Photorealism or/and Non-Photorealism in Augmented Reality

Michael Haller *
Upper Austria University of Applied Sciences
Media Technology and Design, Austria

Abstract

Actual graphic hardware becomes more and more powerful. Consequently, virtual scenes can be rendered in a very good quality integrating dynamic behavior, real-time shadows, bump mapped surfaces and other photorealistic rendering techniques. On the other hand, non-photorealistic rendering became popular as well, because of its artistic merits. An integration in an AR environment is the logical consequence. In this paper we present an AR framework that uses photorealistic rendering as well as non-photorealistic rendering techniques. The prototypes based on these techniques show the advantages and disadvantages of both technologies in combination with Augmented Reality.

CR Categories: I.3.5 [Non Photorealistic Rendering]: Hardware Accelerated Rendering—Painterly Rendering H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities;

Keywords: Augmented Reality, Photorealistic rendering, Non-Photorealisic rendering

1 Introduction

A lot of effort has been investigated in improving Augmented Reality applications. Computer graphics has long been defined as a quest to achieve photorealism. As it gets closer to this grail, the field realizes that there is more to images than realism alone. In this paper we want to give attention to the visualization part of AR applications, where we want to answer the question if we should investigate more effort on technologies to enhance the realism of an Augmented Reality application or it would be more successful for the application to use more abstract techniques, like Cartoon-shaded scenes to emphasize the augmented objects in the real environment.

In contrast to Virtual Reality applications or games, where the whole environment is virtual, we have the real world in the Augmented Reality environment and the fact that this real world can't be blended off. Consequently, we have two possibilities to implement an artificial, superimposed world. On the one hand we have the possibilities to make the augmented world as realistic as possible (with a seamless integration of the virtual objects into the real world), on the other hand we can create an artificial, non-photorealistic world with an artistic background. Glassner showed in [Glassner 2003] some interesting abstract visualized approaches for highlighting the objects in an AR environment. Or do we want to have an artificial world that is indistinguishable from the reality? During our tests with different users we discovered that both, the photorealistic rendering techniques and the implementation of artistic, non-photorealistic rendered (NPR) scenes, are important and must be considered for AR representation.

However, NPR technique in an AR scene allows a more open, more flexible and more interaction with the scene. The user is not amazed if an object has a different (non realistic) behavior - the artificial world has to be at least believable and it must be more convincing. In addition, users expect a realistic behavior if the world is rendered photo-realistic. When talking of NPR applications, we think of more artistic environments. Non-photorealistic pictures can be more effective at conveying information, more expressive or more beautiful. Especially in AR, where the superimposed objects are not part of the real world, the virtual objects should be more convincing than realistic.

Mark Bolas discovered in his research at Stanford that while realistic environments help to engage the user and create a sense of being there, greater abstraction engages the user's senses and imagination to create a greater sense of being elsewhere, "in" the world created.

Another aspect is, of course, if we should limit our approach only to the visual point or if we should see it in a more open way. Ferwerda distinguished in [Ferwerda 2003] three different varieties of realism:

- physical realism, where the virtual objects provide the same visual simulation as the real scene.
- photorealism, where the image produces the same visual response as the scene, and
- functional realism, in which the image provides the same visual information as the scene.

In our case we are concerning only the photorealism aspect - even if physical realism is as important as photorealism. Currently, we are working on the integration of the physics engine ODE into our AR framework. Consequently, the augmented objects react and act as they would do in a real environment.

What's the main goal of an Augmented Reality application? While superimposing virtual objects onto the real picture, the application wants to get the users' attention. Especially when we experiment with AR in everyday settings, the usability part of an AR applications becomes very essential. How can we focus the attention of the user to the superimposed objects? How can we underline the augmented objects?

2 Related Work

To achieve a high quality immersive impression, a number of different directions have to be developed. Roussou et al. present in

*e-mail: haller@fh-hagenberg.at

In ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry, VRCAI 2004, pp. 189-196, Singapore.

[Roussou and Drettakis 2003] the following list that includes the points that are important to augment the immersive impression:

- Consistent lighting,
- use of shadow,
- the realness factor of the virtual experiences (e.g. real-time sound simulation), and
- virtual people.

