Adaptive Billboard Clouds for Botanical Tree Models

Julian KRATT[1], Liviu COCONU[2], Timm DAPPER[2], Jan Walter SCHLIEP[2],
Philip PAAR[2], Oliver DEUSSEN[1]

[1]University of Konstanz, Germany Julian.Kratt@uni-konstanz.de
[2]Laubwerk GmbH, Germany

Abstract

We present a framework for automatic Level-of-Detail control of botanical tree models based on hierarchical billboard clouds. These clouds react to quality measures developed specifically for plant models. They are optimized for providing minimum visual differences compared to the full polygonal models. The tree structure is analyzed to determine the geometric parts of the tree that are substituted by billboards. By computing the implicit surface of the tree model we can efficiently determine the average occlusion of each area within a tree that is used to guide the substitution. Our system is validated by comparing simplified versions of different trees with their full polygonal models. We realized our system as plug-in for MAXON CINEMA 4D.

1 Introduction

A significant part of the challenge in creating interactive photorealistic virtual landscapes is the management of Levels of Detail (LoD) to reach the best possible balance between frame rate and the realism of represented features. Several studies have demonstrated that an appropriate level of detail, in particular for features in the foreground, is a key factor for how people emotionally relate to landscape visualizations (APPLETON & LOVETT 2003, PAAR 2006). At the same time, realistic rendering of large 3D scenes with dense vegetation still remains a challenge due to their high geometric complexity. We present a framework for automatic LoD control of botanical tree models based on hierarchical billboard clouds. While in existing approaches the user has to tune many parameters for an optimal billboard approximation, our method is capable of automatically deciding which parts of the tree a billboard cloud for a certain distance represents. Furthermore, our billboard clouds can be adapted in order to react to quality measures that are developed specifically for plant models. On one side an upper limit for the polygon count can be set. On the other side the resulting billboard representation is optimized for providing minimum visual differences compared to the full polygonal. We analyze the tree structure in a pre-processing step to determine the geometric parts of the tree that are later substituted by billboards. By computing the implicit surface of the tree model we determine the average occlusion of each area within a tree. The geometry of highly occluded parts is more likely to be substituted with billboards. The other important aspect that directs billboard production is given by the hierarchical branching information within the tree. To generate different LoDs for a single tree model with respect to the given quality measure, the distance to the virtual viewer modulates the entire process. We validate our system by comparing simplified versions of different trees with their full polygonal models using a Precision and Recall method. For a wide range of input models our approach achieves good results without the need to define any parameters. We realized our method as plug-in for MAXON CINEMA 4D, thus enabling 3D content creators to directly use the system.

2 Related Work

In existing software systems, the rendering of dense vegetation is still solved with very simple methods. The 3D Tree Showcase of Google Earth (GOOGLE 2013) uses a rather simplistic looking billboard approach, also in ESRIs ArcGIS 10.2, trees are depicted simply by crossed billboards. Trees in Apple Maps and Nokia Here are part of the polygonal ground surface and are represented as coarse ‘blobby'-like shapes. A number of approaches were invented that reduce high polygon counts of complex models (see HECKBERT & GARLAND 1997). However, these methods work on smooth surfaces by removing vertices. For trees this does not work out since these objects are represented by many disconnected triangles. REEVES & BLAU (1985) introduce the first approach specifically for trees by approximating them by a set of colored disks. WEBER & PENN (1995) use a combination of points and lines to reduce the complexity of trees. DEUSSEN et al. (2002) use the same primitives but store them in individual buffers to achieve smooth blending between different LoDs. GARDNER (1984) proposed one of the first image-based representations of natural scenes. A number of textured quadratic surfaces represent natural features such as trees. Using billboards, viewer-facing surfaces, was a common technique for many years, e.g. by ROHLF et al. (1994). Here, a tree is represented by a single image; however, this can be done only in far distance because no 3D information is represented. JAKULIN et al. (2000) use a combination of tree geometry for the trunk with crossing billboards to render the tree crown. DÉCORET et al. (2003) introduces billboard clouds: a number of arbitrarily oriented billboards is used to represent complex geometry. However, for trees this approach needs a lot of billboards to produce good enough results due to their high aggregate detail. As a reaction, BEHRENDT et al. (2005) present a method to generate billboard clouds, which are specifically designed for trees. In contrast to our work this approach needs a lot of user input to create such approximations. Furthermore, the creation of the billboards is not optimal, since no geometric relations within the tree are considered. COLDITZ et al. (2006) use the same metaphor to render huge scenes efficiently. He combines the point and line-based rendering of DEUSSEN et al. (2003) with billboard approximations from BEHRENDT et al. (2005) to create almost invisible transitions from full geometric models to simple billboard representations. To evaluate the quality of our models, we use the method by NEUBERT et al. (2011). This method is based on Precision and Recall, two well-known measures from machine learning, to determine the quality of model approximations using statistics.

