Game Graphics Beyond Realism: Then, Now, and Tomorrow

Maic Masuch and Niklas Röber

Otto-von-Guericke University Magdeburg
Games and Graphics Research Group
Institute for Simulation and Computer Graphics

{masuch | nroeber}@isg.cs.uni-magdeburg.de
http://isgwww.cs.uni-magdeburg.de/games/

ABSTRACT
Photorealism is one of the most quoted aspects of nowadays games. However, realistic game graphics is not the only style desirable. This paper surveys the many graphical styles used in past and current games, analyzes graphical aspects of computer games and discuss the use of realism with respect to game graphics. We show several examples and make references to current research, encouraging game developers to experiment with alternative, more artistic rendering styles, such as non-photorealistic rendering.

Keywords
Game graphics, gameplay, non-photorealistic rendering, cel-shading, realism in games

INTRODUCTION

What is it that makes a computer game fun and interesting to play? This is probably one of the most difficult questions to answer in the field of game development. From the gamer's point of view, great graphics do not necessarily come along with great gaming experience. The time when astonishing graphics alone could sell a game is probably over. Due to constantly evolving graphics hardware, new visual effects are possible which more and more increase visual realism and immersion. Most games try to simulate the visual impression of reality as closely as possible. But this realism stretches only to a certain degree. In games we want to be heroes with superior abilities and hence, we are able to bend the laws of physics to fit the needs of the story. Why not also break the laws of graphical realism in order to enhance the visual experience?

This paper encourages the use of alternative, more artistic rendering styles, such as non-photorealistic rendering. Some games already make use of a cartoon style rendering, but so far no other non-photorealistic rendering techniques have been adopted by the community. These methods could be used for artistic purposes to capture the style of different media (e.g. comics), and to assist in storytelling by changing the style used throughout the game (e.g. to evoke emotions, or establish certain moods).

The paper is outlined as follows: First it analyses the role of graphics in computer games, and presents an overview of different graphical elements used in computer games. Modern games have departed from their primitive graphical roots and much too often, players notice that the game developer had put much more emphasis on neat rendering techniques than on innovative game ideas. This will be investigated by an examination of graphical elements.

Strictly speaking, no game developed so far can truly be called "photorealistic", but many games try to achieve the highest "photorealism" possible. We discuss the use of graphical realism introducing the use of non-photorealistic rendering and its existing techniques applied to games. The paper concludes with a discussion of possible graphical styles in future games.

GRAPHICS IN COMPUTER GAMES

Asking gamers about the most important features of a computer game, the majority will state one answer: Fun. Fun, resulting from a good gameplay and an intoxicating story are vital ingredients, whereas game graphics seem to play only a minor role. Marketing departments of game publishers however, are convinced that screenshots of great graphics on the cover of a game is most important. Whether this is true or not, most game developers and publishers go along with the 3D hype and develop games with eye-catching 3D graphics. Moreover, this process is accelerated by manufacturers of graphics hardware, who push the limits of scene complexity further, while the graphic artists are edaciously waiting to create even more elaborate effects in order to increase the realism of the rendering.

Quite a lot of today's gamers can exactly tell the differences between the most current game engines used in their favorite games. These topics are also discussed in detail in game magazines and throughout the game community and every time where more insights on the next generation game engine are revealed to the public it is received with great acclamation. As a matter of fact, however, mediocre graphics does not ruin a great game, whereas on the other hand outstanding graphics typically does not make a bad game any better.

Gameplay and Graphics

Gameplay is one of the most often used words in the game development community, yet its meaning is not always clear. Generally it is assumed that gameplay is "everything except graphics". Or in other words: gameplay is what happens and how it is achieved in a game from the viewpoint of the player. This can vary from attacking the enemy in first-person shooters to giving orders for the construction of new buildings in strategy games. So, gameplay is a well balanced composition of interaction, navigation, communication, and presentation that allows the players to do what they want to do. A good gameplay grants the player an appropriate interaction within the game world but also sets certain boundaries without obstructing or confining desired actions. Hence it is necessary for the user interface of the game to communicate the games status by various means, e.g. hit points or highlighting objects that react to an interaction. This task is achieved by a number of graphical elements.