This list is aimed for VE based applications. We believe that these factors are important, but there are missing some others that are very important for an AR environment. Moreover, we want to distinguish between photorealistic effects concerning the visualization and realistic behavior that doesn't influence the visualization at all (e.g. physical realism).

Durand describes in [Durand 2002] that the border between photorealism and non-photorealism can be fuzzy and the notion of realism itself is very complex. Another interesting conclusion of his paper is that the virtual world has to be interpreted more convincing rather than realistic. In other words: it is enough if the virtual, augmented, not existing world, *superficially* looks real. It should be a *believable* world. The augmented objects should be expressive, clear, and look aesthetically perfect.

Agusanto et al. present in [Agusanto et al. 2003] the seamless integration of virtual objects in an AR environment using image-based lighting techniques and environment illumination maps. Thereby, the authors postulate a consistent and a coherent virtual world with respect to the real environment.

Finally, Bimber et al. postulate in [Bimber et al. 2003] a consistent lighting situation between the real and the virtual objects to achieve a convincing augmented reality application.

3 Photorealism in Augmented Reality

We developed an AR framework that aims to make AR scenes as realistic as possible. Our framework is based on the AR software library ARToolKit [Kato et al. 1999]. Most of the current AR applications are dealing with the improvement of tracking and with the design of tangible user interfaces. But most of these applications don't give attention to the improvement of rendering quality of the augmented objects. Current computer games are great examples, how realistic artificial, not existing 3D objects can look like. In the following sections we present several rendering stages, where the AR rendered scenes can dramatically be improved. Hence, we start with shading, the integration of shadows (shadows are the first step for making a scene more realistic), and finally the combination of AR with bump mapped virtual objects.

3.1 Shading

Consistent and matching shading is one of the first effects that have to be implemented to achieve a realistic environment. Flat shaded objects don't look realistic - but sometimes this technique could be used to focus the user's attention to augmented objects. One of the key factors is the placement of the virtual light source. It is interesting to see that the location of the real light source can be difficult to estimate. Thus, it doesn't matter if the virtual light source doesn't 100% correspond to the real light source. The only thing that is important is that there should be a light source and diffuse/specular shading. In our case we implemented a Blinn shading in combination with textures.

3.2 Shadows

Concerning photorealism in AR, lighting and shadowing are very important techniques to achieve realistic environments. Using the right lighting model allows the implementation of more plastic and therefore more realistic object surface. In addition, shadows can dramatically improve the realism. Sugano et al. and Madsen et al. underline in the importance of shadows in an Augmented Reality scenario [Sugano et al. 2003][Madsen et al. 2003]. As mentioned by Naemura et al. [Naemura et al. 2002a], the consistency of geometry, time (synchronized world to facilitate a smooth interaction), and illumination is an important issue for Augmented Reality applications. Shading and shadows in both worlds must match to achieve a natural merge [Naemura et al. 2002b].

Shadows add a level of realism to a rendered image. Moreover, they are used as visual clues to determine spacial relationships and real-time shadows will gain importance in current real-time computer graphics applications for this reason. As described in [Naemura et al. 2002b] shadows are a very essential factor for the 3D impression of a scene. The seamless merging of the virtual world and the real world that we live in, is a challenging topic in current Augmented (Mixed) and Virtual Reality research. There is a great deal of key issues that enhance the immersive feeling of the user. Shadows are essential to improvement the visual perception. Moreover, they enhance the 3D impression in a way that the users get a better immersive 3D feeling - We believe that shadows are important in the same way as 3D stereo HMD, because users can sense more exactly the distance between two virtual objects. As a result, the interaction and manipulation of objects get improved [McCool 2000].

Figure 1: A virtual ball projects a shadow onto a virtual book and onto the real desk.

In our prototype we developed the shadow volume algorithm instead of the shadow map algorithm, because we didn't want to destroy the impression by using *big rasterized* shadow map pixels. Twenty-five years ago, Crow published the shadow volume approach for determining shadowed regions of a scene [Crow 1977]. We implemented a modified real-time shadow volume algorithm that can be used in an Augmented/Mixed Reality application. A closer description of our modified shadow volume algorithm can be found in [Drab 2003; Haller et al. 2003].

