

Computer Tool for Automatically Generated 3D Illustration in Real Time from Archaeological Scanned Pieces

Luis López, Germán Arroyo y Domingo Martín

Departamento de Lenguajes y Sistemas Informáticos
ETS Ingenierías Informática y de Telecomunicación. Universidad de Granada. Granada. España

Resumen

El proceso de documentación gráfica de piezas arqueológicas requiere de la participación de un artista capaz de recrear ilustraciones empleando distintas técnicas expresivas. A menudo, la labor del artista se ve limitada por la inconveniencia de trabajar únicamente con fotografías de las piezas a ilustrar. En este artículo se presenta una herramienta informática que permite generar, de una forma sencilla e intuitiva, ilustraciones en tiempo real a partir de modelos 3D escaneados. La interfaz desarrollada permite al usuario simular elaborados estilos artísticos mediante la composición de esquemas de luces virtuales que el computador procesa para generar ilustraciones desde cualquier posición o ángulo. Entre las luces virtuales implementadas se encuentran técnicas bien conocidas como el dibujado de contornos, el rayado o el sombreado plano.

Palabras Clave: VISUALIZACIÓN EXPRESIVA, DOCUMENTACIÓN ARQUEOLÓGICA, INTERFAZ GRÁFICA

Abstract

The graphical documentation process of archaeological pieces requires the active involvement of a professional artist to recreate beautiful illustrations using a wide variety of expressive techniques. Frequently, the artist's work is limited by the inconvenience of working only with the photographs of the pieces he is going to illustrate. This paper presents a software tool that allows the easy generation of illustrations in real time from 3D scanned models. The developed interface allows the user to simulate very elaborate artistic styles through the creation of diagrams by using the available virtual lights. The software processes the diagrams to render an illustration from any given angle or position. Among the available virtual lights, there are well known techniques as silhouettes enhancement, hatching or toon shading.

Key words: NON PHOTOREALISTIC RENDERING, ARCHAEOLOGICAL DOCUMENTATION, GRAPHIC USER INTERFACE

1. Introduction

The graphical documentation process of archaeological pieces usually presents the inconvenience of working only with the photographs of those pieces. More often than not, the reasons involve lack of access to the pieces because of preservation concerns, incompatibility with the visit schedule, distance between the artist work site and the archaeological site, et cetera. Moreover, if the quality of the photographs is rather poor, the final results can be unacceptable.

The possibility of working with 3D scanned models solves all of these inconveniences and offers an exact representation of the piece from any given angle or position. There is no lack of access to the pieces at any moment, no accuracy problem and opens the door to a wider range of variety in the presentation of the illustrations and the techniques applied to them.

As regards the automatic generation of illustrations from photographs and 3D models, an enormous quantity of very valid techniques has been presented. However, there are few of them that have been developed to offer an easy and intuitive manner of combining the individual techniques in order to enrich the rendered illustrations in a meaningful way.

This paper describes a software tool capable of unify both solutions by using an extended virtual light concept. The application user can render in real time a great variety of illustrations that mimic different artistic styles through the composition of the 3D scanned models with many instances of expressive or transformation virtual lights. Furthermore, the user can access to all the intermediate steps at any moment of the whole process and adjust them if he desires to.

2. Related Works

Martín et al. only use the proposed virtual light model in (MARTÍN, 2001) to extract outlines and shapelines from a 3d model. However, this abstraction can be used in many more areas of the non photorealistic rendering since a great deal of rendering techniques relays, at its most basic level, in the position of the light which is the central element of this model.