Graphical elements of Computer Games

From all our senses, vision is the most dominating one. It is estimated that it contributes up to 70% of the information humans perceive. To process this information, we rely on diverse visual cues which all are essential for our orientation and perception. There are several different graphical elements that have to be paid attention to while developing a game. Some of them we are considering here are:

- Dimensionality,
- Perspective,
- Color,
- Presentation, and
- Realism.

The **dimensionality** of game graphics (not counting text here) can vary between 2D, 2_D and 3D graphics. 2D graphics is used on most board based game implementations where a top view of the board displays sufficient information of the game, Figure 1(a). Later 2_ D graphics, also called pseudo 3D, has been adopted from animated movies in order to give the illusion of 3D, although the underlying technique was entirely 2-dimensional.

Figure 1: (a) *Space Invaders* (2D) [1], (b) *Broken Sword* (2_D) [2], (c) *Quake* (3D) [3].

Popular examples are *Spindizzy* [4] or *Knight Lore* [5], and to animated excellence *Broken Sword* [2], which uses an animated character in front of a fore-, middle-, and background-layer just like in traditional cartoon animation, Figure 1(b). Due to advances in graphics hardware, most games use 3D game engines that display a 3-dimensional world using a perspective projection with proper optical qualities. *Quake* [3] was such a milestone displaying a fully textured 3d world, Figure 1(c). Since then, 3D game engines have been applied to nearly every game genre with great success.

Figure 2: (a) isometric projection, (b) central perspective with one, (c) perspective with three vanishing points

When talking about dimensionality, one also has to pay attention to the **perspective** of the displayed scene. First- or third person games developed using a 3d game engine usually use the accustomed perspective camera, while some genres, like strategy or role playing games use an isometric projection model. Figure 2 shows some examples.

In some games, especially when a cartoon style is aspired, the camera is distorted and the perspective is exaggerated, as it is known from some comics. Old classic *Day of the Tentacle* [6], Figure 3(a) had this carefully handcrafted, *Stupid Invaders* [7] 3(b) and *Escape from Monkey Island* [8] 3(c) already used 3D models to incorporate the toon-style in the geometric models.

Figure 3: Cartoon-Style – (a) *Day of the Tentacle* [6], (b) *Stupid Invaders* [7], (c) *Escape from Monkey Island* [8]

Color also plays an important role in establishing an atmosphere. It can convey certain moods, as shown in Figure 4. Like illustrated in Figure 4 with screenshots from *Thief – The Dark Project* [9], the world can change from a bright and safe place to a dark and gloomy cave. Further, the sudden change of color can introduce special situations, like switching to b/w for flashbacks.

Figure 4: Atmospheric use of color in the game *Thief-The Dark Project* [9].

The **presentation** defines how the game world and the player himself manifest on the screen. This can vary from pure text to first or third person perspectives, or even being just the top down view on the games world. It also describes the integration of the user interface and the factor of immersion. Some examples can be seen in Figure 5, where Figure 5(a) shows a textual scene representation from *Zork* [10]. Later sprites where used to depict the character as in *Donkey Kong* [11] and today most games use modeled 3d characters like in *Tomb Raider* [12].

The **realism** is defined by whether the game is (photo) realistic in look and feel, or exaggerate as an example the use of a comic shader and uses a non-realistic game environment. There are a number of aspects that contribute to the perception of realism like realistic sound, realistic character animation or

the believable behaviour of objects and characters (which is controlled by the physics-engine or the AI-engine). On a higher level of abstraction the setting and causality of events or storyline contribute to the users perception of realism[1]. Many games also incorporate a variable flow of time that allows things to speed up or to slow down. Prominent examples are the bullet time mode in Max Payne or the accelerated time scale in