In contrast to [Bimber et al. 2003], we identify four different types of shadow in an AR environment. This fact results because of merging the virtual lights with the real lights. Figure 3 shows in the first two pictures the shadow cast in a real world and in a virtual world. The third picture depicts a complete but isolated shadow

Figure 2: A virtual sphere projects a shadow onto the real mug.

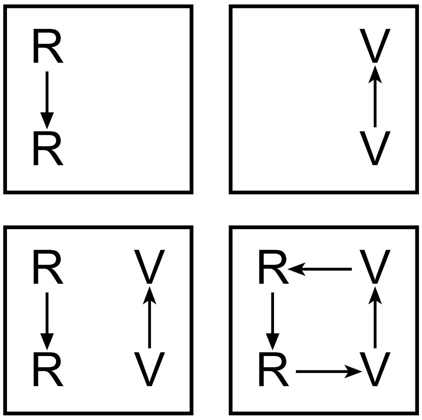


Figure 3: We distinguish four possibilities of shadow casts. The last figure shows the shadow cast in an Augmented Reality application.

cast of real objects onto real objects (the same for virtual objects). Finally, the last image shows the shadow cast in an Augmented Reality scenario. Given one light source, a virtual object can cast a shadow onto a real and onto a virtual object. Moreover, a real object supports shadows onto virtual objects.

The results of our prototype *shadowAReality* were quite convincing and impressive. Figures 1 2 and 4 shows some examples. In our implementation all real objects have a corresponding 3d geometry representation, a so called *phantom model*. In fact, the real scene has to be implemented as a 3D model. However, it is sufficient if the 3D model (we used 3ds loader based on [DigiBen 2001]) corresponds approximately to the real object. The position and orientation tracking of the real object is accomplished by using ARToolKit.

In our prototype, the shadow of the real/virtual objects casted onto the real/virtual objects was achieved by shading the scene with black-transparent color blending. In the beginning we worried about the problems with the real shadows onto the real objects. Much to our surprise we found that this did not become a problem. The 'virtual' shadows (casted by virtual/real objects) onto virtual/real objects did not interfere with the real environment. In most situations, users do not even recognize the differences between real and virtual generated shadows (cf. figure 4).

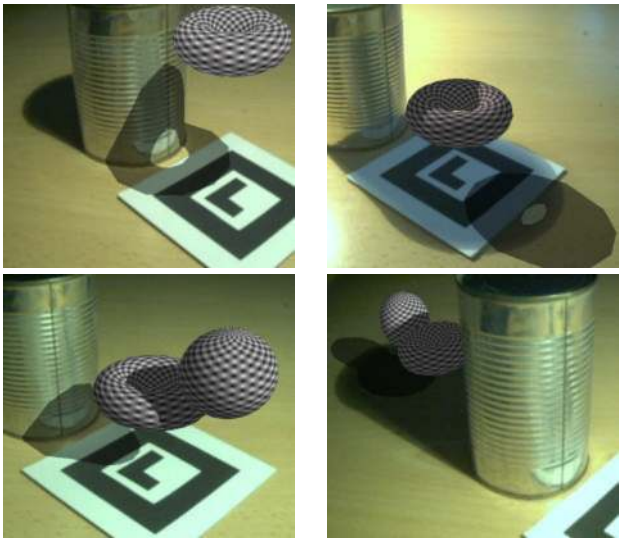


Figure 4: The first figure shows a virtual shadow cast onto a real tin. The second figure depicts the shadow of the tin onto the torus. Next, we have two virtual objects (sphere and torus) casting a shadow onto the tin and finally, the tin casts a shadow onto the torus and onto the sphere depicted in the last snapshot.

3.3 Bump Mapping

Bump mapping simulates the bumps or wrinkles in a surface without the need for geometric modifications to the model and it improves the textured surfaces. In fact, it is a technique to add more realism to synthetic images without adding a lot of geometry. The results are convincing and very impressive. In our AR framework we implemented bump mapping by using a vertex shader and a fragment shader. Moreover, we integrated nVIDIA's NVMeshMender software that allows the calculation of the tangents for a given 3D model. The results of this implementation are depicted in figure 5.

