Volume 29 (2010), Number 4 pp. 1–11 COMPUTER GRAPHICS *forum*

Improved Model- and View-Dependent Pruning of Large Botanical Scenes

B. Neubert[1] S. Pirk[1] O. Deussen[1] and C. Dachsbacher[2]

[1]Universitaet Konstanz, Germany
[2]Computer Graphics Group/ Karlsruhe Institute of Technology, Germany

Abstract

We present an optimized pruning algorithm that allows for considerable geometry reduction in large botanical scenes while maintaining high and coherent rendering quality. We improve upon previous techniques by applying model specific geometry reduction functions and optimized scaling functions. For this we introduce the use of Precision and Recall (PR) as a measure of quality to rendering and show how PR-scores can be used to predict better scaling values. We conducted a user-study letting subjects adjust the scaling value, which shows that the predicted scaling matches the preferred ones. Lastly, we extend the originally purely stochastic geometry prioritization for pruning to account for view-optimized geometry selection, which allows to take global scene information, such as occlusion, into consideration. We demonstrate our method for the rendering of scenes with thousands of complex tree models in real-time.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—I.3.3 [Computer Graphics]: Picture/Image Generation—

1. Introduction

Rendering of natural scenes with vegetation as rich as in the real world has been a motivation of computer graphics research ever since. The complex visual appearance and the inhomogeneous structure of botanical objects makes real-time rendering of large scenes a challenging task that extends to this day. The obvious main reason is the tremendous amount of geometry that is needed to represent trees and plants. Storing as well as rendering such objects with full detail is beyond the capabilities even of modern graphics hardware. However, even if processing and rendering the data were possible, then the small sub-pixel details due to the complex geometry can still cause aliasing artifacts.

Many different approaches have been presented to render trees in real-time. Most often simple billboards or impostors [SSK96] are used, or automatically generated billboard clouds [DDSD03, GSSK05] which are sets of billboards that better preserve occlusion and parallax effects. However, these representations are well-suited for distant objects and trees, but they are typically over simplified and close views reveal the low quality. It is also not possible to achieve coherent shading of the scene or to adapt the level of detail smoothly and without noticeable artifacts due to the planar nature of billboards [LEST06].

In this paper we present a rendering technique for complex botanical scenes based on pruning. Pruning techniques (stochastically) reduce geometry by simply excluding some parts of the model, e.g. leaves, from the rendering and correcting contrast and the total rendered area by scaling the remaining leaves [CHPR07]. We improve upon previous methods in several respects:

- We describe a view-optimized pruning instead of purely stochastic simplification of the geometry. This allows us to account for global scene information, e.g. thick and sparse forest and occlusion from neighboring trees.
- We show that scaling of geometry after pruning should not be inversely proportional to the geometry reduction.
- We formalize this by introducing Precision and Recall as a measure of rendering quality. Our measure does not consider pixel colors, but whether the right pixels of a rendered object are set. We also validate this by conducting a user-study where subjects had to manually adjust the preferred scaling value.

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

1.1. Related Work

Since the very beginning of computer graphics, rendering algorithms have been high consumers of computational resources and memory. In general, level of detail (LOD) methods aim at cutting down the rendering cost, mainly by reducing model detail or shading cost. Covering the huge body of work in this field is beyond the scope of this paper. Thus we refer to Luebke et al.'s excellent textbook [LWC*02] and to the work of Drettakis et al. [DBD*07], which provides an overview over perceptually-based rendering and level of detail in this context.

Here we focus on geometry reduction techniques closely related to our work. There exists a variety of texture-based techniques, most notably impostors [SSK96] and billboard clouds [DDSD03], which have been tailored for tree rendering [GSSK05, LEST06]. These techniques generate discrete levels of detail which require special treatment to avoid distracting popping artifacts when adapting the detail, e.g. using [SW08]. Decaudin and Neyret [DN04] use 3D textures representing parts of a dense forest and aperiodic tiling to render large scenes by volume slicing. However, this method only allows distant views, e.g. as used in flight simulators. They further extended this approach to volumetric impostors [DN09], but this method shares the drawback of high memory consumption with the other texture-based techniques. Rebollo et al. [RRCR06] describe a GPU-friendly rendering technique for foliage using a multiresolution representation obtained from split and collapse operations, and Gumbau et al. [GCRR10] extend this idea by a view-dependent measure. Both however, require additional memory for storing the multiresolution model.

Due to the aforementioned limitations of these approaches, we base our method on pruning techniques. Especially the simple and efficient idea of stochastically removing geometry makes such approaches extremely simple to use and implement. Stochastic simplification can be easily used with geometry representations that do not require topology information, such as point rendering methods. The QSplat algorithm [RL00] and sequential point trees [DVS03] adapt the size of prepositioned splats (rendered point primitives) to the required sampling density. Point samples can also be created on the fly, e.g. distributed randomly onto surfaces [WFP*01], stratified [WS02], or adaptively [SD01]. All these methods have in common that—when using fewer samples—they preserve the total area by scaling the rendered remaining primitives. Klein et al. [KKF*04] used a stochastic simplification for polygonal scenes, however, the scene elements are only discarded, not altered, and thus this method is only suitable for coarse previews. Deussen et al. [DCSD02] applied stochastic simplification to rendering complex ecosystems. For reducing geometry they replaced the original triangles successively by lines and then points. Cook et al. [CHPR07] transfer this idea to complex geometry not restricted to point representations. They demonstrate