3 System Overview

Our method can be applied to a wide range of models: they may be generated by procedural methods such as L-Systems, manually designed (found in large Internet repositories) or models from real trees obtained by reconstruction methods.

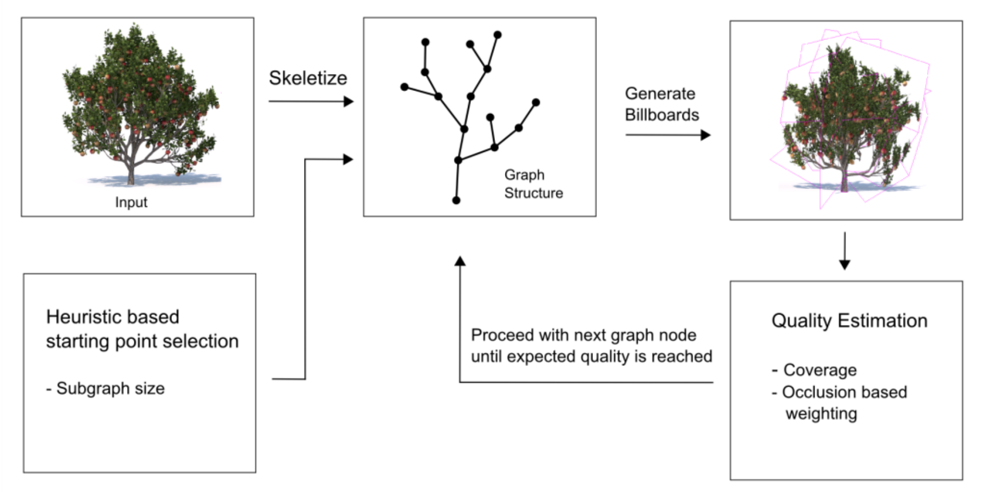

Fig. 1: Overview of the system.

Given the polygonal tree model, first the skeleton structure is determined and stored as a graph, represented by a set of points with connectivity information. Our algorithm then makes an initial pick of sub-graphs to be turned into billboards using a simple heuristic based on the sub-graph size. Starting with this initial set of sub-graphs, our algorithm then iteratively generates billboards for each sub-graph. An error metric is applied to estimate the quality of their billboard cloud representation. The error metric is evaluated for each sub-graph independently and decides if to keep the billboard representation (if the accepted error range is met), to merge sub-graphs, to split them up further if the acceptable error range is not met. We evaluate the simplification quality of the result with respect to a given upper limit of the polygon count and/or a maximum visual error that is acceptable. This process is repeated until all sub-graphs pass the quality check (cf. Figure 1).

3.1 Input Model

As mentioned above, our method is able to handle many tree models from different sources. The only requirement is that the model has to be described by a polygonal mesh. This mesh is analyzed and the skeletal structure is extracted. Our algorithm detects extruded elements connected by polygons and converts them into chains of graph nodes. In a next step these chains are connected at points of overlap, so each branch (except for the trunk) is connected to a parent chain. We assume that leaves are represented as disconnected geometry. These are collected and similarities are computed. Leaf objects that meet a certain similarity criterion are considered only once. This allows us to use a relatively lightweight data structure in which the original topological structure of the tree is directly represented and the following steps can be executed on.