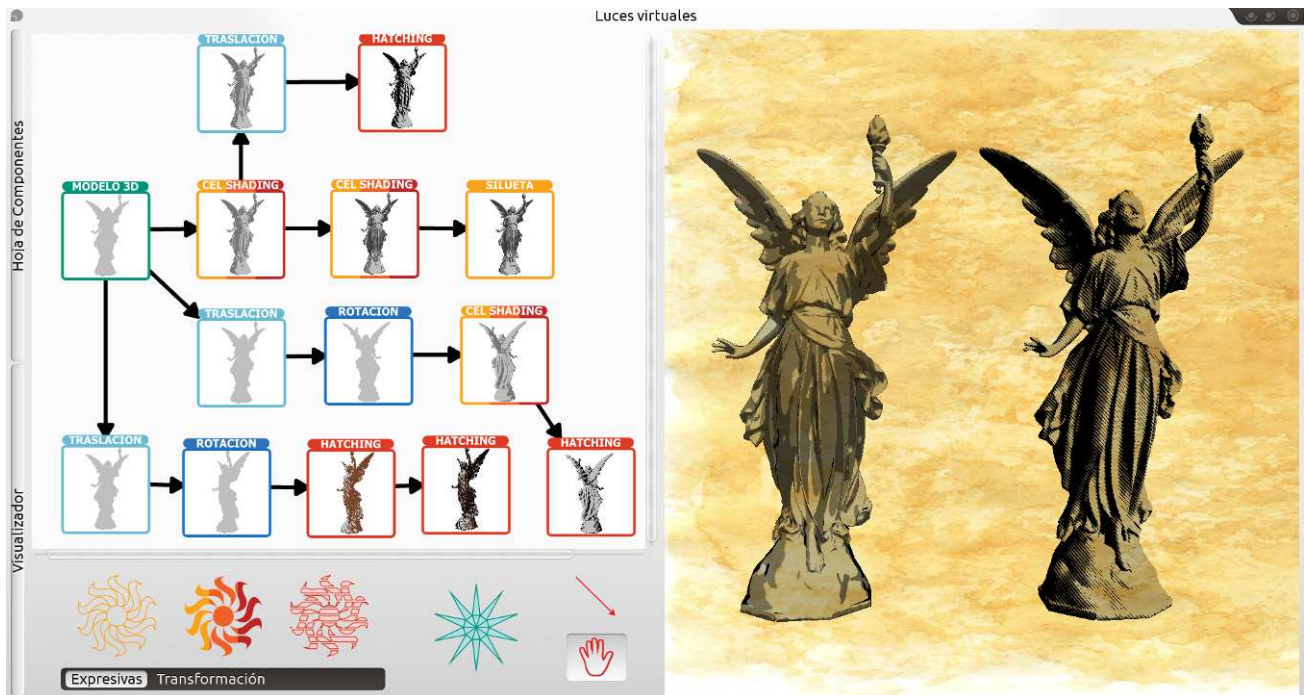


Figure 1. Screenshot of the application with different illustrations of the same 3D scanned model.

Rost et al. (ROST, 2009) offer a shader implementation of the technique proposed by Bert Freudenberg in his thesis Stroke-based Real-Time Halftoning Rendering (FREUDENBERG, 2003). This implementation was shared by Freudenberg to the publication of this book. Although the results are effective and suitable, they are limited by the necessity for this shader to work in the texture coordinate space.

Doss (DOSS, 2008) presents an algorithm to extract and render silhouettes and crease edges using geometry shaders. The main problem with this approach is the lack of guarantees in the continuity of the rendered silhouette. The greater the width of the silhouette is, more visible gaps can be easily spotted. Hermosilla et al. (HERMOSILLA, 2009) offer an acceptable solution to this problem and a broader range of stylization for the rendered silhouettes.

Decaudin (DECAUDIN, 2006) presents one of the first implementations of the toon shading technique. Years later, Lake et al. offer a much more simple solution to this technique known as Hard Shading in (LAKE, 2000).

3. System Overview

The proposed system uses 3D scanned models as input. The models will be rendered without any kind of effects after the software analyze and process them. By making use of the available expressive and transformation virtual lights, the user will be able to create illustrations with intricate artistic styles combining them.

The huge amount of algorithms used by the application can be organized in the following manner:

1. **3D Model Processing Algorithm:** it uses 3D scanned models as input. It analyzes and processes them in order to create the necessary data structures for the next steps.
2. **Diagram Resolution Algorithms:** using the data structures created by the 3D Model Processing Algorithm and the virtual lights diagrams designed by the user, this set of algorithms will take care of the dependencies and they will propagate the necessary information from the root node to the multiple leaf nodes that will be rendered. In view of the interactive nature of the process, these algorithms will react in real time to the changes made by the user: alteration of the 3D scanned model, changes in the virtual lights diagram structures, changes in the parameters of the rendering algorithms, etc.
3. **Rendering Algorithms:** using the updated data structures generated by the Diagram Resolution Algorithms and the parameter values set by the user, this set of algorithms will modify the appearance of the 3D model in order to get the wanted visual style.

