Interactive generation of (paleontological) scientific illustrations from 3D-models

Sebastian Schäfer, Christoph Knopp, Detlef Krömker

Goethe-Universität Frankfurt

Figure 1: *A Leptictidium nesutum tooth left upper M^2. Layers: illumination, outline, period stippling, irregular stippling and final image.*

1 Introduction

Scientific illustrations play an important role in the research of natural sciences and are complex drawings that are manually created. The desired images usually combine several drawing techniques such as outlines, distinctive types of stippling or an abstract shaded surface in a single image, as is exemplified in figure 2. Each of these elements aim to focus the viewers attention to certain details like e.g. shape, curvature or surface structure. As to our knowledge no tool exists with which similar images can be produced and the creation of such an illustration is a tedious and time-consuming task, we examined paleontological scientific illustrations together with researchers and artists from *Senckenberg Research Institute and Natural History Museum* with the intention to create an application to render scientific illustrations from a 3D-model.
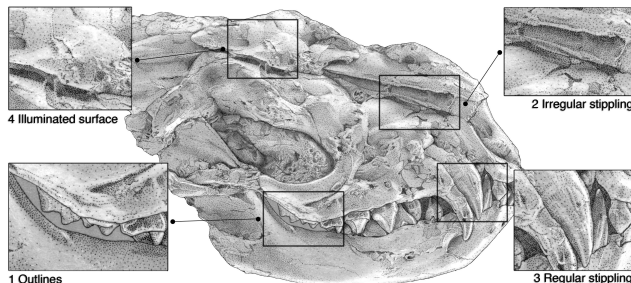
Figure 2: *A Kopidodon head drawn by Juliane Eberhardt.*

As a result we introduce a novel layer-based approach to interactively create scientific illustrations from a 3D-model with well-proven Non-Photorealistic-Rendering algorithms. The whole pipeline is designed to run on the GPU to achieve a real-time rendering by relying on screen-space and image-based methods.

2 Artist Interaction

While it is imperative that the user can freely combine the aforementioned techniques, such a tool must also offer the user other possibilities to interact with the result. It is quite common that the object depicted is an abstraction from the real specimen at hand. The artist should therefor be able to sculpt the base model into the idealized form he imagines.

Another direct interaction is the addition of manually drawn images or the completion and correction of automated results. There are two reasons why this interaction is of high importance. First of all the result of any algorithm might not fully satisfy the taste of the artist and he or she wants to correct e.g. some outlines or stippling points. The other reason is that not all features can be successfully detected by algorithms. This was especially true for curvature detecting algorithms that did not produce satisfying results.

3 Our approach

In the first step the 3D-model is rendered in a single render-pass into G-Buffer textures as these are a good starting point for image space GPU algorithms. Every layer can be freely configured by the user with respect to it's contents, source and how it is rendered into the final image. Once all user chosen layers are rendered they are drawn with alpha blending into a final texture that is then shown with an applied FXAA anti-aliasing filter. The workflow is as follows:

1. Load model, adjust lighting and pick desired view.

2. Setup layers: select characteristic and how to show it.

3. Manually fine-tune the parameters, curvatures or stippling.

4. Store the setup and export a screenshot.

To render each layer we implemented state-of-the-art NPR techniques that can be executed on the GPU. Outlines are created by *Image Enhancement* [Luft et al. 2006] or an edge detection filter applied to normal- and depth-map. Stippling points are set regularly using *Electrostatic Halftoning* [Schmaltz et al. 2010] and irregular using *Recursive Wang tiles for real-time blue noise* [Kopf et al. 2006].

The shaper function is realized by painting on a displacement map. The 2D-painting is implemented by painting onto additional surface textures. The painting mechanism can also be used to explicitly move, add or remove single stippling points.

References

KOPF, J., COHEN-OR, D., DEUSSEN, O., AND LISCHINSKI, D. 2006. Recursive Wang tiles for real-time blue noise. *ACM Transactions on Graphics 25*, 3 (July), 509.

LUFT, T., COLDITZ, C., AND DEUSSEN, O. 2006. Image enhancement by unsharp masking the depth buffer. *ACM Transactions on Graphics 25*, 3, 1206.

SCHMALTZ, C., GWOSDEK, P., AND BRUHN, A. 2010. Electrostatic Halftoning. *Computer Graphics.*