3.2 Initial Sub-graph Selection

It is possible to select the graph for the whole tree as a starting point for the computation and just let the billboard creation and quality estimation split the graph until the specified detail range is met. However, this will take a large number of steps the higher the quality requirements are chosen and the bigger the tree model gets. Therefore, choosing a good initial sub-graph size is not a functional requirement for our algorithm but can result in a considerable performance boost. These steps need a guess of about how large a sub-graph bounding box can get while still meeting the quality requirements. It then walks recursively through the node graph from the tips to the trunk and picks sub-graphs that are close to the selected bounding box size for initial generation of billboards.

3.3 Billboard Generation Algorithms

Because plant models exhibit a large variation in their nature, there is no "one size fits all" representation that works well in for all plants, or even for all parts of the same plant. For example, some "amorphous" plant structures such as the foliage of most deciduous trees can be represented well by two cross billboard planes. Other structures such as long leaves or leaf structures in tropical trees are not well suited for such an approximation. A long and curved strip made up of multiple segments might yield better results here.

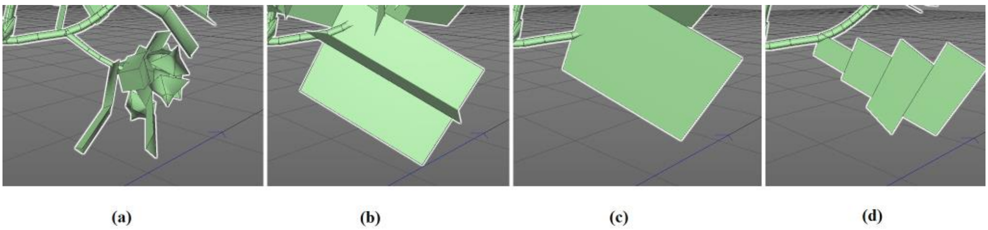

Fig. 2: Original geometry (a), basic 2-cross (b), plane fit (c) and plane strip fit (d).

Unlike previous systems that required human intervention, our goal was to make an automatic choice among different representation options, while still allowing some degree of user intervention, if required. Given a sub-tree of the original model that needs to be represented, our system iteratively attempts to find the best representation within the quality metric constraints at each node. Thus, different billboard generators can be freely mixed, which offers a significant degree of flexibility.

Basic 2-Cross: The most simple billboard generation algorithm just generates two orthogonal quads that share the main axis of the bounding box of the respective model part (see Figure 2b). This axis is computed as an average of all vectors emerging from the connector point (the vertex corresponding to the node in the skeleton from which we are generating the billboard) to all vertices of the geometry that should be represented. This representation is typically a good fit for typical branch foliage.

Plane Fit: A more evolved version fits an arbitrarily oriented plane through all vertices. It then generates a quad in this plane (Figure 2c). This approach yields potentially better results than cross billboards if the foliage of a branch is rather flat. This generator could be extended to generate several planes that better fit more complex foliage structures; however, at the expense of more polygons and the risk of potentially unpleasant artifacts due to intersecting planes.

Plane Fit Strip: For matching long and curved foliage structures, a third and more complex billboard generation strategy is used. First, a main bounding box axis is computed as described in the basic algorithm above. The original vertices are partitioned along this axis into a predefined number of equally sized sections. For each section, we fit a freely oriented plane to the vertices in this section that is constrained by the end points of the previous section (Figure 2d). The resulting quad can be connected to the previous section. Optionally, we merge sections if the resulting error stays below a certain threshold.

3.4 Simplification Quality

In order to guide our generation process for billboard clouds we define a measure that reflects the quality of simplified parts compared to their geometric representation. The simplification is given by a set of textured planes. Similar to DÉCORET et al. (2003) we use the notion of validity and coverage. A point is considered as valid if the Euclidean distance to its simplification, thus the distance to one of the planes, is less than a certain threshold ε. This idea can be generalized to faces (triangles or other simple objects): a face is considered to be valid if every vertex of the face is valid. In contrast to DÉCORET et al. (2003) we are interested in analyzing given simplifications with respect to their approximation quality. In consequence not every point has to fulfill the validity criterion. Let us assume that we have one part of the original tree geometry defined by a number of faces and its simplification given by a set of billboards P. The valid set of faces is denoted by valid(P). The remaining set of faces is denoted by invalid(P). Following the idea of coverage we compute for every valid and invalid face the projected area of the triangle on its nearest billboard plane. Intuitively, high coverage is achieved when all points of the geometry are valid and the area of the projection is relatively large. However, the more points are invalid, the less coverage would be expected. In our work we extend the formula of coverage presented in DÉCORET et al. (2003) by taking this into account:

$$C(P) = max\left(0.0, \sum_{f \in valid\ (P)} area(f) - \sum_{f \in invalid\ (P)} area(f)\right)$$

By doing so, invalid faces are penalized. In order to return always-positive values for the coverage we clamp negative values to zero. We know that negative values may also be interesting for consideration, but for our purpose and further computations this is not necessary. DÉCORET et al. (2003) introduce a penalty term that also penalizes planes that “miss” faces. This is necessary for the definition of a proper plane space in order to find an optimal set of billboards. However, in our case we already have a set of planes that we want to evaluate, thus we directly incorporate the penalty into the coverage term.

Occlusion based Weighting: Many tree models have a huge number of leaves and small twigs. These geometric entities define the characteristic shape of the tree. In addition they are also often hiding the inner parts of the tree. Those highly occluded areas should be more likely to be substituted by billboards. To account for this we extend our approximation-based measure with an additional weighting term.

For a given point p within the tree we can efficiently determine the occlusion by surrounding geometry if we define this geometry by an implicit function. Following the idea of Metaballs (MURAKAMI et al. 1987, WATT 2000) we define an implicit surface by a set of generator points P. Each generator point $p_i \in P$ has a certain influence radius r_i.The influence for point q is then given by a density function D_i:

$$D_i(q) = \begin{cases} \left(1 - \left(\frac{\|q - p_i\|}{r_i}\right)^2\right)^2, & if \ \|q - p_i\| < r_i \\ 0, & if \ \|q - p_i\| \geq r_i \end{cases}$$

The implicit surface can now be represented by the density field $\mathcal{F}$ defined by the sum over the individual density functions. Following the approach presented in LUFT et al. (2007) we use the set of leaves of the tree model to define the generator points. A leaf is given by one geometric entity, such as a single point, triangle or any other complex polygon. The position of the generator point for one leaf is given by the center of mass of the leaf geometry. The influence radius is one parameter that has to be set by the user. However, in all our tests a value of 25.0 leads to good results. To compute the occlusion value for a point q, we directly evaluate the density field: $d_q = \mathcal{F}(q)$. Large positive values describe highly occluded parts within the tree. Similar to LUFT et al. (2007) a transfer function is used to map the occlusion value into the range [0...1]:

$$\sigma(q) = \begin{cases} 1, & , d_q \leq 0 \\ 1 - \frac{d_q}{d_{min}} & , 0 < d_q < d_{min} \\ 0 & , d_q \geq d_{min} \end{cases}$$

with $d_{\min}$ as lower bound for all density values. We finally define our quality measure that reflects how well a simplification approximates its original geometry with:

$$quality(P) = \sigma(P) * C(P),$$

where $\sigma(P)$ represents the normalized average occlusion of all points $p_i \in P$ of the geometry.

4 Evaluation

To validate the visual quality of the simplified tree models we use Precision and Recall (PR), well-known statistical measures for exactness and completeness that are mainly used in information retrieval. Assuming a database containing a collection of documents the quality of a query can be determined by computing PR. The result of the query can be seen as a binary classification. All documents that are wanted in the result are correctly classified and all others not. Given the total number of all documents in the database, the correctly classified and not correctly classified ones PR can be computed.

