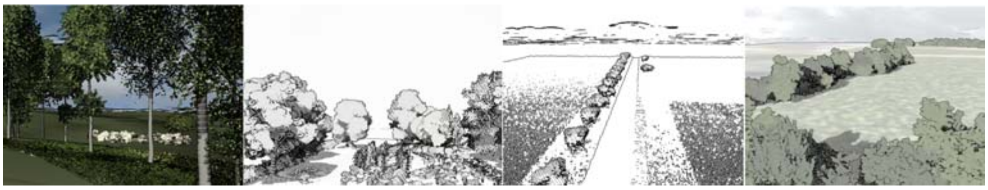
Seamless Integration of Stylized Renditions in Computer-Generated Landscape Visualization

Liviu Coconu[1], Carsten Colditz[2], Hans-Christian Hege[1] and Oliver Deussen[2]



Abstract

We propose enhancements and generalizations of 3D real-time computer visualization techniques used in landscape planning and architecture. As opposed to pure photorealism, illustration and non-photorealistic (NPR) techniques can be particularly efficient in conveying and transmitting selected visual information. By combining the power and accuracy of photorealistic models with the flexibility of abstract sketches, we have developed a flexible environment for the automatic creation of sketchy real-time rendering of landscapes. This allows for freedom of choosing the style and degree of abstraction of the visualization. All elements that contribute to the picture are seamlessly integrated and parameterized, including the transition from photorealism to non-photorealism. We expect a significant improvement of computer-aided landscape visualizations in our approach as such renditions are no longer limited to photorealism which often entails too much detail.

1 Introduction

3D real-time computer visualization has evolved into a practical alternative to traditional hand-drawn presentation techniques in landscape planning and architecture. A central question remains, however: which visualization methods are best suited for the particular purposes of this field. Accurate plant and vegetation models allow good photorealistic visualizations that have been the goal of recent research. However, as pointed out by potential users, photorealism is not always able to offer the necessary flexibility in selectively emphasizing essential information, which is a major goal in practice.

In contrast, illustration and non-photorealistic (NPR) techniques can be particularly efficient in conveying and transmitting selected visual information. Significant progress has been made late in this field of computer graphics, but existing algorithms are usually limited to specific applications. On the other hand, sketched drawings are the traditional method for rendering in landscape applications. The starting point in our work has been the

[1] Zuse Institute Berlin (ZIB), Germany
[2] University Konstanz, Germany

idea of enhancing the palette of available landscape visualization techniques by combining the power and accuracy of photorealistic models with the flexibility of abstract sketches.

We developed a flexible environment for automatic creation of sketchy real-time rendering of landscapes where the user has the freedom of choosing the style and degree of abstraction of the visualization himself. For this purpose, all elements that contribute to the picture are seamlessly integrated and parameterized, including the transition from photorealism to non-photorealism. By simply using sliders for changing a small set of parameters, a whole range of styles can be created. The ultimate goal is to allow the user to develop a suitable "visual language" for creating storytelling presentation with high 3D quality.

From this approach, we expect a significant improvement of computer-aided landscape visualizations, as such renditions are no longer limited to photorealism with all its drawbacks. This has been previously a major objection against 3D landscape visualization. Further enhancements are possible, based on the feedback from potential users that we hope to receive.

2 Previous Work

To our knowledge, an attempt similar to ours to enrich real-time 3D presentation techniques for landscapes has not been made before. Of course, we build on previous work in several research domains. Modeling and rendering complex plant models has always been a challenging research area. 3D plant models can be created either using L-system approaches (Prusinkiewicz Lindenmeyer 1990, Mech Prusinkiewicz 1996) or interactive approaches (Xfrog plant modeler by Lintermann Deussen 96, Deussen Lintermann 97). In Deussen et al 98, impressively complex landscape scenes with realistic plants are modeled and rendered. Because of the high complexity, level-of-detail approaches and the use of alternative primitives like points, lines and billboards are required to accomplish rendering in real-time (Deussen et al 2002, Coconu Hege 2002). All these approaches using high detailed models are targeting photorealistic visualizations.

