Enhanced Cartoon and Comic Rendering

Martin Spindler † and Niklas Röber and Robert Döhring ‡ and Maic Masuch

Department of Simulation and Graphics, Otto-von-Guericke University of Magdeburg, Germany

Abstract

In this work we present an extension to common cel shading techniques, and describe four new cartoon-like rendering styles applicable for real-time implementations: stylistic shadows, double contour lines, soft cel shading, and pseudo edges. Our work was mainly motivated by the rich set of stylistic elements and expressional possibilities within the medium comic. In particular, we were inspired by Miller's "Sin City" and McFarlane's "Spawn". We designed algorithms for these styles, developed a real-time implementation and integrated it into a regular 3D game engine.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Three-Dimensional Graphics and Realism]: Color, shading, shadowing, and texture

1. Introduction

For some years, the field of non-photorealistic rendering (NPR) has also focused on the representation of comic-like computer graphics, being limited to typical characteristics of cel shading. The gap in the expressiveness of traditional comics over their adoption in computer applications still remains, because only little research has been done in reflecting, which artistic style is appropriate for a certain purpose, being with [Hal03] the only exception. Additionally, more enhanced comic rendering techniques, such as double contour lines, stylistic shadows or a soft cel shading, hardly exist. In this paper, we will focus on presenting our results regarding the second aspect.

1.1. Non-Photorealistic Rendering techniques

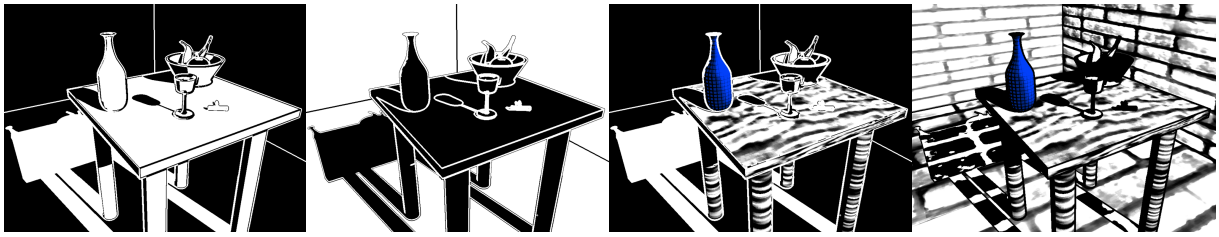
The field of NPR aims at simulating handmade illustrations with computer algorithms. It is mainly motivated by traditional illustration techniques used in the area of medicine, archeology, and architecture. In the first instance this includes feature line algorithms [ST90, Her99, BS00, IFH*03] and half-toning techniques, such as stippling [DHvOe00], hatching [PHWF01, FMS02], and engraving [Ost99]. In many cases, these techniques are better in emphasizing the essential aspects of a concept, when compared to a photoreatlistic rendition.

Another traditional technique, known as cartoon- or cel shading, has also found its way into the field of computer graphics [Dec96, LMHB00]. Since its origin is in comic books and cartoon movies, this technique is often used in entertaining environments, such as computer games [Ubi]. A typical cartoon-like depiction is characterized by silhouette and object edges and discrete color tones. Unfortunately, the majority of comic rendering systems confine themselves to these two attributes only, and ignore the interplay with other NPR techniques.

1.2. Traditional Comics

The modern comic, as we understand it today, evolved from comic strips in newspapers and magazines at the end of the 19th century. In its further development, regional variations played an important role and led to a creation of independent styles in the USA (western comic, e.g. *Superman*), Europe (frankobelgian comic, e.g. *Asterix et Obelix*) and Japan (manga comics, e.g. *Akira*). However, some elements are common within all regional styles, making it possible to recognize a comic as such. These elements include certain camera techniques and perspectives, special effects such as speedlines, temporal and spatial arrangements in panels, the

† {spindler,niklas,masuch}@isg.cs.uni-magdeburg.de
‡ robb.dee@web.de



(a) Inverse silhouette style (b) Silhouette style (c) Foreground shaded only (d) Shaded + inverse shadows

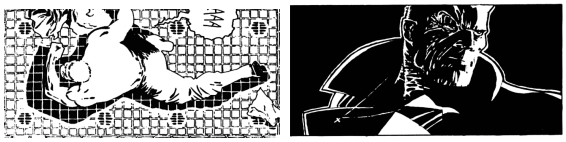
Figure 1: *Comic rendering style inspired by Frank Miller's "Sin City" [Mil93] rendered with our system.*

combination of pictures and text, but also level of detail, abstraction and a from real-world physics deviating light- and shadow interaction.