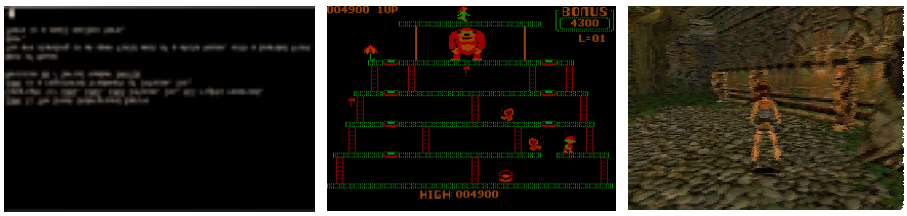

Figure 5: (a) Text: *Zork* [10], (b) Sprite: *Donkey Kong* [11], (c) 3D model: *Tomb Raider* [12]

simulations. But the most crucial reference to realism is probably referring to the depiction of game elements, the game graphics (as it is controlled by the render engine). Since the very beginning of game graphics each era produced its visual masterpieces, let it be *Battlezone* [13], *Myst* [14] or *Half-Life2* [15], compare Figure 6. The realism in computer graphics will be discussed in further detail in the next section.

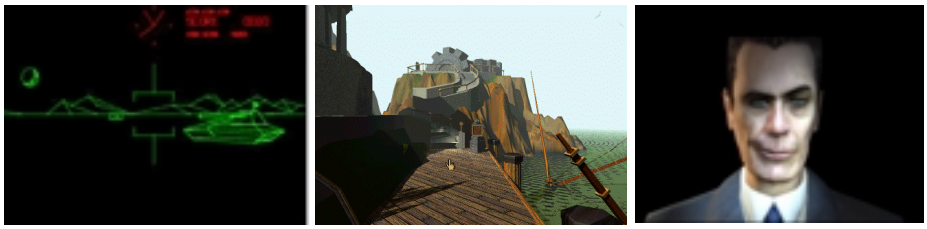

Figure 6: Different milestones of r*ealism* in Computer Games, (a) *Battlezone* [13], (b) *Myst* [14], (c) *Half-Life2* [15]

ALTERNATIVES - NON-PHOTOREALISTIC RENDERING FOR GAMES

Since the beginning, computer games have always been one of the most demanding applications and driving force for computer graphics hardware. This evolved from simple 2D pixel sprites in early games back in the 80s to nowadays 3-dimensional photorealistic environments with a believable atmosphere and correct physical simulations. While most researchers in computer graphics have focused on re-creating the physical world, some have concentrated on non-photorealistic rendering, mostly inspired by human drawing techniques. With the recently developed graphics accelerators, many of these techniques are now ready to be implemented in real time and can be integrated in computer games.

---

[1] We completely disregard the notion of realistic interaction as adequate force-feedback experiences are not in sight even on the long term.

This chapter describes the benefits, as well as the drawbacks, of these NPR techniques for computer games and briefly discusses a few examples algorithms. When we are investigating realism in game graphics, we first have to examine our understanding of the term “realism”.

Realism

Despite all the recent improvements, strictly speaking, no game graphic today can truly be called “photorealistic”. The term “photorealism” in computer graphics originated in the 80s, when the first single images rendered for hours and sometimes even days. Nowadays, modern rendering algorithms can generate images that are indistinguishable from photos, but still, they need quite some time for the rendering. Games however, have to render graphics in real time to provide interactivity. Surely, in a few years from now on, game graphics will resemble the aesthetic quality of today’s rendered movies. This is just a matter of processing power, but we still see the “engine race”. A substantial amount of developing efforts of most game companies goes into the ever advancing ambition trying each new title to look better than the ones from the competitors, adding more and more new visual effects. Recently, Rubin had made a very good point in explaining that although graphic improvement will continue, for the majority of titles it will no longer serve as the driving force of success [16].

Motivation for Non-Photorealistic Graphics in Games

Non-photorealistic rendering techniques (NPR) are beneficial in many ways: They can support in storytelling, be expressive, giving the game a certain artistic look and feel. The most important factors of great games are gameplay and the factor of immersion. The quality of immersion directly depends on consistency in graphical presentations and behavior of game world objects. According to Disney, realism in depiction is not the real question; it is believability [17].

It is surprising that NPR – though relatively new – has not yet found its way out of academia. A number of NPR research articles state that non-photorealistic renditions appear more natural and are perceived far more easily than complicated technical drawings or photorealistic renditions. Further, NPR can be used to explicitly highlight and emphasize certain parts of an image without disturbing the image or breaking the atmosphere [18].