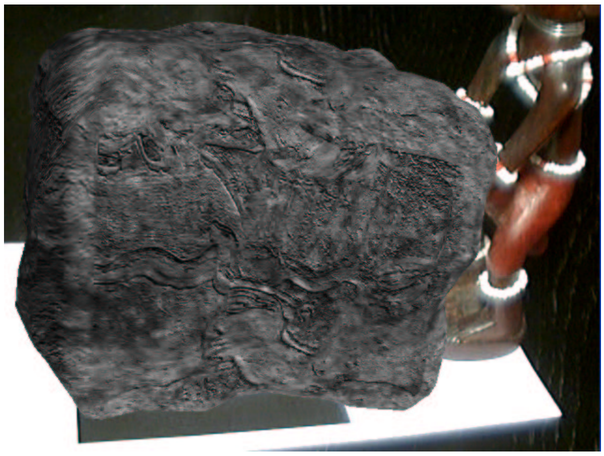


Figure 5: A stone with bump mapped texture.

4 Non-Photorealism in Augmented Reality

The main focus in computer graphics has always been the rendering of photorealistic imagery. While a tremendous amount of research

(a) The traditional visitor has no interaction with the painting.

(b) Our MR application allows the visitor to see a small model of the painting.

(c) Different shaders (e.g. Cartoon-shaders) allow experimenting with the model.

Figure 6: MR allows the visitors to have more fun in a museum. Non-photorealistic rendering techniques allow great aesthetic possibilities.

has been done in this area, non-photorealistic rendering is a relatively young field of research. In many areas, a NPR approach can be more efficient than photorealism, since it can focus on the information to be conveyed. As described in [Fernando and Kilgard 2003] objects in wireframe, flat shading or NPR shading are easily discernible. Consequently, Cartoon-shading and painterly rendering have become very important not only in game design [Kenichi Anjyo and Katsuaki Hiramitsu 2003], but also in architecture [Freudenberg et al. 2001] and even in medical applications [Feng Dong et al. 2003].

In paintings, a scene represents an artists' view of the world. All the information they want to convey with the painting has to be assembled by strokes of a brush. The attributes of each stroke can affect various characteristics of the painting. The size of a stroke determines the maximal detail of an object, direction and color describe the quality of the surface. By choosing different brush strokes, a painter can emphasize the parts of the scene he/she considers most important and create a certain mood and/or atmosphere.

Many algorithms have been developed in the last decade that create images which resemble art made by humans [Gooch et al. 2002; Hertzmann and Perlin 2000]. Different art techniques and styles can be simulated, such as pen and ink [Winkenbach and Salesin 1994; Salisbury et al. 1994], hatching [Praun et al. 2001], water color [Curtis et al. 1997] and impressionism [Haeberli 1990; Meier 1996]. The fast progress of graphics hardware affordable for consumers is allowing more and more of these algorithms to be processed in real-time (e.g. [Kaplan et al. 2000; Lander 2000; Markosian et al. 1997; Majumder and Gopi 2002]).

NPR is mainly used in AR when we want to describe an abstract information that is not representable or if we want to focus on a special detail. Thinking of a complex machinery where we want to highlight missing parts. In this case more abstraction makes sense and should be used. If we want to have a smooth transition from the real environment to the virtual superimposed objects, we have to use the photorealistic metaphor (cf. if we want to see our new virtual furniture in our apartment).

Figure 6 depicts a future scenario of a possible museum's application, where the visitors don't have a flat 2d impression of the painting. In contrast, they get the possibility to *immerse into* the painting. Moreover, in our application we support different NPR rendering techniques, such as toon-rendering and painterly rendering. Both rendering techniques are described more closely in the next sections.