In contrast, non-photorealistic graphics attempt to create sketchy, abstract views, analogue to hand-drawn images. This has been an active field of research lately, methods and algorithms are usually tailored for specific applications, however. Algorithms for detection of silhouettes other feature lines have been developed. Augmenting photorealistic visualizations with sketchy elements like contours has been used in Hertzmann 2000 and Isenberg 2003. In Praun et al 2001, a method for real-time hatching of smooth objects has been developed. For trees, the representation of foliage with small, abstract primitives (disks) has been used in Reeves and Blau 1985 and inspired later Kowalski et al. 1999. The most representative related work for this paper are Deussen at al 2000 and more recently Deussen 2003, where abstraction and rendering mechanisms specific for plants are derived from traditional hand-drawn imagery to create pen-and-ink illustration of landscapes.

The novel ideas presented in this paper are several computer graphics techniques for abstract rendering of plants (contours, leaf abstraction, hatching and lighting). Further, at a higher level, we seamlessly integrate and extend previous work together with our new contributions in a flexible framework capable of adapting the visualization appearance to the needs of the user. The rest of the paper is organized as follows: we briefly describe the general modeling and rendering pipeline of the landscape visualization system (Lenne3D) we use. Then, in Section 4, we derive drawing elements from traditional hand-drawn illustrations. The implementation and integration of different illustration elements are described in Sections 5 and 6. We then present current results and our conclusion in Section 7.

3 Pipeline

In order to be useful in practice, any landscape visualization technique must meet certain requirements. Our major criteria are:

- real-time capability
- controllable degree of abstraction and appearance
- use of available data.

We built on the real-time landscape visualization system Lenne3D, so that we benefit from its whole automated visualization pipeline, as depicted in Figure 2.

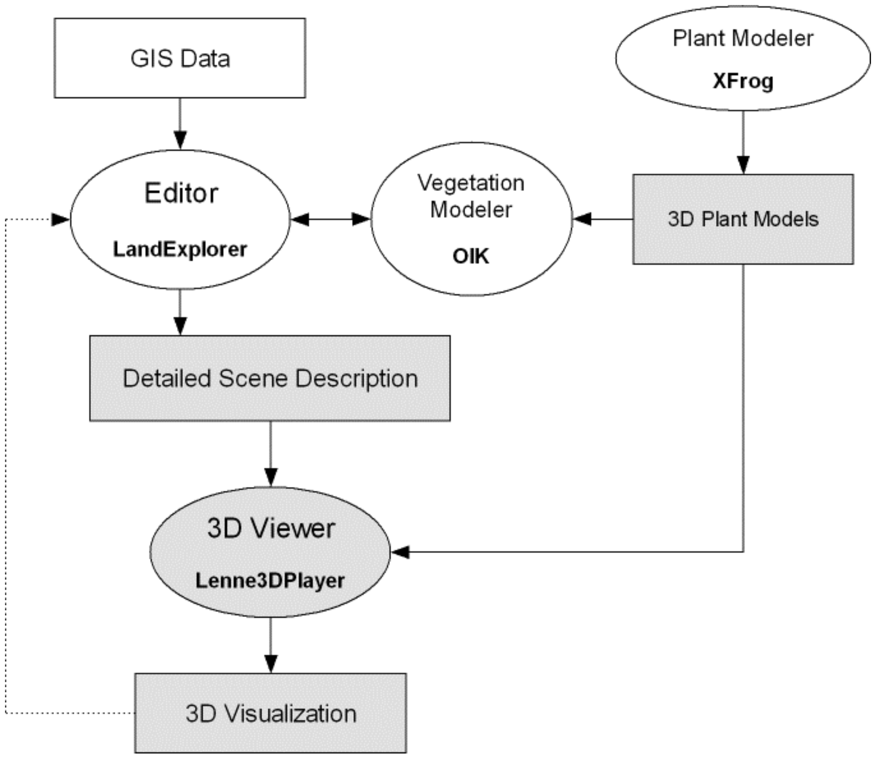


Fig. 2: The Lenne3D visualization pipeline.