During our research in local comic stores, we realized that modern comic publications arbitrarily mix different styles and techniques, leading to an increased feasibility of expression. The use of different styles and elements is aimed very precisely and is not only chosen by artistic considerations, but is also geared to the visual support of the plot. This is especially true for Frank Miller's "Sin City" [Mil93] and Todd McFarlane's "Spawn" [McF97], which we both analyzed in detail and tried to capture their essence (see also Sections 2 and 3).

2. Frank Miller's "Sin City" style

Frank Miller's "Sin City" [Mil93] comics are characterized by their own very unique "pen and ink" look, but Miller does not commit himself to only one style, but rather chooses an unconventional combination of hatching, stippling, large black faces, and monochromical silhouettes seeking the contrast to the background. Miller's style is particularly qualified in communicating a dusky and bleak mood. Furthermore, it allows the author to apply additional colored elements to highlight certain objects. Our analysis of the "Sin City" style regarding a game engine implementation resulted in four different plot-dependent cases: (a) *silhouette style*: background (bg) white + foreground (fg) black, (b) *inverse silhoutte style*: bg black + fg white, (c) *Fg shaded only*: bg black, and (d) *Fg + bg shaded* (see figure 1). Additionally, we found two very characteristic style elements: *stylistic shadows* and *double contour lines*.



(a) Stylistic shadow (b) Double contour lines

Figure 2: *Extracts of Frank Miller's "Sin City" comics.*

2.1. Stylistic shadows

There are two different variations of representing shadow in "Sin City". The first one is the traditional shadowing technique, while the second one - *stylistic shadows* - is more complex: if the shadow falls upon a foreground object, it is drawn in black (even if this is the shadow throwing object). However, if the shadow falls upon the background, the color is inverted, ensuring that the background structure is still noticeable (see figure 2(a)). For an adoption of *stylistic shadowing* to a game engine we need to be able to produce real-time shadows as well as to distinguish between foreground and background objects. This is possible by extending the stenciled shadow volume approach to incorporate an additional object property, which is stored in a "Foreground"-buffer.

2.2. Double contour lines

The drawing of contour lines in foreground objects differs in "Sin City" to silhoutte edges in common cel shading: They are comparitively thick and often drawn doubly, as can be seen in figure 2(b). In order to integrate these *double contour lines* into a game engine we handle them as thick white lines surrounded by a very thin black outline. This simplification allows us to utilize common morphologic operations (dilation filters), which are used to determine and broaden edges. Since *double contour lines* only exist for foreground objects, we additionally make use of the "Foreground"-buffer and apply another dilation filter to draw a black thin outline arround the appropriate edges.

2.3. Rendering Pipeline

The complete rendering pipeline for the "Sin City" style requires three passes. In the first pass, a projection of the scene is computed including information about colors, normals, depths and objects ids. Besides the determination of shadow regions, this pass is also responsible for distinguishing between foreground and background objects. The second pass is used for the image-based computation of the edges by utilizing the information about colors, depths, normals and object ids obtained in the first pass. In the third pass, shadow

areas of the background are identified and inverted to achieve *stylistic shadows* (see section 2.1). Subsequently, edges are broadened, possible gaps are filled, and *double contour lines* are generated (see section 2.2). At last, the colors are combined with the edge information to obtain the final image (see figure 3).