4.1 Toon Shading

One possibility of non-photorealistic rendering is Cartoon-style rendering (cf. [Lander 2000]). The idea is to present the augmented data in a certain style that does not necessarily have a real-life analogue. The objects are rendered with a constant, sharply delineated color with an outline of the object. In our case we replace the diffuse and specular lighting colors with a two-valued step function, where the values are calculated by using a 1D texture as a lookup table. Mathematically, we can say:

$$I = (k_d(N \cdot L) < \varepsilon) \quad (1)$$

where ε is a kind of threshold.

$$\sum_{u=0}^{1} ShadeTable[u] = (k_d(u) < \varepsilon) \quad (2)$$

Our Cartoon-rendering technique was implemented with nVIDIA's vertex and fragment shader language Cg. Figure 7 shows a result of the Cartoon-renderer.

Figure 7: Close-up of the cartoon rendered painting of Van Gogh's Bedroom at Arles.

Using a vertex and a pixel shader for the renderer resulted very good performance. In combination with the ARToolKit we received about 56 fps using a model with about 11.000 polygons. In our application we achieve two advantages: firstly, the visitors of a museum have a three dimensional view of the painting and secondly, they can change the rendering style (as depicted in figure 7).

4.2 Painterly Rendering

As described before, our framework allows a flexible change of the rendering technique. At the beginning, we wanted to implement only a painterly rendering algorithm that runs in real-time. Our rendering technique is based on B. J. Meier's "Painterly Rendering for Animation" [Meier 1996]. We modified the algorithm for real-time environments by extensively using modern 3d hardware. Vertex and pixel shaders allow both the rendering of thousands of brush strokes per frame and the correct application of their properties.

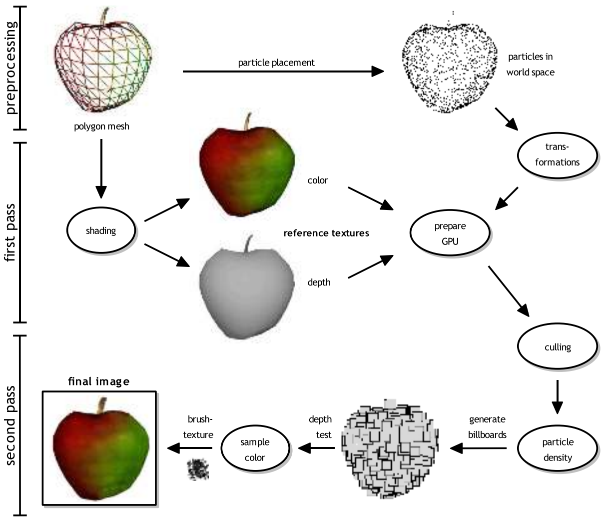


Figure 8: The pipeline of the painterly renderer.

Our modified painterly algorithm can be divided into three steps:

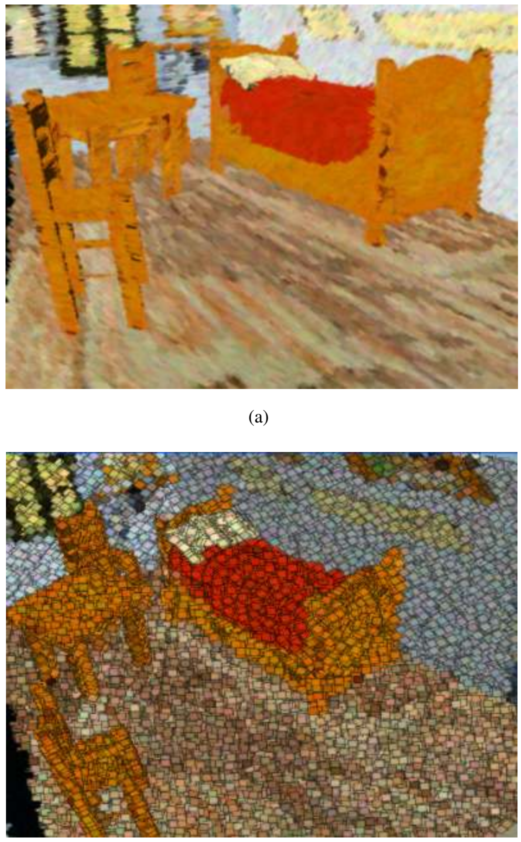
1. Before we can start to render the scene, we need a **preprocessing** step. The polygon meshes are converted into 3d particles. The attributes of brush strokes that remain constant in each frame can be computed simultaneously.