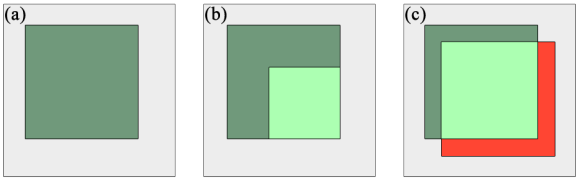

Figure 1: *a) dark green: pixels covered by original "model". b) light green: pixels covered by the "model" rendered with reduced geometry and without scaling: no additional pixels are covered and thus only the Recall value is affected, while the Precision score remains 1. c) the simplified and scaled "model" covers pixels that were not covered by the original model (red, false positives). Dark green pixels are false negatives, light green ones are true positives.*

simplification by pruning and scaling adapting not only to an object's screen size, but also to motion blur and depth of field. Their work is closely related to ours, however, we show that the rendering quality can be improved by cleverer scaling. An overview over various further techniques for rendering vegetation, such as fractal-based or space partitioning methods, is given by Zhang and Pang [ZP08].

2. Improved Scaling for Pruning Algorithms

In this section we briefly recap the pruning and scaling described by Deussen et al. [DCSD02] and Cook et al. [CHPR07], and discuss the drawbacks of these approaches. Next we introduce the Precision and Recall measure for pruning and scaling, and present the results of our user-study conducted for validation.

2.1. Area Preservation and Optimal Scaling

The main objective of simplification algorithms is to preserve the overall appearance of the rendered models whilst using less geometry and thus reducing rendering cost. Deussen et al. [DCSD02] as well as Cook et al. [CHPR07] propose a simple, and at first sight plausible rule: when the geometry is reduced down to a certain fraction then the remaining geometry is scaled such that the total area of rendered surfaces is equal to the original area. Cook et al. [CHPR07] denote this scaling factor as $s = 1/\lambda$, where λ is the fraction of rendered geometry. However, the surface area that is visible after rendering the remaining geometry heavily depends on the actual rendered model. One can easily think of models where a lot of geometry can be removed and they would still cover the same projected area, i.e. the remaining geometry covers (almost) the same pixels for a certain view direction.

Stochastically pruning the geometry does not only change the area, but also—in particular when pruning strongly—the depth complexity of the rendered model. Cook et al. [CHPR07] account for this by calculating the expected

Figure 2: The Precision-Recall diagram for different plant models, five different geometry levels and varying scaling values. An interesting case is the Ulmus model (right): it does not benefit from scaling for high values, and scaling even lowers the PR-score, i.e. the distance of the PR-coordinate to the top-right corner which represents the optimal score and $P = R = 1$.

visible area of a subset of randomly chosen elements of a model, and adapt the scaling factor accordingly. Our results demonstrate an important and interesting fact: the largest decrease in rendering quality due to wrong scaling does not occur for strong, but for slight and moderate pruning, where the depth complexity correction has only little influence. In the next section we introduce the Precision and Recall measure which does not only take the number of pixels but also their classification into correctly set and unset pixels into account.

2.2. Precision and Recall

Precision and Recall (PR) are well-known statistical classifications or measures for exactness and completeness. They are widely applied in the domain of information retrieval and are closely related to sensitivity and specificity to measure the performance of binary classification algorithms, such as support vector machines and Bayesian networks [Rij79].

Precision is defined as the ratio of correctly identified items (true positives) to both correctly and incorrectly identified items (sum of true positives and false positives). In our case, when rendering a pruned model it is the ratio of pixels that are correctly set, i.e. they would have been rendered for the full-detail model as well, and the total number of set pixels.

Recall is the quotient of correctly identified items (true positives) and all relevant items (sum of true positives and false negatives). Again translated into this scenario: the ratio of correctly set pixels and the number of correctly set pixels plus the number of pixels that should have been rendered, but which are not covered by the pruned model.

Thus, Precision and Recall (PR) are defined as (true positives tp, false positives fp, and false negatives fn):

$$P = \frac{tp}{tp + fp} \quad \text{and} \quad R = \frac{tp}{tp + fn}. \tag{1}$$

Table 1 gives an overview of the relevant pixels sets, which are shown in Fig. 1 for a simple example. More formally, we denote the set of all pixels covered by the original model as P_{orig}, and the set of pixels rendered for the simpli-

set	pixels that are...
true positives	...correctly set, i.e. rendered for the original and for the simplified model.
false positives	...wrongly set, i.e. rendered only for the simplified model.
false negatives	...rendered for the original model, but not covered by the simplified one.

Table 1: Classification of pixels for Precision and Recall.