4 Traditional Illustration

When designing visualization techniques for landscapes, it is helpful to analyze the numerous collections of hand-drawn plant illustrations, since this used to be and, in a certain degree, still is, the presentation method of choice for landscape planners and architects. We believe that traditional, sketchy presentation styles combined which are automatically generated in 3D out of accurate models are better suited for many practical purposes.

In Deussen 03, besides an overview of existing non-photorealistic rendering techniques in the computer graphics, a classification of the traditional drawings form a technical point of view is given - mainly based on a sketch collection by Evans 96. Such insights of plant illustration, which we also use in this work, are very useful to derive computer graphics techniques that implement sketchy drawings. The novelty is that in this work all drawing elements and styles are functionally integrated together, rather than separately implemented. The main characteristics of a plant illustration are:

- difference in drawing between trunk and foliage, which allow for different abstraction mechanisms. While foliage tends to form compact structures (”clouds”) of leafs, the trunk keeps it’s shape and can be abstracted using lines.
- the overall figure can be communicated in two ways: figure abstraction and visual agglomeration. The former technique renders the plant using few simple lines (contours), while the latter makes use of many small primitives for the leaves to define the plant foliage.
- usage and representation of light and shadow. In botany, only the pure shape of a plant is drawn, whereas in architectonic applications light and shadows are important as they bring spatial and relational information in the scene. There are again two main ways to represent shadows and light: detail variation and hatching. The latter can be simultaneously used to reveal shape.

5 Implementation of Illustration Elements

Having in mind the aforementioned illustration principles, our sketchy rendering framework combines several drawing elements, which are described in this section, together with parameters that allow the user to control the appearance and abstraction degree, as well as to combine the different elements together. The requirements mentioned in Section 3 must be fulfilled, the most notable and restrictive being the real-time capability: the computer graphics counterpart of each drawing element must be able to run at interactive frame rates for complex scenes.

Silhouettes and contours are perhaps the most important visual clues of a plant. For contour detection, we enhanced an image space technique using the depth buffer commonly available on graphics hardware. The scene is being rendered with depth buffer (at each pixel, the depth buffer contains the distance from observer to the first visible object part in the scene) and then we apply a carefully tuned mathematical operator on the depth buffer to detect discontinuities in the depth values. Wherever the discontinuity exceeds a certain threshold, we have a contour pixel. The method is fast and does not depend on the complexity of the scene but only on the image resolution. A nice intrinsic feature is that, due to the non-linear distribution of values in the depth buffer, it detects more detail for closer parts of the scene and "abstracts out" the far parts (see Deussen 2000). By varying the threshold for the discontinuity, we bring more or less contours in the image, thus controlling the abstraction degree.

We improved the basic technique in several ways:

- better noise filtering (better coherence) by removing isolated flickering pixels
- seamless adjustment of contour thickness within small limits; a contour thickness of zero means that no contours are drawn.
- combination with model abstraction.

.

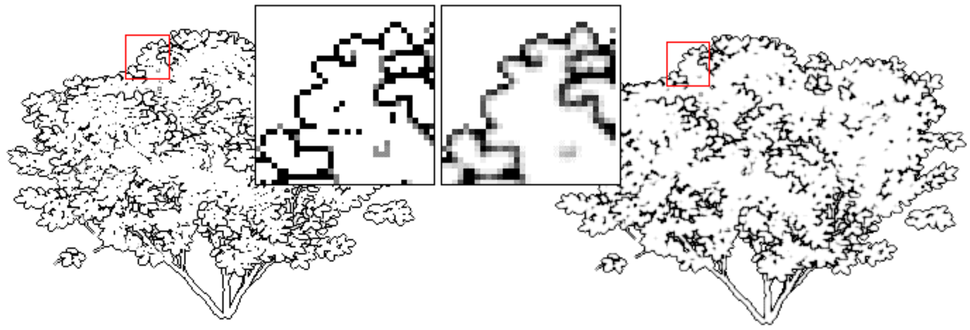


Fig. 3: Contour drawing.