2. Next, the rendering can start and it takes at least two passes. During the **first pass**, two reference images that contain color and depth information are rendered and stored into textures. An arbitrary rendering algorithm can be used.

3. In the **second pass**, an optimal amount of brush strokes is rendered using billboards. Reference pictures from the first pass are used to define color and visibility, while the surface normal determines the orientation of each stroke. This is the most expensive part of the rendering process, since a large number of strokes needs to be processed.

The graphics-pipeline of the described system is shown in figure 8. A closer description of the modified algorithm can be found in [Haller and Sperl 2004; Sperl 2003].

Direct rendering to textures allows rapid generation of reference pictures and assures great flexibility, since arbitrary rendering systems can be combined (e.g. painterly rendering of toon shaded objects, etc.). A result scenario of the algorithm is depicted in figure 9.



(a)

(b)

Figure 9: Close-up of Van Gogh's bedroom model rendered with the painterly rendering algorithm using different strokes.

5 Results and discussions

The results of the presented applications are presented in this section.

5.1 Discussions

As we have seen in the implemented examples described above, both approaches, the photorealistic rendering as well as the non-photorealistic rendering has its value in an AR application. The question we have to answer is if it is more important to achieve a greater visual realism through photorealistic or non-photorealistic visualization. First of all, NPR makes sense in AR, because the overlapped objects are indirectly emphasized. Moreover, non-photorealistic rendering, like Cartoon-style rendering, can be useful for CAD based AR applications as well as for its artistic merits (cf. [Dietrich 2000]). On the other hand, sometimes the non-realistic approach and the non perfect rendering style shouldn't influence the overall picture of the scene. Consequently, a non perfect behavior of an augmented virtual object would interfere with the perfect world.

Very often, when talking with end users, we came to the conclusion that they wanted to have something absolute different what the AR developers wanted to have to. In the EU funded project AMIRE (Authoring Mixed Reality) we implemented an AR maintenance application for refinery employees. The interesting thing was that the 3D objects looked nice for the AR experts and it was

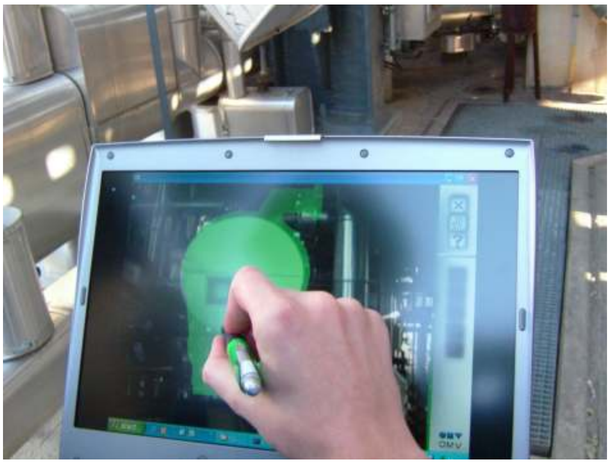


Figure 10: In our refinery application the employees are using a Tablet PC. The real objects are highlighted in green. Consequently the users know that they can interact with the object.

a very interesting feature for the specialists (e.g. engineers) to have an overlapping 3D object onto the corresponding real object. However, for a refinery employ, the content is more important. They are familiar with schemes and 2D CAD maps and they would miss such abstract information in an AR application. Currently, we are working on an iPAQ version for the AR maintenance application that should be used in the refinery. Due to the small size of the display and the performance problems a photo-realistic rendered scenario is currently absolutely impossible.

Navab presented in [Navab 2003] an interesting visualization metaphor for an industrial AR application: The drawing, images, and 3D models, in fact the 2D CAD results, have beeon projected onto the real 3D objects of the real environment. Employees from the industry like that metaphor, because they know how to read CAD drawings. The same phenomena can be found in the architecture area. An architect don't like to see a perfect 3D high resolution textured object, but in contrast they want to see it more abstract. Consequently, Cartoon-shaders became more and more important for this kind of applications. On the other hand, customers want to have a 3D object with bump mapped surfaces.