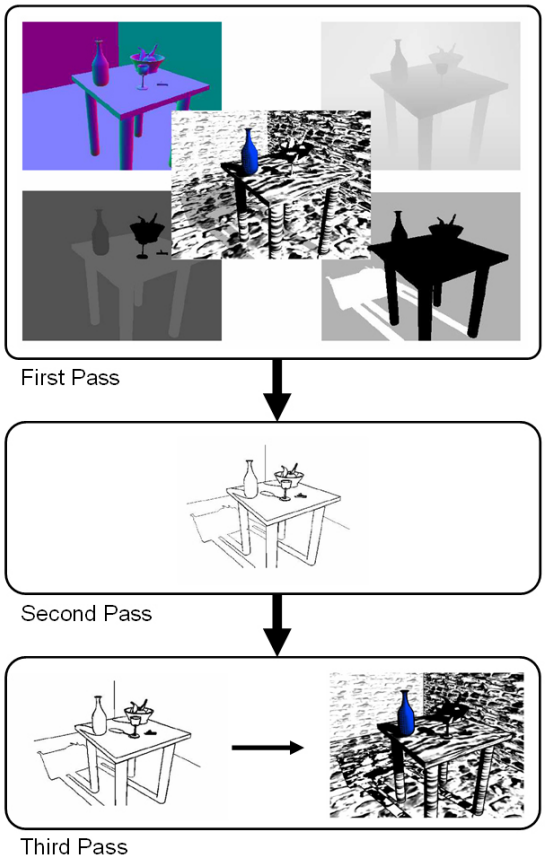
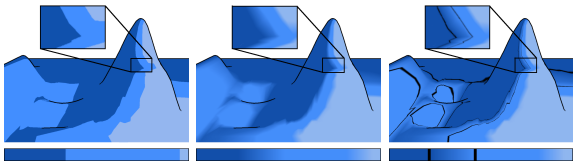


Figure 3: *Rendering pipeline for the "Sin City" style.*

3. Todd McFarlane's "Spawn" style

Todd McFarlane's "Spawn" [McF97] is a typical representative for another cartoon style characterized by a variety of colors (even with color gradients) and many feature edges accentuating small details such as cloth creases. Although this kind of style provides a realistic look, a cartoonish appearance is still retained.

Focussing on a real-time implementation we extended the approach of Lake et al. [LMHB00], which uses a 1D texture lookup as demonstrated in figure 4(a). Here, no color gradients are used and therefore only a small texture resolution is necessary. Since the "Spawn" style uses single-colored areas combined with color gradients, we increased the resolution of the 1D cel texture allowing us to easily define both, single-colored areas and color gradient areas. This so-called *soft cel shading* technique is shown in figure 4(b). A very useful aspect of this solution is the fact, that a designer can freely define, whether a region should be single colored or not.



(a) Cel shading (b) Soft cel shading (c) Pseudo edges

Figure 4: *Enhanced cel shading.*

To add the missing feature lines, we exploit the fact, that they often occur between single-colored areas and color gradient areas. Therefore, we further extend the 1D cel texture by simply setting small texture regions to the edge color (black), as shown in figure 4(c). Thereby, we achieve what we call *pseudo edges*. We also use this technique for simulating cross-hatching (see figure 5). A problem when blending between different level of detail, which occurs is demonstrated in figure 5(b).

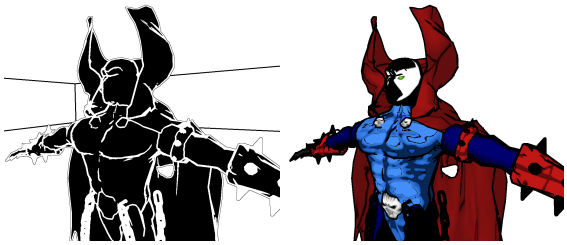
(a) Detail Level 0 (b) Blended levels (c) Detail Level 1

Figure 5: *Hatching by exploiting pseudo edges. Blending the detail levels* (a) *and* (c) *can result in hatching artefacts as shown in* (b).

4. Implementation and Results

The concepts introduced in section 2 and 3 have been implemented within the Fly3D game engine [Pol06] using Cg shaders. Depending on the scene complexity ($10,000 - 50,000$ triangles) we achieve frame rates between $7\,fps$ and $15\,fps$ at a 1024×768 resolution. The results have been measured on an Athlon XP 2400+ system with 1.25 GB RAM and an NVIDIA GeForce 6600 GT graphics processor. With shadows turned off, the frame rate increases to $30\,fps$ and $45\,fps$. This disproportional large performance gap is caused by the volume-shadow technique used for dynamic shadowing in the Fly3D engine, which is not very practical for large screen resolutions.