This is a huge advantage and has barely been utilized in computer games. One fine example is the announced and highly anticipated game *XIII* from Ubisoft [19], see Figure 7. In this game, the overall continuity of a comic-look is even enhanced by using additional features of comic-elements like insets or onomatopoeia. These features support the storytelling and give extra information to the player. Additionally, as the game *XIII* is derived from a comic, the cartoon rendering of all game elements reflect the original media resulting in a unique and dense atmosphere.

Even with the highest advanced graphics hardware available today, all so-called photorealistic games simply lack true photorealism. The improvements in image quality compared to games some 10 years ago are simply amazing. Nevertheless, there are a lot of (sometimes only minor) details of incorrect realism. Immersion is like a soap bubble: Even little discrepancies like an incorrect shadow will break the illusion of photorealism and thus the

immersion and the bubble are gone. This can be a very disappointing experience for both, game developers and players alike.

Surprisingly, if non-photorealism is used for the game graphics, this is true for a far lesser extend. Here, inconsistencies seem far more tolerable as non-photorealism comes along with deviations and sketchiness per default. Hence, the design and the feel of the game appear to be smoother.

Figure 7: With stylish cartoon rendering and the adoption of comic elements such as insets or onomatopoeias, Ubisofts *XIII* [19] turns to be the graphical most innovative game for years.

A very important feature of animated non-photorealistic renditions is the demand of frame coherency, which means that lines depicting objects should preserve a certain form over time, otherwise – if they are randomly changing from frame to frame – the overall impression of the animation can be disorienting and disturbing.

Current and Future NPR Styles for Games

The only best known and commonly used NPR technique so far is cel- or cartoon based shading. Other techniques like pen-and-ink drawings are not yet used in commercial games. These methods can be employed as an innovative rendering style, not displacing photorealistic rendering but serving as an alternative. Although some of the NPR techniques take advantage of new hardware features such as vertex- or pixel shader, many can be implemented efficiently, running in real-time on moderate systems.

Cel-Shading

Cel-Shading techniques are used from the beginning of computer games for the graphical presentation. Early graphics hardware was far away from rendering million polygons per second and the complete rendering was done in software. Cartoon rendering is the simplest NPR technique to implement and widely familiar from comics, cartoons and other media.

Even though these methods are not broadly used in games yet, there are a few examples to mention. One popular example, which uses this cartoon style rendering, is: *Cel Damage* [20]. Some of the older games were developed this way because at this time photorealism simply was not achievable; some of the later games used this style to explicitly differentiate from the masses or to simulate a certain media, *XIII* [19]. Other games make use of a pop art like

cartoon rendering as can be seen in *PaRappa the Rapper* [21] or *Jet Set Radio Future* [22].

Cartoon shading can be accomplished in a variety of styles depending on the targeted graphics. The character or object is often shaded using three basic colors, one for the specular, one for the ambient and a third one for the shadow term. Details are highlighted trough a silhouette and characterizing edges, like ridges and valleys on the polygonal model. Figure 8 shows an example.

Figure 8: Example for Cel-Shading from the title *Cel Damage* [20].

The implementation of cel-shading is straightforward and explained in great detail in several research articles [23] [24].

Pen- and Ink Drawing

One characteristic of pen-and-ink style drawings is the absence of colors other than black and white, and that shading is achieved through hatching instead of differently colored areas. One of the earliest examples of real-time computer generated pen-and-ink drawings were presented by Praun et al. [25]. An implementation suitable for games is given in [26] and [27]. Figure 9 displays a screenshot illustrating this technique implemented in a real-time environment. The algorithm uses standard polygonal models. The shading is evaluated by the lighting equation and few hatching texture maps which are blended onto the objects for shading. The number of hatching lines varies depending on how the object is lit.

Figure 9: Example for a pen-and-ink style game engine.

More experimental approaches came from Mohr et. al. who 'hijacked' the OpenGL library to render applications using OpenGL in a different way. Their method can be applied to any OpenGL based application as well as games. They used Quake as an example, rendering it in a non-photorealistic style [28]. *Pencil Whipped* [29] is an innovative 3D shooter that uses sketchy, hand drawn textures and is really fun to watch.