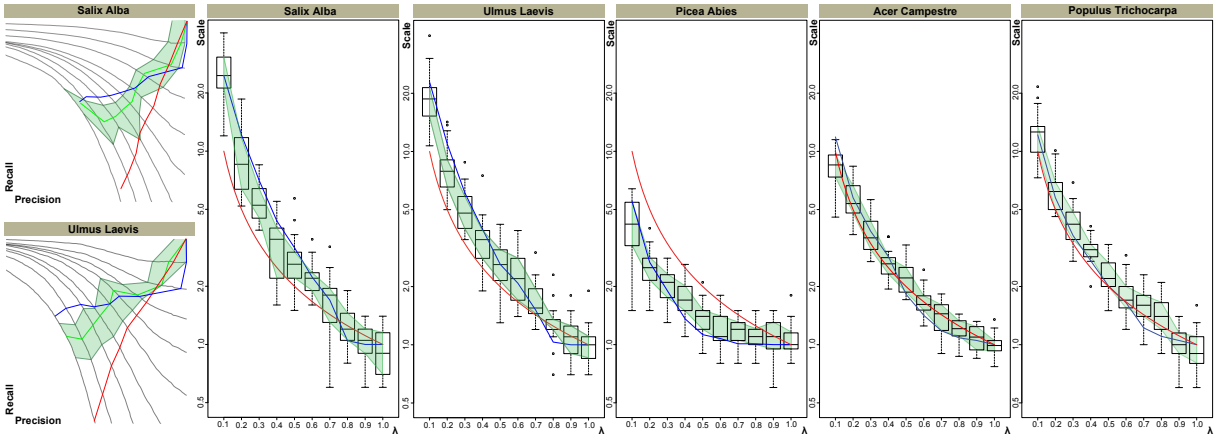

Figure 5: *Comparison between user preferred scaling values indicated in green (user median, first and third quartile), scaling according to cook (red), and scaling predicted by our method (blue). The λ-values of the PR-diagrams (left) are sampled with step size* 0.1 *starting at* $\lambda = 0.1$. *The diagrams on the right show the user selected scaling values for the same λ-levels as on the left. While the user preferred scaling values for the Picea Abies model are significantly lower than the suggested Cook scaling, the scaling values for Salix Alba and Ulmus Laevis are higher. In all five cases the optimal scaling values predicted with our method are in close range to the user preferred values.*

The study revealed that in particular for little simplification ($\lambda > 0.8$) the user-preferred scaling values were not only considerably different for every model, but also on average smaller than the scaling values computed according to Cook et al. [CHPR07]For smaller values of λ the standard deviation of the preferred scaling increased considerably, however, the median value was typically very close to our s_{opt}. The large standard deviation can be explained by the fact that these geometry levels are actually only used to render trees at large distances, while the model presented in the user study was rendered at full size. This obviously makes it harder to judge the appearance of the model and led to larger deviations in the preferred scale value.

Fig. 3 and Fig. 5 show the results of the user study. The preferred scaling values are shown in the respective PR diagram in green, next to s_{Cook} and s_{opt}. On the left in Fig. 5 the values are shown in a PR diagram and additionally the scaling values with respect to the different λ values on the right. For two models the user selected values are in general larger than s_{Cook}, for the Picea Abies model the user selected values are smaller; in all five cases our method faithfully predicts suitable scaling values.

2.4. Impact of Scaling and View Direction

Precision and Recall are defined in image space and therefore view-dependent measures. However, our experiments indicate that for natural objects, which we target in our work, the deviations in the measure are very small. This is because such objects, e.g. trees, typically do not have a dominant view direction but rather uniformly distributed normals and vertex positions. This was also confirmed by our user study where we analyzed user-selected scaling values for two different views of every model and geometry level (Fig. 6). For both views, the variance and mean are very close (Fagus Sylvatica: $s(\lambda = 0.2, V_1) = 6.89$ and $s(\lambda = 0.2, V_2) = 6.75$; $s(\lambda = 0.6, V_1) = 1.94$ and $s(\lambda = 0.6, V_2) = 1.81$. The variance analysis (ANOVA) of the user-study data indicates that the hypothesis (H_0: mean is the same for both views) can be accepted with $F_{V1} = 0.07$ and $F_{V2} = 0.62$ for Fagus Sylvatica, and $F_{V1} = 2.66$ and $F_{V2} = 0.14$ for Picea Abies, well below the critical F-Value of $F_{crit}(1,36) = 4.11$ for $\alpha = 0.05$.

A very important property of the PR measure, and thus for the application of our method, is that PR scores are invariant to rendering the models at different screen sizes, i.e. scaling the model without changing λ. This can be explained as follows: when reducing the size of a (pruned) model it is more likely that multiple triangles are projected onto the same pixels, and thus it is also more likely that all pixels covered by the original model and also covered by the pruned one.

2.5. Detail Level Selection

Rendering complex scenes is only possible if we reduce the level of detail for distant trees and only render with high quality when trees are close to the camera. Using the PR-score from Sect. 2.2, we define the quality Q of a rendering as the distance of the PR-vector to the optimal value $(1,1)$:

$$Q(\lambda, s) = 1 - \sqrt{(1 - P(s,\lambda))^2 + (1 - R(s,\lambda))^2}. \quad (4)$$

For rendering we want to ensure that a tree at a certain distance, d, to the camera will be rendered at a given minimum quality. This means that the PR-vector for a model rendered

with a geometry level $\lambda(d)$ has to be within a certain proximity to the top-right corner of the PR-diagram (see Fig. 7). That is, for rendering we need to sample and store the function $\lambda(d)$ to provide this desired minimum quality for a given d. Note that determining this function takes place in a precomputation step for every tree model.

There are various options to define minimum quality, e.g. letting the user define a given maximum deviation from the optimal PR-scores and computing $\lambda(d)$ accordingly. To compare our method to Cook et al.'s, however, we determine $\lambda(d)$ such that the rendering quality of our PR-optimized pruning and scaling matches their quality for the same distance d. That is, we render a model with the same PR-scores, but with less geometry if possible.

