# INTEGRATING REAL-TIME 3D CARTOON RENDERING INTO 2D ANIMATIONS

**Johan Claes**
University of the Balearic Islands – Palma de Mallorca – Spain

**Frank Van Reeth**
Limburg University Centre – EDM – Diepenbeek – Belgium

JClaes@kmm.uib.es FVReeth@luc.ac.be

**ABSTRACT**

In this paper we describe how to achieve nice-looking cartoon style rendering and animation utilising 3D input and animation control. Besides a technical description, we also show how this rendering is smoothly integrated into traditional 2D animations. This way we combine the benefits of both worlds, giving plenty of opportunities to creativity. Maximally exploiting the power of current graphics hardware to obtain real-time rendering speeds is another key issue.

**Keywords:** cartoon animation, non-photorealistic rendering, NPR, cartoon style rendering, real-time animation, computer animation, computer graphics.

## 1 INTRODUCTION

Techniques originally meant for 3D computer graphics are starting to find their way into 2D animation. Cartoon style rendering is typical something that is considered to be purely 2D. We show how 3D scenes effectively can be rendered and animated using adequate algorithms for cartoon style rendering. Furthermore, via maximally exploiting current graphics hardware, we realise real-time execution speeds.

## 2 RELATED WORK

Paul Haeberli [Haebe90] was one of the first pointing out to the computer graphics community that computers can help in much more creative ways than only via photorealistic rendering. Non-photorealistic rendering intends to reach good-looking images and animations without the need to fill in each and every detail. A good overview of how non-photorealistic techniques are evolving in the nineties can be found in [Rober97].

The research about non-photorealistic rendering is diverging in multiple directions. Some people concentrate on merely improving the quality of 2D standalone drawings [Salis96]. Other incorporate real 3D input and render everything in more artistic ways [Winke94]. One step further is animating this 3D non-photorealistic view of the world. [Meier96] describes how 3D painted drawings can be animated.

The specific problems around 3D cartoon rendering are described in the PhD thesis of Philippe Decaudin [Decau96]. We used his research as a starting point and speeded up his approach drastically. Also we succeeded in severely improving the quality of the images.

## 3 CARTOON STYLE RENDERING

In order to get a typical cartoon style look, some specific means of expression are used, like silhouette lines, sharp edges and limiting the number of colours.

### 3.1 Silhouette lines

For a given view direction, silhouette lines are drawn everywhere on the border of the visible part of an object. One way to find these silhouette lines, is analysing the discontinuities in depth information of a rendered image, as described in [Decau96]. We are using a different approach: a topological datastructure helps in quickly locating the joint of frontfaces and backfaces.

### 3.2 Sharp edges

The sharp edges are the discontinuities of the surface normals [Decau96]. Analysing the input and maintaining our special datastructure makes sure we find these edges quickly.

### 3.3 Limiting the number of colours

The most typical feature in a cartoon style drawing, is the limited use of colours. In our cartoon renderer, the user chooses two or three colours for a given surface to be rendered. For every vertex of every visible face, a colour value is calculated. In the final rendering step, the colour for every face is determined. The simplest approach is giving the cartoon-colour that most closely matches the mean colour of the face to the entire face. But better results are obtained by subdividing the faces. Via an optimal subdivision technique, good results can be obtained without using too much polygons.

### 3.4 Composing everything together

Together with the faces, also the newly added lines are treated as having depth information. That way OpenGL can take care of properly hiding the invisible parts of lines and faces.

## 4 AN EXAMPLE

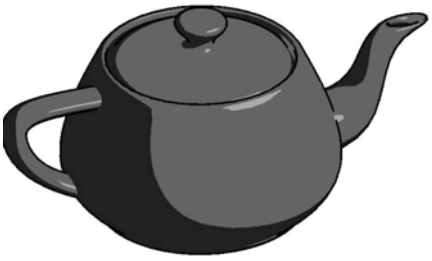

Fig. 1 A rendering of a famous teapot using three cartoon-colours.

The 7744 faces of fig. 1 are rendered at a speed of 15 images per second on a PentiumII-400Mhz using a standard OpenGL supporting graphical board. We used a resolution of 400x400 pixels.

## 5 FUTURE RESEARCH

In the future we want to further extend the possibilities of our cartoon style renderer. Cartoon style texture-mapping is one of the topics. Integrating some of the research about stylised procedural animation, carried out at the Glasgow University by Yu and Patterson [Yu96] is another.

## 6 ACKNOWLEGDEMENTS

We would like to express our thanks to the many people providing ideas and helping with the implementation. Especially Bart Claes, Gert Vansichem and Marc Flerackers for the implementation work and Liesbeth Beckers for the graphical design. We also want to express our gratitude to the fact being part of the European PAVR project, which is partly funding this research.

## 7 REFERENCES

[Decau96] Ph. Decaudin, *Modélisation par Fusion de Formes 3D pour la Synthèse d'Images -- Rendu de Scènes 3D imitant le Style «Dessin Animé»*. Ph.D. Thesis at the Université de Technologie de Compiegne (France), December 1996.

[Haebe90] P.Haeberli, *Paint By Numbers: Abstract Image Representations*, Computer Graphics (Siggraph), vol. 24 - 4, p. 207-214, Augustus 1990

[Meier96] B. Meier, *Painterly Rendering for Animation*, Computer Graphics (Siggraph), vol.30, p.477-484, 1996

[Rober97] B. Robertson, *Different Strokes*, Computer Graphics World, December 1997

[Salis96] M. Salisbury, C. Anderson, D. Lischinski, D. Salesin, *Scale-Dependent Reproduction of Pen-and-Ink-Illustration*. Computer Graphics (Siggraph) Vol.30, p. 461-468, 1996

[Winke94] G. Winkenbach, D. Salesin, *Computer Generated Pen-and-Ink-Illustration*, Computer Graphics, p. 91-100, ACM SIGGRAPH, 1994

[Yu96] J. Yu, J. Patterson, *A Fire Model For 2D Computer Animation*, Computer Animation and Simulation'96, New-York, 199