Oil Painting

Up to now, oil painting and related styles are not used in games. The characteristics of oil paintings vary widely and strongly depend on the artist who drew them. They can look photorealistic, cubistic, pointillistic or whatever painting methods were used. The technique to simulate them using computer graphics is a little more complicated than the other methods discussed so far, as paintings are drawn with thousands of single brush strokes, often layered on top of each other. In order to achieve such a look, this process has to be simulated, as well as the media they are drawn on.

Figure 10: Example for oil painting.

In order to maintain frame coherency in the animated images, particles are used in a pre-processing step that are evenly distributed over the polygonal surface to later serve as seed points for the brush strokes. The strokes can vary in color, length, curvature and style.

An example of a landscape painting can be seen in Figure 10. The first animated example of frame coherent oil painting was presented by Meier [30]. Later, several approaches used a similar principle for real-time implementations, some of them hardware accelerated [31] [32].

Pencil or Colored Crayon Drawing

Pencil or charcoal drawings are alike to pen-and-ink styles. In contrast to pen-an-ink, pencil drawings are often gray, as from a pencil or charcoal, but can also be colored as when using crayons [31] [32]. The objects are drawn and characterized by

silhouettes and discriminating edges, and shaded by hatching, smudging, halftoning or stippling methods. An example of a simple scene is displayed in Figure 11.

Figure 11: Example for a colored crayon drawing.

Realism and Beyond – The future of Game Graphics

We examined realism in computer games and showed that realism is already extended in various aspects. We argue that with respect to the graphical presentation of a game, photorealism is not always the most desirable aim, moreover it is believability. Non-photorealistic rendering techniques offer a broad variety of styles that still need to be explored in future games.

The question what comes next is as old as humanity. Looking at game graphics, however, opens up some exiting possibilities we can only oracle about. As we have shown in this article, there are many graphical elements that provide some degree of freedom during the development of a game. They are mostly applied in a standardized fashion, depending on the targeted game genre. Here, realism is one of the most interesting ones as realism is already bent in many ways in all kind of games. As games do not exactly reflect our own reality, some common aspects of reality e.g. the laws of physics or graphical appearance can be altered. By using non-photorealistic rendering techniques, these laws can be 'adjusted' even further without distraction, and due to the new graphical style, minor imperfections are tolerable and welcome.

The photorealistic high-end visualization paradigm will be an ideal for some time, further advancing – not with such quality leaps like in the last decades – but gradually approximating towards real photorealism. However, only developers with a big budget will be able to achieve the state-of-the art graphics, as it will become harder and more expensive to keep up. Consequently, graphics will no longer be able to serve as a unique selling point for games. Gamers might become saturated and will look for something different. This *different* on the graphics side can be filled using NPR techniques, which allow one to create a virtual reality that looks very different from our own. In addition, NPR techniques can also be used to support storytelling and to fulfill an artistic vision. Something that unleashes the power of dreams and fantasy and which allows us to drift away from our own world, just limited by our own imagination. Non-photorealistic rendering consists not only of toon shading and there is a broad variety of different graphical styles just waiting to be explored.

ACKNOWLEDGMENTS

Many thanks to Bert Freudenberg for discussion, proofreading and late night pizza.