In general *Precision* is given by the fraction of correctly classified items (true positives) to all existing items, thus correctly identified and not correctly identified items (true positives and false positives). The ratio describes the exactness or accuracy of a query. *Recall* is defined by the ratio of correctly classified items (true positives) and all relevant items (true positives and false negatives) and is a measure for the completeness:

$$P = \frac{tp}{tp + fp} \text{ and } R = \frac{tp}{tp + fn},$$

with true positives *fp*, false positives *fp* and false negatives *fn*. NEUBERT et al. (2011) use this concept to measure how well a simplified tree model represents the information of the original one. They consider the set of pixels covered by the original tree model (P_{orig}) in comparison to the set of pixels covered by the simplified one ($P_{\text{simplified}}$). Following the notation of NEUBERT et al. (2011) we compute PR with:

$$tp = \{p \mid (p \in P_{simplified}) \wedge (p \in P_{orig})\}$$
$$fp = \{p \mid (p \in P_{simplified}) \wedge (p \notin P_{orig})\}$$
$$tn = \{p \mid (p \notin P_{simplified}) \wedge (p \in P_{orig})\}$$

To determine the *true positive* rate we count the pixels that are covered by both models, the original rendered at full detail and the simplified one using billboard clouds. *False positive* describes the number of pixels that are not correctly set, i.e. the number of pixels that are only set for the simplified model. *False negative* is given by the number of covered pixels that are missed from the simplified model, i.e. the number of pixels that are only covered by the original model. We applied our method to a set of tree models to automatically compute different level of details. For each simplification we determined the PR score with respect to the original model. Some of our results are presented in Figure 3. We achieve high PR scores even when the full polygonal tree model is highly simplified.

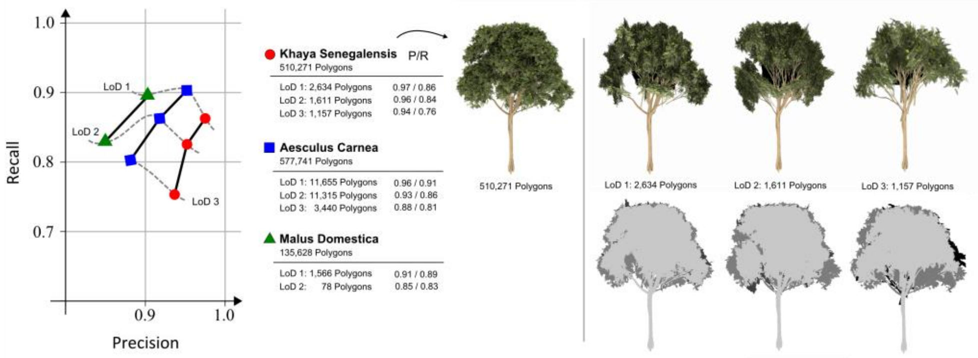

Fig. 3: Precision and Recall diagram for different tree models and their LODs (left). Visualization of PR for one of the tree models: light grey *true positive*, grey *false negative*, black *false positive* (right).

5 Implementation and Results

We realized our system as plug-in for MAXON CINEMA 4D enabling content creators to directly use our method. While this gave us the freedom to concentrate on the core algorithms without spending time on regular “housekeeping” tasks such as user Interface and 3D-viewing methods, the technology itself is completely independent from CINEMA 4D and could be brought into most other 3D-applications or turned into a standalone program if needed. Figure 4 shows a typical application of our method. Given a full polygonal tree model we can automatically produce different levels of detail that can we used to efficiently render large populated scenes.

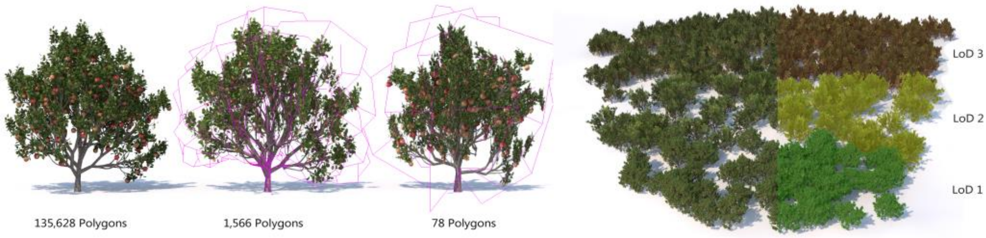

Fig. 4: Left side: Full polygonal tree model and two simplified versions each with an upper limit for the polygon count generated with our system. Right side: Visualization of a large botanical scene. Color indicates different levels of detail.