Leaf primitives. The contour detection is capable of rendering both the trunk and the leaves of a plant - actually, the results depends only on whatever we draw into the depth buffer. While the original complex geometry is suitable for the trunk, drawing the leaves in the photorealistic complexity usually results in too much visual agglomeration. Fortunately, one of the level-of-detail mechanisms employed in Lenne3D (Deussen et al 2002) is suitable for abstraction, as it progressively reduces the number of leaves with the distance and keeps the leaf size approximately constant. One parameter we can seamlessly set is thus the leaf primitive size, usually much larger than for photorealistic rendering. Another abstraction mechanism is the shape of the leaf: in the present system we can seamlessly pass over from the complex photorealistic shape to an abstract disk or ellipse, but virtually any shape can be used for the leaf, as an alpha-matted texture.

Fig. 4: Seamless abstraction of the leaf shape.

Hatching is a versatile and suggestive element in drawings. Hatching plant scenes in real-time is a challenging task: existing computer graphics algorithms only deal with compact and smooth objects with an underlying parameterization (Praun et al 2001). Another issue is that hatching techniques for plants can be quiet different than those used for other objects. As opposed to hatching on smooth objects, where hatching strokes tend to be aligned to the object's curvature, plants are drawn using more or less straight lines in one direction. For this task, we developed a novel real-time method that is capable of mapping hatch strokes from a texture (tonal art map, Praun et al 2001) in a view-dependent and coherent way onto complex, non-compact objects like trees. The user can seamlessly control the amount of hatching and the tone mapping, resulting in darker or brighter illustration. Different styles of hatching can be easily stored in special textures. Additionally, shape suggestion can be also added, in this case the hatching strokes are redirected to suggest the curvature.

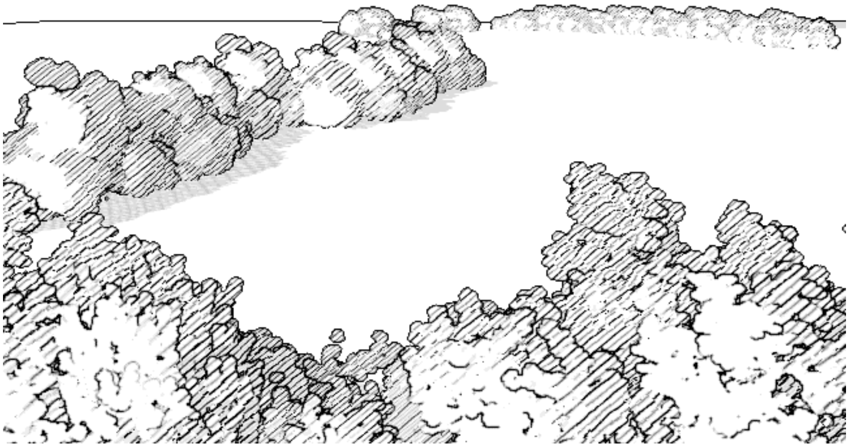


Fig. 5: Hatching.

Light and shadows can be rendered using the two abovementioned techniques: detail variation and hatching. Hatching is quite straightforward: the hatching tone is adjusted according to the tone after lighting and shadow calculations. In order to vary the detail level, it suffices to modulate the contour detection threshold accordingly so that more and thicker contours are drawn in shadow areas. Of course, this can also be seamlessly controlled with a parameter.

6 Seamless Integration

Again we must emphasize one of the most important features of our system: all rendering elements can be seamlessly and independently controlled by the user in order to obtain the targeted scene appearance. As described above, each individual rendering element (contours, primitive shape and size, hatching, light and shadows) can be faded in and out and have several other parameters controlling its appearance. Besides that, sketchy renderings are no longer separated from the photorealism, but rather the user can choose the desired proportion between photorealism and different sketchy styles, as shown in Figure 6.