REFERENCES

1. *Space Invaders*. Developed by Taito. Published by Atari. Platform: Atari 2600. 1980.
2. *Broken Sword: The Shadow of the Templars*. Developed by Revolution Software, Ltd. Published by Virgin Interactive Entertainment. Platform: PC. 1996.
3. *Quake*. Developed by idSoftware. Published by idSoftware. Platform: PC. 1996.
4. *Spindizzy*. Developed by Electric Dreams. Published by Electric Dreams. Platform: C64. 1986.
5. *Knight Lore*. Developed by Ultimate Play The Game. Published by Ultimate Play The Game. Platform: C64. 1984.
6. *Day of the Tentacle*. Developed by Lucas Arts. Published by Lucas Arts. Entertainment. Platform: PC. 1993.
7. *Stupid Invaders*. Developed by Xilam. Published by Ubisoft Entertainment. Platform: PC. 2001.
8. *Escape from Monkey Island*. Developed by Lucas Arts. Published by Lucas Arts. Platform: PC. 2000.
9. *Thief – The Dark Project*. Developed by Looking Glass Studios. Published by Eidos. Platform: PC. 1998
10. *Zork: The Great Underground Empire*. Developed by Infocom. Published by Infocom. Entertainment. Platform: AppleII. 1980.
11. *Donkey Kong*. Developed by Coleco. Published by Nintendo. Platform: Atari2600. 1982.
12. *Tomb Raider*. Developed by Core Design. Published by Eidos Interactive. Platform: PC. 1996.
13. *Battlezone*. Developed by Atari. Published by Atari. Platform: Atari2600. 1983.
14. *Myst*. Developed by Cyan, Red Orb Entertainment. Published by Brøderbund. Entertainment. Platform: PC. 1995.
15. *Half-Life 2*. Developed by Sierra. Published by Sierre. Platform: PC. 2003.
16. Jason Rubin, Great Game Graphics… Who cares?, Talk at GDC 2003
17. Frank Thomas and Olli Johnston, The Illusion of Life, Hyperion 1981
18. Nick Halper, Mara Mellin, Christoph Hermann, Volker Linneweber, Thomas Strothotte. Psychology and Non-Photorealistic Rendering: The beginning of a beautiful relationship. To appear in: *Mensch und Computer 2003*.
19. *XIII*. Developed by UbiSoft. Published by UbiSoft. Platform: PC. 2003.
20. *Cel-Damage*. Developed by Pseudo Interactive. Published by Electronic Arts. Platform: XBox. 2001
21. *PaRappa the Rapper*. Developed by NaNaOn-Sha. Published by SCEI. Platform: PlayStation. 1996.
22. *Jet Set Radio Future*. Developed by Smilebit. Published by Sega. Platform: XBox. 2002.

23. Lena Petrovic, Brian Fujito, Lance Williams, Adam Finkelstein: Shadows for cel animation. *Proceedings of SIGGRAPH 2000*: 511-516

24. Adam Lake, Carl Marshall, Mark Harris, and Marc Blackstein. Stylized rendering techniques for scalable real-time 3d animation. *International Symposium on Non-Photorealistic Animation and Rendering* (NPAR 2002)

25. E. Praun, H. Hoppe, M. Webb, and A. Finkelstein, "Realtime hatching," in Proceedings Computer Graphics (ACM SIGGRAPH), 2001, pp. 581--586.

26. Bert Freudenberg, Maic Masuch, Thomas Strothotte: Real-Time Halftoning: A Primitive For Non-Photorealistic Shading. *In Rendering Techniques 2002, Proceedings 13th Eurographics Workshop*, pp. 227-231, 2002. ISBN 1-58113 534-3

27. Bert Freudenberg, Maic Masuch, and Thomas Strothotte, Walk-Through Illustrations: Frame-Coherent Pen-and-Ink Style in a Game Engine. *In Computer Graphics forum*, vol. 20 (2001), no. 3. ISSN 0167-7055 Manchester, UK, 2001

28. Alex Mohr, Michael Gleicher. HijackGL: Reconstructing from Streams for Stylized Rendering. *International Symposium on Non-Photorealistic Animation and Rendering* (NPAR 2002)

29. *Pencil Whipped*. Developed by ChiselBrain Software. Published by ChiselBrain Software. Platform: PC. 2000.

30. B. J. Meier. Painterly rendering for animation. *Proceedings of SIGGRAPH 96*, pages 477–484, 1996.

31. Derek Cornish, Andrea Rowan, and David Luebke, View-Dependent Particles for Interactive Non-Photorealistic Rendering., Proceedings of Graphics Interface 2001 (June 2001).

32. Peter Krüger, Echtzeit-Rendern goßer visuell homogener Datensätze in nicht-fotorealistischen Landschaften, Master's Thesis, Otto-von-Guericke Universität Magdeburg, 2003

33. A. Majumder and M. Gopi. Hardware accelerated real time charcoal rendering. In NPAR 2*002: Symposium on Non Photorealistic Animation and Rendering*, June 2002.

34. Mario Costa Sousa and John W. Buchanan. Computer-generated graphite pencil rendering of 3d polygonal models. *Computer Graphics Forum*, 18(3):195--208, September 1999.