6 Future Work

So far, we focused on generating a single billboard representation for a given model and quality metric. For different levels of detail, apart from the obvious approach of generating several independent representations, our system can be naturally extended to generate a hierarchical representation of billboards at different quality metrics, which can be chosen and transitioned in a view dependent manner. Billboard representations that allow smoother transitions should be investigated. Another potential area of investigation and improvement is a better method for picking an initial set of nodes for billboard generation, this would save the time spent by unnecessarily generating and evaluating sets of billboards.

References

APPLETON, K., LOVETT, A. (2003): GIS-based visualization of rural landscapes: defining ‘sufficient’ realism for environmental decision-making. Landscape and Urban Planning, Vol. 6,:117-131.

BEHRENDT, S., COLDITZ C., FRANZKE, O., KOPF, J., DEUSSEN, O. (2005): Realistic Real-Time Rendering of Landscapes Using Billboard Clouds. Computer Graphics Forum (Proceedings of Eurographics), Vol. 24(3): 507-516.

COLDITZ, C., COCONU, L., DEUSSEN, O., HEGE, H-C. (2006): Real-Time Rendering of Complex Photorealistic Landscapes Using Hybrid Level-of-Detail Approaches. In: Buhmann/Paar/Bishop/Lange (Eds.), Trends in Real-Time Landscape Visualization and Participation, Proceedings at Anhalt University of Applied Sciences. Wichmann Verlag

DEUSSEN, O., COLDITZ, C., STAMMINGER, M., DRETTAKIS, G. (2002): Interactive visualization of complex plant ecosystems. In Proceedings of the IEEE Visualization Conference.

DEUSSEN, O. (2003): A Framework for Geometry Generation and Rendering of Plants with Applications in Landscape Architecture. Landscape and Urban Planning, Vol. 64:1, Elsevier Science: 105-113.

DÉCORET, X., DURAND, F., SILION, F.X., DORSEY, J. (2003): Billboard clouds for extreme model simplification. ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2003:689-696. GARDNER, G. (1984): Simulation of natural scenes using textured quadric surfaces. In ACM Transactions on Graphics (SIG- GRAPH), pages 11–20

GOOGLE (2013): GoogleEarthShowcases- 3D Trees. http://www.google.com/earth/explore/showcase/trees.html (08.10.2013).

HECKBERT, P., GARLAND, M. (1997): Survey of polygonal surface simplification algorithms. Technical report, Carnegie Mellon University, 10

JAKULIN, A. (2000): Interactive vegetation rendering with slicing and blending. In A. de Sousa and J.C. Torres, editors, Proceedings of Eurographics `00.

LUFT, T., BALZER, M., DEUSSEN O. (2007): Expressive Illumination of Foliage Based on Implicit Surfaces. Eurographics Workshop on Natural Phenomena: 71-78

MURAKAMI S., ICHIHARA, H. (1987): On a 3Ddisplay method by metaball technique. Transactions of the Institute of Electronics, Information and Communication Engineers. J79-D, 1607-1615

NEUBERT, B., PIRK, S., DEUSSEN, O., DACHSBACHER, C. (2011): Improved Model- and View-Dependent Pruning of Large Botanical Scenes. Computer Graphics Forum 30(6): 1708-1718. PAAR, P. (2006): Landscape visualizations: Applications and requirements of 3D visualization software for environmental planning. Computers, Environment and Urban Systems, Elsevier, 30, 815-839.

REEVES, W., BLAU, R. (1985): Approximate and probabilistic algorithms for shading and rendering structured particle systems. In ACM Transactions on Graphics (SIGGRAPH 1985 Conference Proceedings), pages 313–322

ROHLF, J., HELMAN, J. (1994): IRIS Performer: A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics. SIGGRAPH '94, pages 381-394 WATT, A. (2000): 3D Computer Graphics. Addison Wesley

WEBER, J., PENN, J. (1995): Creation and rendering of realistic trees. In ACM Transactions on Graphics (SIGGRAPH), pages 119–128