4. User Interface

The developed user interface consists of two distinct modules:

1. **Component Sheet:** section of schematic nature where the user can operate with the available virtual lights by using directional arrows in a friendly and intuitive manner.
2. **Visual Display Unit:** the loaded 3D model will be displayed in this section and every single change in the virtual light diagrams built in the Component Sheet will be rendered in real time here. Furthermore, the user will be

able to interact with the position of the selected virtual light in an intuitive manner.

Figure 1 shows the screenshot of a complex pipeline applied to a very detailed 3D scanned model. The functionality of the user interface is explained in the next paragraphs:

- The vertical tabs, Component Sheet and Visual Display Unit, located at the left side of the interface allow to toggle the display state, hidden or visible, of both modules according to the needs of the user.
- The Component Sheet has a display sheet and a series of buttons that the user can use in order to insert, delete, manipulate or link the displayed elements.

There are six virtual lights operators organized in two groups: expressive lights and transformation lights.

Expressive Virtual Lights:

- Silhouette: this virtual light will produce the 3D model silhouette. The look of the silhouette can be altered according to its colour, width and style.
- Cel Shading: this virtual light will produce a toon shading effect on the model surface. The stripe number, colour and width are the parameters which the user can change.
- Hatching: this virtual light will produce a hatching effect on the model surface. The look can be changed according to the hatching direction, the luminosity, the line frequency and their width.

Transformation Virtual Lights:

- Translation: this virtual light will translate the 3D model to the specified coordinates.
- Rotation: this virtual light will rotate the 3D model according to the specified angle and direction.
- Scale: this virtual light will scale the 3D model according to the specified scale factors.

Every single virtual light has a unique configuration window where the user can change its parameters according to his needs.

3. **The Visual Display Unit** shows the results in real time and allows the user to change the position of the virtual lights in a 3D space. In case of some virtual lights, there are additional manipulators in order to further personalize the look of the effects. That is the case of Hatching which has another manipulator to change its direction in an easy way.

The figures 2 and 3 show two 3D scanned models without any virtual light applied. Figure 4 and 5 show those models with a series of virtual lights applied to them. Figure 6 shows the results of the complex pipeline used in Figure 1.

5. Results

The developed application only runs in Linux at the moment but it has been developed with portability to other operative systems in mind. There are no libraries specific to this system and the creation of the executables for MS Windows or Mac OS X should not require any considerable effort. Thanks to the emphasis in accessibility and ease of use, the training period required to familiarize oneself with the application is negligible, hardly a few minutes. Likewise, the ease of use allows users without any artistic training to create illustrations that compete in quality with the works of professional artists. And these illustrations can be made right away, barely a few minutes will be needed in order to create intricate designs with the added advantage that provide the independence of the operations and the reusability of the diagrams between different models.



Figure 2. *Angel Lucy* from Stanford 3D Scanned Repository.

Figure 3. *Ippolita Sforza's* statue sculptured by Francesco Laurana.

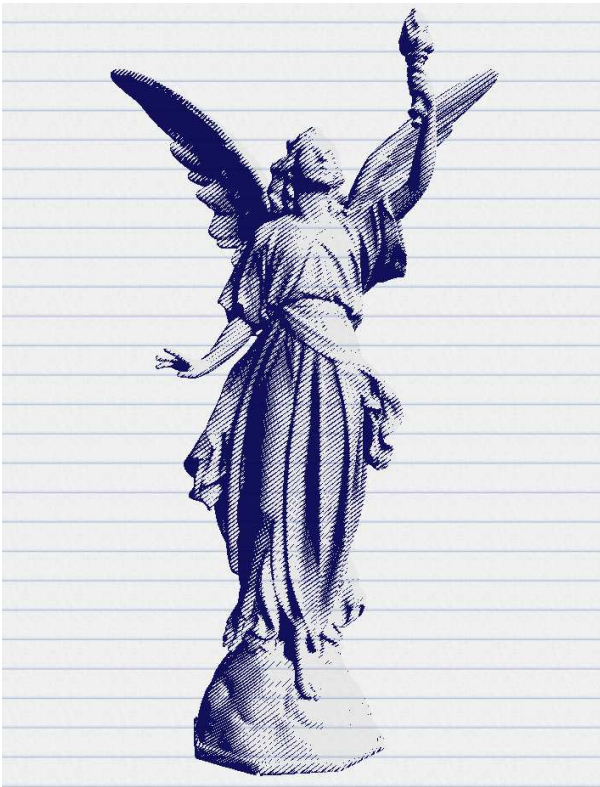


Figure 4. Angel Lucy from Stanford 3D Scanned Repository, with hatching applied.