This works as follows: Cook et al. use $\lambda_{Cook}(d) = (1-d)^2$ (with d normalized to $[0;1]$) as a simple relation of distance and geometry. Rendering the model with this pruning yields a quality $Q(\lambda, s_{Cook})$. Next, we determine the smallest λ_{opt} whose rendering with the optimal scaling $s_{opt}(\lambda_{opt})$ (Sect. 2.2) yields equal or better quality, i.e. $Q(\lambda_{opt}, s_{opt}) \geq Q(\lambda, s_{Cook})$. This compound mapping yields a λ_{opt} and an associated s_{opt} for a given view distance d. Obviously this precomputation can only be carried out for a finite number of values. Therefore we use 10 equidistant samples in $[0;0.1)$ and $[0.1;1.0)$, respectively, and linearly interpolate λ_{opt} from the stored samples.

Fig. 7 shows the mapping of distance to pruning for three different tree models. The plots show that model (a) and (c) can be rendered with high quality (PR-score within the second circle depicted in Fig. 7) even for a low value of $\lambda = 0.2$. For model (b) a higher λ-value is required even for larger viewing distances. The plots of rendering with higher λ-values reveal that model (b) and (c) suffer stronger from pruning, while model (a) preserves most of the rendered pixels of the original model.

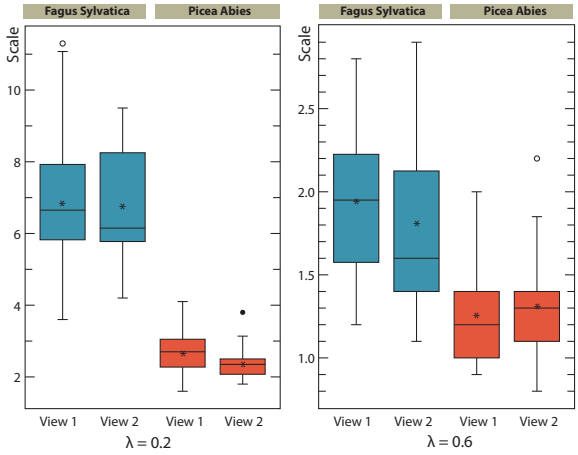

Figure 6: *Comparison of the user selected scaling values for two different tree models and two geometry levels (left: $\lambda = 0.2$, right: $\lambda = 0.6$). For the Fagus Sylvatica model the users preferred larger scaling values (independent of the viewing direction) compared to the Picea Abies model. Besides the view independence the large difference in preferred scaling values for different plant models is significant and underlines the need for model dependent scaling. Outliers are indicated as circles, and the user average as stars.*

3. Rendering Priority

The rendering priority reflects the order in which geometry is removed from the original model with decreasing λ. Cook et al. [CHPR07] tried to avoid correlation in the rendering priority between order and position, size, surface normal, and color as much as possible, and thus prevent disturbing artifacts when rendering with reduced geometry. However, they stated that in some cases the priority order might be found procedurally. In this section we will propose different ways to find the rendering priority order algorithmically in a way that ensures higher Precision and Recall values. Note that it is only the PR-scores that make it possible to compare different prioritization heuristics.

3.1. Silhouette Preservation and Density Normalization

In order to optimize the rendering priority we need to determine which parts of a model are close to the boundary and will potentially be part of the silhouette, and which regions exhibit a high or low density of geometry.

To this end, we will need to define what the "boundary" of a (botanic) model is. For this, we use implicit surfaces that tightly enclose a model. They have also been used to generate normal distributions for such models that provide more realistic and expressive illumination of foliage [LBD07]. Implicit surface can be generated using metaballs [Bli82]: first, a set of generation points P is chosen and an influence radius r_i is assigned to every point. As generation points we use the center of leaf-triangles and choose influence radius proportional to the overall plant height (5% in our case). The contribution of a single generator point $p_i \in P$ at a point in space, q, to the global density function is defined as:

$$D_i(q) = \left(1 - \|q - p_i\|^2 / r_i^2\right)^2.$$

The sum over the contributions of all p_i yields the global density function: $F(q) = \sum_i D_i(q)$. An iso-surface is then defined by a given iso-value a with $F(q) = a$, and can be triangulated using marching cubes (see inset). To extract a tree's tight hull we choose an iso-value such that $a = 0$.

In the following we will discuss different prioritization heuristics based on the global density function. Again, we measure the quality using PR-scores.