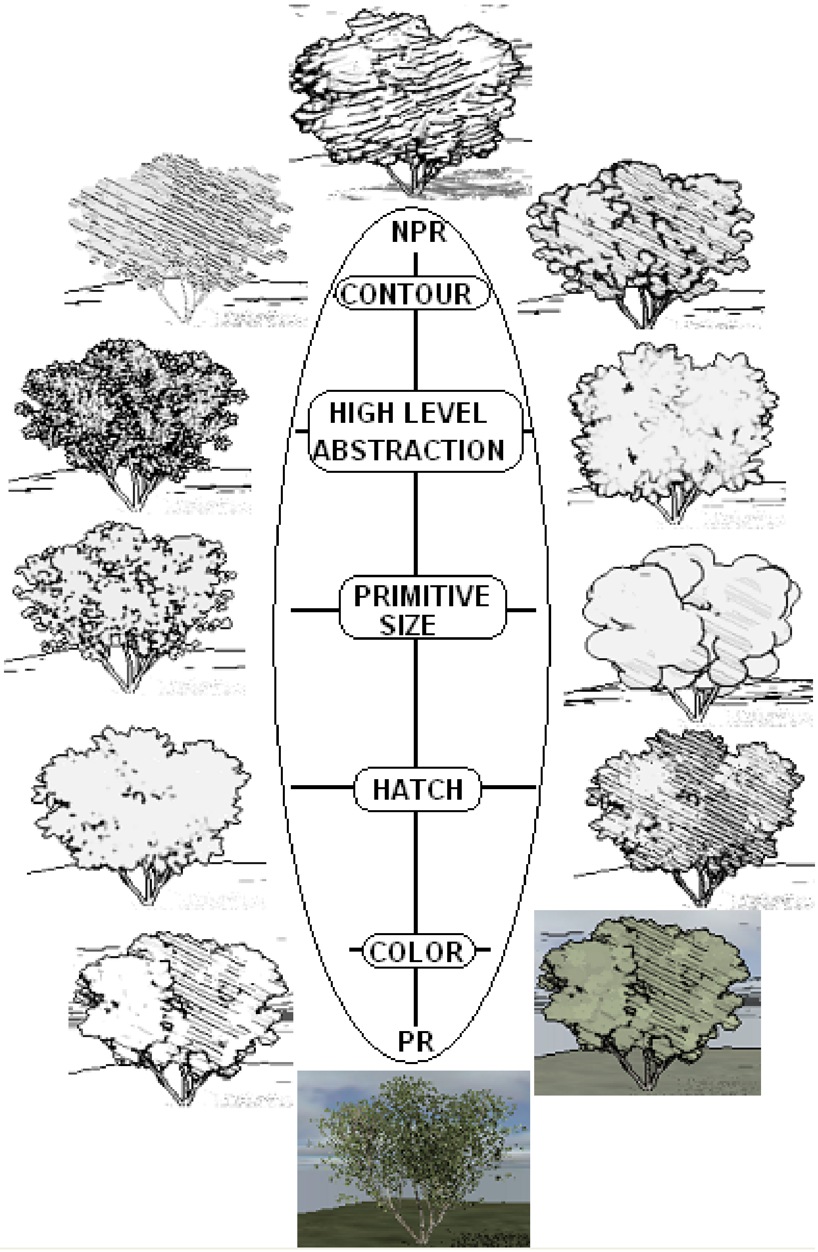


Fig. 6: Most important system parameters.

7 Results, Conclusion and Future Work

Different styles of renditions that can be accomplished with our system can be seen in Figure 7. We demonstrate the usability of the presented techniques with real, complex landscape scenes which have been created with the Lenne3D system. In Table 1 we show that the overhead introduced by the new rendering framework at the same scene complexity is small, making the approach viable in practice. Moreover, depending on the degree of abstraction, sketchy rendering tend to be faster because the number of primitives being rendered is reduced. At some point, the computational burden of sketchy rendering is overbalanced by the reduction of number of primitives.

	# polygons	PR		NPR	
		% rendered	fps	% rendered	fps
1(c)	978,944	29.00	25.0	15.00	12.5
1(d)	9,969,464	3.70	18.0	0.57	11.7
1(b)	81,106,940	0.66	8.6	0.20	6.2
1(a)	87,208,496	2.06	3.2	0.56	6.7

Table 1: Rendering performance.