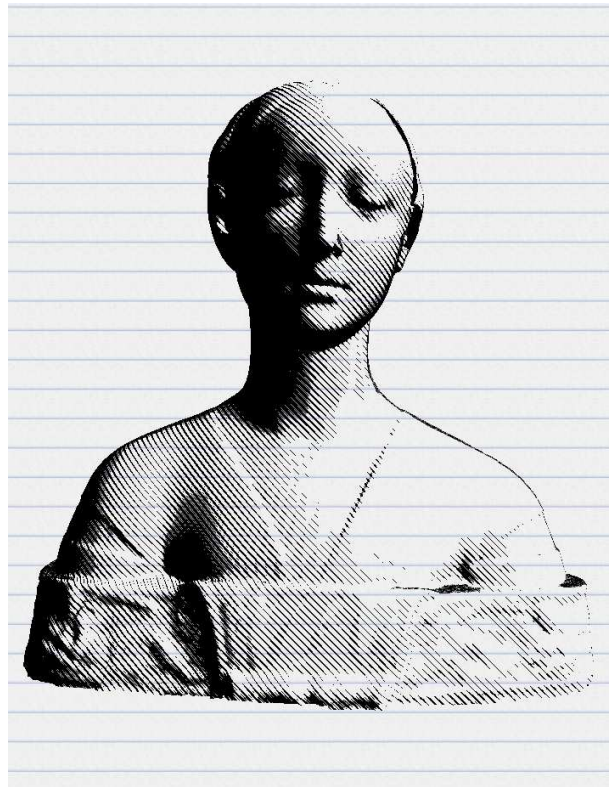


Figure 5. Simple pipeline applied to Ippolita Sforza's statue.



Figure 6. Final results of the complex pipeline shown in Figure 1. Only one model was used to render this illustration.

6. Conclusions

This paper describes a computer tool that allows the easy generation of illustrations with intricate artistic looks from one or multiple 3d scanned models. The user can achieve these results by means of the creation of diagrams that combine well known techniques as hatching, toon shading and silhouette enhancement. The whole process is extremely intuitive, it allows a wide range of characterization through an extensive number of parameters and the user can change every single step whenever he wants in real time.

The variety of styles that can be created with this tool is rather large thanks to the possibilities opened by the mixture of the chosen artistic techniques. Likewise, the created diagrams can be easily reused between different 3D models given the independent nature of the extended concept of virtual lights implemented. Finally, the illustrations rivals in quality to that of professional artists.

Acknowledgments

The writers are grateful to Elodía López for her help. Thanks to the Consejería de Innovación y Empresa of the Junta de Andalucía that has partially funded this article through the project of excellence PE09-TIC-5276.

References

- DECAUDIN, P. (1996): "Rendu de scènes 3D imitant le style <<dessin animé>>", Rapport de Recherche 2919, Institut National de Recherche en Informatique et en Automatique.
- DOSS, J. (2008): "Inking the Cube: Edge Detection with DIRECT 3D 10, [online] <http://software.intel.com/en-us/articles/inking-the-cube-edge-detection-with-direct3d-10/>
- FREUDENBERG, B. (2003): *Stroke-based Real-Time Halftoning Rendering*. Ph. D. thesis, University of Magdeburg.
- HERMOSILLA, P. et al. (2009): "Single Pass GPU Stylized Edges", *IV Iberoamerican Symposium in Computer Graphics – SLACG*, pp. 1-8.
- LAKE, A. et al. (2000): "Stylized Rendering Techniques for Scalable Real-Time 3D Animation", *NPAR2000: First International Symposium on Non-Photorealistic Animation and Rendering*.
- MARTIN, D. et al. (2001): "Rendering Silhouettes with Virtual Lights". *Computer Graphics Forum*, vol. 20, n° 4, pp. 271-282.
- ROST, R. J. et al (2009): "Open GL Shading Language 3rd Edition". Addison-Wesley.