Figure 1 and 6 show some of our results achieved. In figure 1 the four plot-dependent cases from "Sin City" are displayed, note the *stylistic shadow* in figure 1(d) and the *double contour lines* in figure 1(b) and 6(a). Figure 6(b) displays



(a) "Sin City" style (b) "Spawn" style

Figure 6: *"Spawn" rendered with our system in different styles.*

the use of the "Spawn" style including *soft cel shading* and *pseudo edges*.

5. Conclusions and Future Work

We compared traditional comics and computer games and can conclude, that (1) traditional comics provide a much richer set of elements compared to a modern computer game and (2) the combination of different styles play an essential role in many traditional comic publications, but not in computer games. With regard to further development, not all traditional comic elements are adaptable as animated sequences. Problems are frame-to-frame coherence and the restrictions in low pixel-resolution (stippling). Moreover, in traditional media a comic artist creates every single picture only considering his own artistic requirements, whereas the beholder is forced to remain passive. In contrast, interactive computer graphics produce pictures dynamically, i.e. freedom of decision is taken away from the artist partially. This raises the question, how computer algorithms can counterbalance this gap, e.g. by finding decision criteria for the rendering style to be used. Therefore, a detailed evalutation of all rendering styles should be aspired. Hopefully, our work encourages more research into the fusion of traditional comics and computer games.

References

[BS00] BUCHANAN J. W., SOUSA M. C.: The edge buffer: a data structure for easy silhouette rendering. In *NPAR '00: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering* (New York, NY, USA, 2000), ACM Press, pp. 39–42.

[Dec96] DECAUDIN P.: *Cartoon-Looking Rendering of 3D-Scenes*. Research Report 2919, INRIA, 1996.

[DHvOe00] DEUSSEN O., HILLER S., VAN OVERVELD C., E T. S.: Floating points: A method for computing stipple drawings. *Computer Graphics Forum 19*, 3 (2000).

[FMS02] FREUDENBERG B., MASUCH M., STROTHOTTE T.: Real-time halftoning: a primitive for non-photorealistic shading. In *Proceedings of the 13th Eurographics Workshop on Rendering* (2002), Eurographics Association, pp. 227–232.

[Hal03] HALPER N.: *Supportive Presentation for Computer Games*. PhD thesis, Otto-von-Guericke-University Magdeburg, Germany, 2003.

[Her99] HERTZMANN A.: Introduction to 3D Non-Photorealistic Rendering: Silhouettes and Outlines. In *ACM SIGGRAPH 99 Course Notes. Course on Non-Photorelistic Rendering*, Green S., (Ed.). ACM Press, New York, 1999, ch. 7.

[IFH*03] ISENBERG T., FREUDENBERG B., HALPER N., SCHLECHTWEG S., STROTHOTTE T.: A Developer's Guide to Silhouette Algorithms for Polygonal Models. *IEEE Computer Graphics and Applications 23*, 4 (July/Aug. 2003), 28–37.

[LMHB00] LAKE A., MARSHALL C., HARRIS M., BLACKSTEIN M.: Stylized rendering techniques for scalable real-time 3d animation. In *Proceedings of the first International Symposium on Non-Photorealistic Animation and Rendering* (2000), ACM Press, pp. 13–20.

[McF97] MCFARLANE T.: *Spawn*, vol. 21-59. Image Comics, 1994-1997.

[Mil93] MILLER F.: *Sin City: The Hard Goodbye*. Dark Horse Comics, 1993.

[Ost99] OSTROMOUKHOV V.: Digital Facial Engraving. In *SIGGRAPH '99: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1999), ACM Press/Addison-Wesley Publishing Co., pp. 417–424.

[PHWF01] PRAUN E., HOPPE H., WEBB M., FINKELSTEIN A.: Real Time Hatching. In *ACM SIGGRAPH 2001* (2001), ACM Press.

[Pol06] POLICARPO F.: *Fly3D Game Engine*, 2006. http://fabio.policarpo.nom.br/fly3d.

[ST90] SAITO T., TAKAHASHI T.: Comprehensible rendering of 3-d shapes. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques* (1990), ACM Press, pp. 197–206.

[Ubi] UBISOFT: *XIII*. http://www.xiii-thegame.com.