As this is work in progress, we currently have to deal with some problems and limitations of the system. The image-based contour detection still has some coherence problems (flickering) and is intrinsically limited as expression possibilities: contours are detected as unconnected pixels, so it is not possible to apply line styles. We intend to investigate the possibility to include such variations. Another problem is the insufficient abstraction degree that can be achieved only with the variation of the leaf primitive size. In the future, selective abstraction mechanisms at higher levels are planned.

Of course, our system cannot replace a good artist - this is generally considered true for any automatic computer graphics algorithm. But this was also not the purpose, rather to offer the possibility of interactive 3D navigation through complex landscape scenes with the advantages and flexibility of sketchy visualization, claimed to better suit the needs of landscape planners and architects. It is intended as an efficient instrument in communicating visual information and creating story telling presentation easier than per hand. Its design after the principle of maximum flexibility should allow the user to develop an own application-dependent ”visual language”.

Fig. 7: Continuous abstraction of a scene.

8 Acknowledgements

We thank all collaborators in the Lenné3D project for fruitful discussions and manifold stimuli.
All plant models have been built using Xfrog by Greenworks and converted to the Lenné3D plant format.

9 References

Coconu, L., Hege, H.-C. (2002): *Hardware-Accelerated Point-Based Rendering of Complex Scenes,* Eurographics Workshop on Rendering, pp. 43 - 52.
Deussen, O., Lintermann, B. (1997): *A Modelling Method and User Interface for Creating Plants*, Proc. Graphics Interface 97, Morgan Kaufmann Publishers, pp. 189-198.
Deussen, O., Hanrahan, P., Lintermann, B., Mech, R., Pharr, M., Prusinkiewicz, P. (1998): *Realistic modeling and rendering of plant ecosystems, SIGGRAPH 1998*, pp. 275-286.
Deussen, O., Strohotte, T. (2000): *Computer-generated pen-and-Ink illustration of trees,* Computer Graphics 34(4) (SIGGRAPH 1998), pp. 13-18.
Deussen, O., Colditz, C., Stamminger, M., Drettakis, G. (2002): *Interactive Visualization of Complex Plant Ecosystems*. IEEE Visualization 2002, pp. 219-226.
Evans, L. (1996): *The New Complete Illustration Guide: The Ultimate Trace File for Architects, Designers, Artists, and Students,* Van Nostrand Reinhold.
Hertzmann, A., Zorin, D. (2000): *Illustrating smooth surfaces*. SIGGRAPH 2000 Conf. Proc., pp. 183-192.
Isenberg, T., Freudenberg, B., Halper, N., Schlechtweg, S., Strothotte, T. (2003): *A Developer's Guide to Silhouette Algorithms for Polygonal Models.* IEEE Computer Graphics and Applications, 23(4), pp. 28-37.
Kowalski, M.A., Markosian, L., Northrup, J. D., Bourdev, L., Barzel, R., Holden, L.S., Hughes, J.F. (1999): *Art-Based Rendering of Fur, Grass, and Trees*, SIGGRAPH 1999 Conf. Proc., pp. 433-438.
Lintermann, B., Deussen, O. (1996): *Interactive Modelling and Animation of Branching Structures*, Computer Animation and Simulation ´96, Springer, Berlin, pp. 139-151.
Měch, R., Prusinkiewicz, P. (1996): *Visual models of plants interacting with their environment*, SIGGRAPH 96 Conf. Proc., pp. 397-410.
Praun, E., Hoppe, H., Webb, M., Finkelstein A. (1996): *Real-Time Hatching*, SIGGRAPH 2001 Conf. Proc., pp. 579-584.
Prusinkiewicz, P., Lindenmeyer, A. (1990): *The Algorithmic Beauty of Plants,* Springer, New York.
Praun, E., Hoppe, H., Webb, M., Finkelstein A. (1996): *Real-Time Hatching*, SIGGRAPH 2001 Conf. Proc., pp. 579-584.
Reeves, W.T., Blau, R. (1985): *Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems,* Computer Graphics (Proceedings of SIGGRAPH 85). 19(3), pp. 313-322.