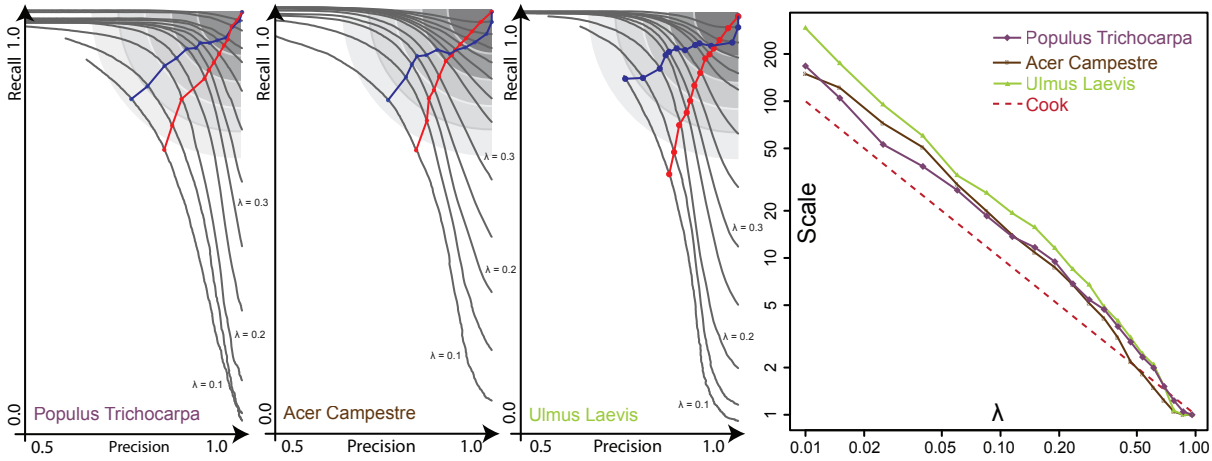

Figure 7: *The first three graphs show PR-Values for three different tree models. The blue graph shows the optimal scaling values s_{opt}, the red one the standard scaling values s_{Cook} (according to Cook et al.). For rendering we choose λ such that we maintain a minimum quality that is required for a given viewing distance. The minimum quality requirements are indicated by the circles centered at the top-right corner of the PR diagram. As we can see, it is possible to reduce the geometry to a larger extend for model Populus Trichocarpa compared to model Acer Campestre, in particular for small values of λ. Right: a log-log plot of scale vs. fraction of remaining geometry, i.e. λ. Note that the connected (s_{opt}, λ) and (s_{Cook}, λ) lines in the PR diagram (left three plots) look very rough. The values plotted against the geometry level λ, however, are smooth with the expected exponential behavior. The smoothness is important to avoid popping artifacts during rendering. The λ-values are sampled with 0.05 step size from $\lambda = 0.1$ to $\lambda = 0.4$ and with step size 0.1 above.*

3.1.1. Varying Density

First, we use the global density function to identify regions of high geometric density within a model. Triangles that are close to each other are likely to be projected to the same location in image space. Thus removing triangles in very dense region lowers the probability of overdraw while still keeping chances high that all original pixels are covered even without scaling the remaining geometry. We evaluated this guided geometry prioritization using our PR measure, and experiments showed that the quality improves for high values of λ. For such values, there are larger variations in local density (see Fig. 8), as no, or little, density controlled pruning did take place. These variations obviously vanish when reducing more and more geometry prioritized in dense regions, which makes the density variation become more uniform. When this point is reached, i.e. for smaller λ, we switch back to pure stochastic prioritization. The performance increase due to density prioritized pruning depends on the variance of $F(q)$ within a model, and thus models with almost uniform density do not benefit from this strategy.

3.2. Orientation

As second heuristic we investigated the improvement of rendering prioritization based on the deviation between a triangle's normal and the normal on the nearest point on the implicit surface (denoted as α). Preserving geometry facing outwards, i.e. small α, generally enforces a pixel coverage of the rendered model that is closer to that of the full detail model. Scaling triangles that resemble the models isosurface turned out to perform well, especially for small values of λ (see Fig. 8).

Pruning triangles close to the implicit surface with lower probability, however, leads to inferior results. While at first sight it seems reasonable to preserve the silhouette, keeping and scaling triangles close to the surface results in many false positive pixels, in particular for strong scaling with small λ. This leads to visible artifacts and low Precision and Recall scores.

3.3. Combined Prioritization

Both heuristics determine "survival probabilities" for the triangles of a model. In our implementation we use an empirically found weighting to combine both of them, where the orientation heuristic has smaller impact. We choose $P_{Combined} = (P_{Silhouette})^2 + P_{Density}$ normalized to $[0,1]$ and then sort the triangles for descending survival-probability (adding a small amount of randomness) to obtain a single list representing the entire model. Similar to Sequential Point Trees [DVS03], we can then render only a prefix of the list, according to λ, to render a pruned model.

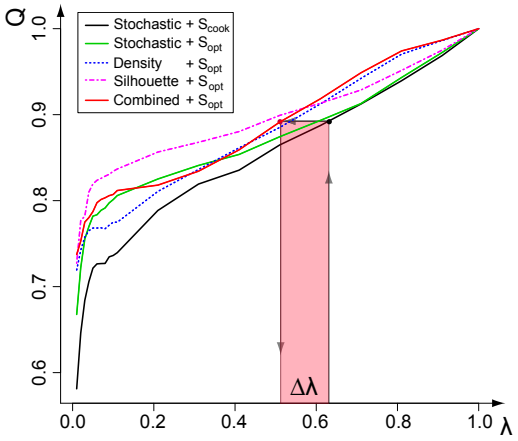

Figure 8: *Comparison of prioritization heuristics to pure stochastic ordering for the Ulmus model. The red area shows the benefit of switching from stochastic order with s_{Cook} to the combined prioritization with s_{opt}. It is possible to maintain the quality with less geometric detail, denoted as $\Delta\lambda$.*

4. View-dependent Optimization

Considering the viewing distance and thus the projected size of a model is of course one important aspect for choosing the geometry level. However, occlusion also has impact on the required detail: partially occluded trees, or trees that are completely surrounded by others, do not contribute considerably to the scene's appearance and thus should be rendered using less geometry. In this section we show how we can determine occlusion of trees at run-time, and control our rendering accordingly.

For this we analyze the occlusion of each tree by testing the visibility of a set of sample points distributed on its iso-surface. In principle there are various possibilities to perform the visibility test, e.g. ray casting, or using some form of precomputed visibility information. To facilitate real-time rendering without precomputation, we use an image space approach relying on the depth buffer of the camera image only. Obviously this depth buffer is not available before actually rendering the geometry. However, if we assume smooth camera movement we can exploit frame-to-frame coherency and test the visibility of sample points using the depth buffer and transformation of the previous frame. For abrupt movements, or sample points that are projected outside the viewport, we conservatively assume full visibility and thus render the models at possibly higher detail than actually necessary.

