blending to create a final transparent image of the entire model.

Compute either L1 loss or L2 loss between the rendered image and ground truth depending on the experiment.




Rendering Pipeline



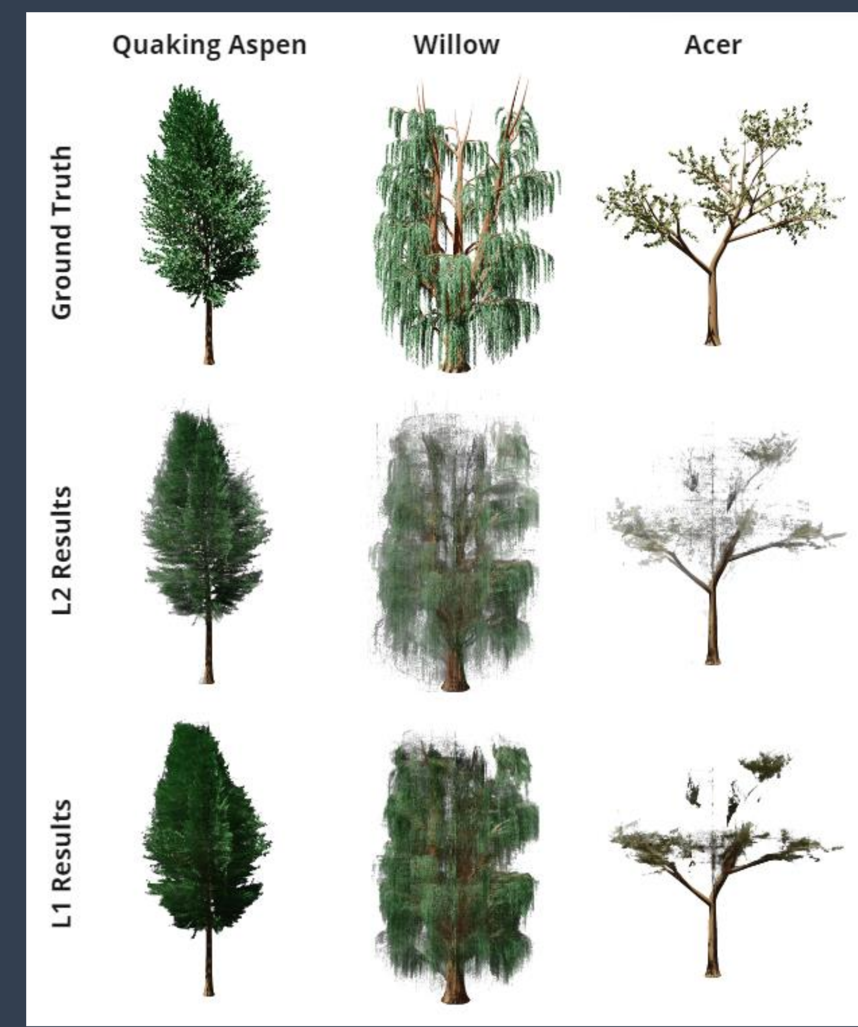

Optimization Loop - Repeat for n Iterations

We propagate the loss to optimize the texture of the model. By repeating this loop for 10,000 iterations we reach a model that resembles the original.

After training, we render all angles in the test set and compute the average **RMSE**, **SSIM**, **PSNR**, and **SRE** for evaluation.

## 4. Evaluation and Results

We find that having more billboards in the model generally has a positive impact on the results of the optimization. However, not all species of trees respond positively. Trees with fewer leaves such as the willow and acer may benefit from lower numbers of billboards instead.




Moreover, the optimization function chosen has a major impact on the final images. L2 loss has better quantitative results but produces models with blurry edges, while L1 loss creates sharper images but they are darker and have worse quantitative results.

In addition, bounding the billboards of the models to the sizes of the trees also improves results. This helps by reducing the area that can be optimized, limiting it to the size necessary to accurately portray the tree.

A plane can only hold one texture when it comes to rendering it, creating a limit to how much it can resemble two opposing perspectives of the same tree. This can be improved by simulating double-sided billboards by creating two adjacent planes instead of one, each with a separate texture that is applied. This greatly improves the quality of the results for all tree species.

## 5. Conclusion and Discussion

Each tree species responds differently to our methods and, although a general setup that can produce the most optimal models for each species is not possible with our current method, specialized parameters can successfully compress any species of tree.

Additionally, our method can be improved by optimizing the initial models the tree is compressed onto. One such way is to compute the best sizes and orientations for the billboards of the model such that each covers an important dimension of a tree. These dimensions can be main branches or large collections of leaves, chosen to best fit the shape of a billboard.

Finally, our approach reduces the size of the initial model to at most 0.028% of the size of the original model when it comes to the number of vertices. This allows for much quicker render times for both individual models and for entire forests of trees. The tradeoff, however, is in the quality of the results, where depending on the used loss function the compressed model may become blurrier or darker than the original. This demonstrates that, with few improvements, this approach can generate faster renders and lighter-to-store models than the ones that inspired it.