From the visibility of the sample points we then deduce an approximate occlusion factor for each tree. The fraction f of visible to the total number of sample points is used to control λ, in a way such that $\lambda_{occl} = \lambda_{opt}(d) \cdot \max(f, 0.1)$. To avoid popping artifacts, we use a hysteresis function and smooth the λ_{occl}-values over time.

5. Color Variation

Our method is suitable for rendering (groups of) objects that are aggregated from a large number of randomly oriented and placed geometric details. Typically we can also assume a near uniform color distribution, or large-scale gradations, for botanical objects. Cook et al. [CHPR07] intentionally do not correlate the rendering priority order to the color distribution or any other model characteristics. Apart from color variation Cook et al. proposed a method to preserve color contrast during simplification. In this section we show the effect of our pruning algorithm on three different kinds of color variations (Fig. 9). While large colored regions across the objects are preserved (Fig. 9, top), smooth color gradations show the effects similar to quantization artifacts, which is expected due to the geometry reduction (bottom). The pruning becomes most apparent for small, randomly distributed and salient details (red leaves in Fig. 9). Under strong pruning the fraction of covered pixels is still preserved, but the distribution becomes less random due to the smaller number of samples (bottom right). Note that all these artifacts become less apparent if the model is rendered with a size according to the geometry level. Color variations between models are of course preserved, as we do not prune across objects, and thus individual trees can still be identified (Fig. 11).

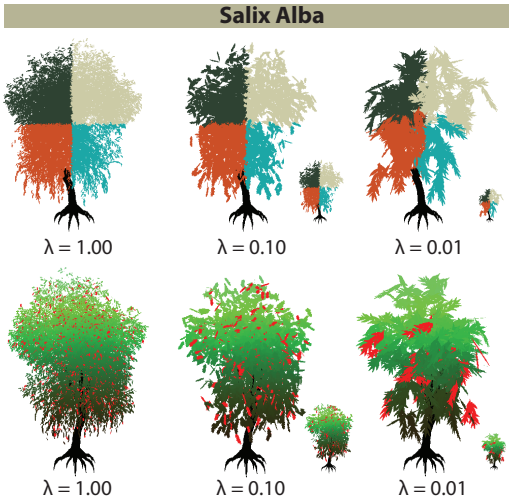

Figure 9: *Effect of model simplification to intra model color variation: boundaries of large colored regions are maintained even for low detailed models (top row). Smooth gradations do not exhibit noticeable changes under stochastic pruning, apart from effects similar to color quantization. The arrangement of small details (e.g. the red leaves) obviously changes, but still does not cause flickering. Note that strong pruning typically occurs for distance trees.*

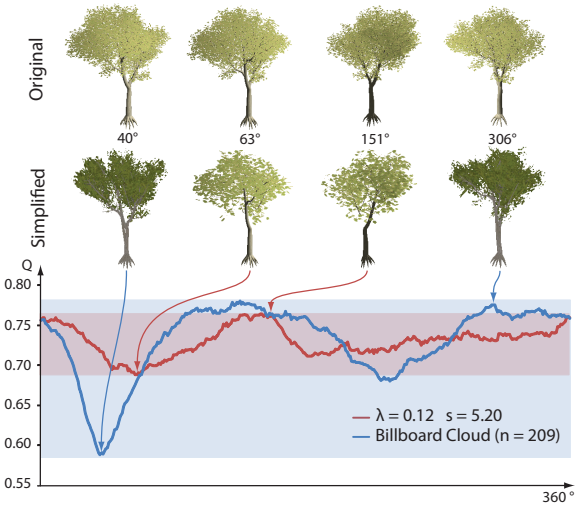

Figure 10: *Comparison of rendering a 360° rotation of a billboard cloud model (209 billlboards) and a pruned model ($\lambda = 0.12$, $s = 5.2$). The parameters for the latter are chosen to match the average quality Q of the billboard model. Note that the variance for the billboard model is much higher.*

6. Results and Comparison

In this section we present results of our method and comparisons to Cook et al.'s [CHPR07] method to assess rendering performance, and to billboard clouds [DDSD03] to demonstrate the benefits of our (view-dependent) pruning over texture-based representations. We implemented our method using OpenGL and performed all tests and measurements using an Intel Core i7 at 2.8Ghz, with 4GB of memory, and a NVIDIA Geforce GTX 295 GPU.

Comparison to Billboard Clouds For rendering botanical models most real-time applications resort to a representation with relatively few textured polygons recreating the original model. For video games these models are often created manually, while billboard clouds [DDSD03] can be used to obtain such reduced models automatically. Note that these representations do not provide a "continuous" level of detail and switching between different levels is prone to popping artifacts. We compared our results to billboard clouds by evaluating the rendering quality according to our measure Q (Sect. 2.5). We observed that the rendering quality varies strongly with the view direction when using billboard clouds, and significantly less with our optimized pruning. Fig. 10 shows this comparison where the parameters of our pruning are adjusted to match the average quality of a billboard representation. Note that another applications of our PR-measure can be the billboard cloud generation itself, where it can be used to identify bad views for which the billboard representation needs to be improved.

Rendering Performance Our method allows us to render complex scenes with 5000 tree models at interactive to real-time rates, i.e. 8 to 25 frames per second at a resolution of 1600×1200 (Fig. 13). The full-detail geometry of the scene consists of more than 1.3 Billion vertices and renders at only 0.8 frames per second on the same hardware, i.e. far from interactive speed. Our optimized and prioritized pruning, together with the view-dependent visibility tests reduces the number of vertices per frame to about 26 million vertices. On average this yields a performance increase, compared to Cook et al., of approximately 60-70% while maintaining the same quality (determined using the PR scores). Fig. 11 shows a complex scene with exaggerated color variation, individual tree models can still be identified even for low geometry levels (top layer). The bottom two layers visualize the color coded geometry level. While λ_{Cook} is choosen depending on the camera distance λ_{Opt} is individually choosen for each tree model according to Sec. 2.5. Fig. 12 shows a detailed evaluation of a standard camera path through a scene with 1276 trees without culling following the equal quality approach. While choosing an optimal scaling value with stochastic prioritization (S_{opt}, Fig. 12 green graph and Tab. 2) already gives on average a 10 percent geometry reduction (choosing a lower geometry level with equal PR-scores), additionally changing the prioritization according to the proposed heuristics gives an average geometry reduction between 17 and 40 percent (see Tab. 2). Although the combined prioritization performs better for close views (see Fig. 8) for most camera positions in the test scene with a mixture of distant and close models the silhouette-based prioritization performs best.

	Stochastic $+S_{opt}$	Density $+S_{opt}$	Silhouette $+S_{opt}$	Combined $+S_{opt}$
Avg	0.101	0.174	0.407	0.301
Min	0.043	0.022	0.308	0.014
Max	0.437	0.236	0.693	0.392

Table 2: *Geometry reduction with our different prioritizations (Fig. 12) compared to Cook et al. [CHPR07]. For example, the combined heuristic requires 30.1% less geometry on average to render at the same quality as Cook et al.*

7. Conclusions and Future Work

In this paper we introduced Precision and Recall as a measure of quality for rendering complex geometry with pruning. We further improved on previous methods by applying model specific geometry reduction and optimized scaling as well as view-optimized pruning. We evaluated our method by means of a user study which indicates a considerable improvement compared to naive and purely stochastic pruning. However, our work also raises new questions. One interesting direction of future research is to consider more than just correct and incorrect pixels in PR, e.g. by accounting for de-

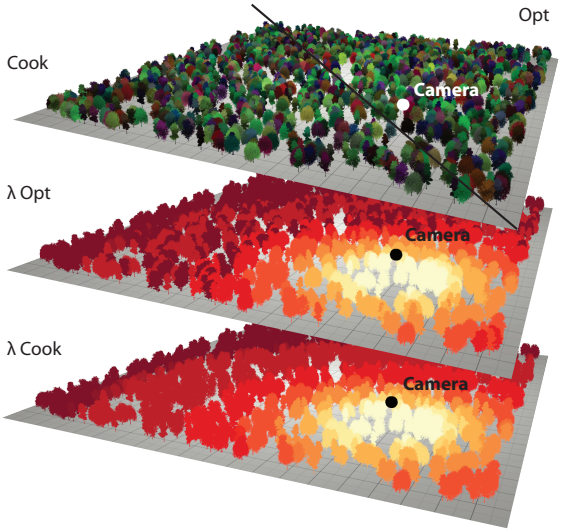

Figure 11: *A scene with intentionally exaggerated high inter model color variance. Even for distant areas with a low amount of geometry individual tree models can be identified (top layer). Bottom two layers: Geometry distribution according to Opt and Cook.*

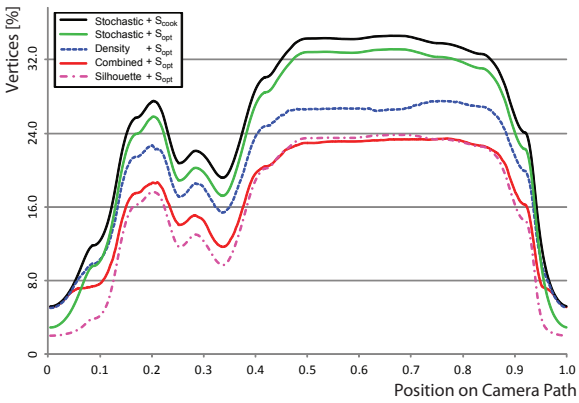

Figure 12: *Performance evaluation of different heuristics for a camera path through a scene with 1276 tree models without culling (∼ 63M vertices). It can be seen that the silhouette based prioritization performs best for distant views (start and end of the camera path). While the combined prioritization performs slightly better in walk through camera positions.*

viations in the normals, measuring contrast and color differences, or to evaluate how visible differences predictors can improve the measure and whether their use amortizes.

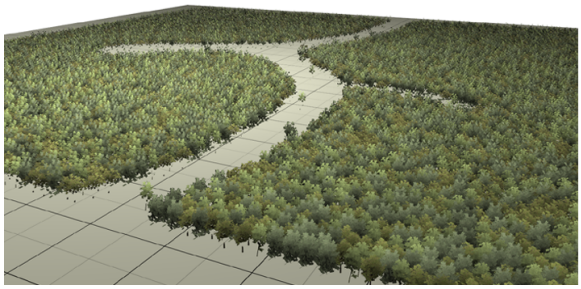

Figure 13: *This scene consists of 5000 trees (in total 1.3 Billion vertices). With our optimized pruning we can render this scene with 15 frames per second when all 5000 trees are visible. Pruning according to Cook et al. [CHPR07] renders an image of the same quality at 9 frames per second.*

8. Acknowledgements

The authors would like to thank the reviewers for their insightful comments. This work is supported by DFG Research Training Group GK-1042 "Explorative Analysis and Visualization of Large Information Spaces", University of Konstanz and the project "Information at your Fingertips Interactive Visualization for Gigapixel Displays", which is supported by the "Information Technology Baden-Württemberg (BW-FIT)" funding program.

References

[Bli82] BLINN J. F.: A generalization of algebraic surface drawing. *ACM Transaction on Graphics 1*, 3 (1982), 235–256. 6

[CHPR07] COOK R. L., HALSTEAD J., PLANCK M., RYU D.: Stochastic simplification of aggregate detail. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007) 26*, 3 (2007), 79. 1, 2, 4, 5, 6, 8, 9, 10

[DBD*07] DRETTAKIS G., BONNEEL N., DACHSBACHER C., LEFEBVRE S., SCHWARZ M., VIAUD-DELMON I.: An Interactive Perceptual Rendering Pipeline using Contrast and Spatial Masking. In *Rendering Techniques 2007 (Proceedings of EGSR)* (2007). 2

[DCSD02] DEUSSEN O., COLDITZ C., STAMMINGER M., DRETTAKIS G.: Interactive visualization of complex plant ecosystems. In *VIS '02: Proceedings of the Conference on Visualization '02* (2002), pp. 219–226. 2

[DDSD03] DÉCORET X., DURAND F., SILLION F. X., DORSEY J.: Billboard clouds for extreme model simplification. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003) 22*, 3 (2003), 689–696. 1, 2, 9

[DN04] DECAUDIN P., NEYRET F.: Rendering forest scenes in real-time. In *Rendering Techniques '04 (Proceedings of EGSR)* (2004), pp. 93–102. 2

[DN09] DECAUDIN P., NEYRET F.: Volumetric billboards. *Computer Graphics Forum 28*, 8 (2009), 2079–2089. 2

[DVS03] DACHSBACHER C., VOGELGSANG C., STAMMINGER M.: Sequential point trees. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003) 22*, 3 (2003), 657–662. 2, 7

[GCRR10] GUMBAU J., CHOVER M., REMOLAR I., REBOLLO C.: View-dependent pruning for real-time rendering of trees. *Computers & Graphics to appear* (2010). 2

[GSSK05] GARCIA I., SBERT M., SZIRMAY-KALOS L.: Leaf cluster impostors for tree rendering with parallax. In *EG Short Presentations* (2005), pp. 69–72. 1, 2

[KKF*04] KLEIN J., KROKOWSKI J., FISCHER M., WAND M., WANKA R., HEIDE F. M. A. D.: The randomized sample tree: A data structure for interactive walk-throughs in externally stored virtual environments. *Presence: Teleoper. Virtual Environ. 13*, 6 (2004), 617–637. 2

[LBD07] LUFT T., BALZER M., DEUSSEN O.: Expressive illumination of foliage based on implicit surfaces. In *Eurographics Workshop on Natural Phenomena* (2007). 6

[LEST06] LACEWELL J. D., EDWARDS D., SHIRLEY P., THOMPSON W. B.: Stochastic billboard clouds for interactive foliage rendering. *Journal of Graphics, GPU, and Game Tools 11*, 1 (2006), 1–12. 1, 2

[LWC*02] LUEBKE D., WATSON B., COHEN J. D., REDDY M., VARSHNEY A.: *Level of Detail for 3D Graphics*. Elsevier Science, 2002. 2

[Rij79] RIJSBERGEN C. J. V.: *Information retrieval*, 2 ed. Butterworths, London, 1979. 3

[RL00] RUSINKIEWICZ S., LEVOY M.: QSplat: A multiresolution point rendering system for large meshes. In *SIGGRAPH '00* (2000), pp. 343–352. 2

[RRCR06] REBOLLO C., REMOLAR I., CHOVER M., RIPOLLÉS O.: An efficient continuous level of detail model for foliage. In *Proceedings of 14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision* (2006). 2

[SD01] STAMMINGER M., DRETTAKIS G.: Interactive sampling and rendering for complex and procedural geometry. In *Rendering Techniques 2001 (Proceedings of Eurographics Workshop on Rendering* (2001), pp. 151–162. 2

[SSK96] SCHAUFLER G., STÜRZLINGER W., KEPLER J.: A three dimensional image cache for virtual reality. *Computer Graphics Forum 15*, 3 (1996), 227–236. 1, 2

[SW08] SCHERZER D., WIMMER M.: Frame sequential interpolation for discrete level-of-detail rendering. *Computer Graphics Forum (Proceedings EGSR 2008) 27*, 4 (June 2008), 1175–1181. 2

[WFP*01] WAND M., FISCHER M., PETER I., MEYER AUF DER HEIDE F., STRASSER W.: The randomized z-buffer algorithm: interactive rendering of highly complex scenes. In *SIGGRAPH '01* (2001), pp. 361–370. 2

[WS02] WAND M., STRASSER W.: Multi-resolution rendering of complex animated scenes. *Computer Graphics Forum (Proceedings of Eurographics 2002) 21*, 3 (2002). 2

[ZP08] ZHANG Q.-L., PANG M.-Y.: A survey of modeling and rendering trees. Edutainment 2008, Eds. Z. Pan et al., 2008. 2