Finally, we implemented a virtual sculpture in the same real environment but with different rendering modes to find out, which one could be the best for AR (cf. figure 11). We haven't done intensive tests with more people yet. But already now we can say that people like the NPR rendering techniques in AR. The Cartoon-renderer was the top favorite - even if it was less complex to implement than the painterly renderer. The transparent mode is important for objects that should be highlighted (as seen in our refinery application). Interesting was to see that people liked more the wireframe rendered sculpture instead of the Gouraud shaded model.

5.2 Conclusions

Thanks to the graphics hardware we have two possibilities in an AR environment: both approaches, the photorealistic as well as the non-photorealistic rendering techniques work in real-time. We have described both approaches - both applications are implemented using the ARToolKit library and nVIDIA's Cg language. In the photorealistic AR application we discussed the enhancement of impression using shadow volumes and bump mapping. In addition, we presented the non-photorealistic AR version using a Cartoon-style renderer and a painterly renderer. We have seen that non-photorealistic rendering techniques are not only used for achieving artistic effects. Of course, it depends on the type of the applications and each AR application has its domain focus - consequently, we have different end users and different goals that require different rendering techniques. The visual improvements depend on the application (e.g. in a MR/AR based furnishing application the virtual furniture objects shouldn't destroy the overall impression). We have also seen, that in AR, the user should immediately be confronted with the abstracted model - in fact, with the superimposed, augmented object. Therefore, it would be very helpful, if his/her attention directly jumps to the superimposed object (e.g. by using NPR). We fully agree with the statement of Durand, where he says in [Durand 2002] that the virtual world has to be interpreted more *convincing* rather than *realistic*. The world has to look *superficially* real and believable - not only in the visual sense - we should also think in other dimensions, such as integrating physical behavior of objects.

5.3 Acknowledgement

We would like to thank Daniel Sperl for the implementation of the painterly renderer, Stephan Drab for the implementation of the ShadowAReality application, and finally many thanks to J"urgen Zauner for the great integration of the NVMeshMender class into the 3ds loader. This work is supported by the European Union, IST-2001-34024.

References

AGUSANTO, K., LI, L., CHUANGUI, Z., AND SING, N. W. 2003. Photorealistic rendering for augmented reality using environment illumination. In *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR 2003)*, IEEE Computer Society, 208–216.

BIMBER, O., GRUNDHOEFER, A., WETZSTEIN, G., AND KNOEDEL, S. 2003. Consistent illumination within optical see-through augmented environments. In *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR 2003)*, IEEE Computer Society, 198–207.

CROW, F. 1977. Shadow Algorithms for Computer Graphics. In *Proceedings of SIGGRAPH 77*, 242–248.

CURTIS, C. J., ANDERSON, S. E., SEIMS, J. E., FLEISCHER, K. W., AND SALESIN, D. H. 1997. Computer-generated watercolor. In *Proceedings of SIGGRAPH 97*, 421–430.

DIETRICH, S. 2000. Cartoon rendering and advanced texture features of the geforce 256 texture matrix, projective textures, cube maps, texture coordinate generation and dotproduct3 texture blending. Tech. rep., nVIDIA.

DIGIBEN, 2001. GameTutorials, URL: http://www.gametutorials.com/TermsOfUse.htm.

DRAB, S. 2003. *Echtzeitrendering von Schattenwurf in Augmented Reality Anwendungen*. Master's thesis, Upper Austria University of Applied Sciences, Media Technology and Design.

DURAND, F. 2002. An invitation to discuss computer depiction. In *Proceedings of the second international symposium on Non-photorealistic animation and rendering*, ACM Press, 111–124.

FENG DONG, GORDON J. CLAPWORTHY, HAI LIN, AND MELEAGROS A. KROKOS. 2003. Nonphotorealistic Rendering of Medical Volume Data. *IEEE Computer Graphics and Applications 23*, 4 (July/August), 44–52.

FERNANDO, R., AND KILGARD, M. J. 2003. *Cg Tutorial, The: The Definitive Guide to Programmable Real-Time Graphics*. Addison-Wesley.

FERWERDA, J. 2003. Three varieties of realism in computer graphics. In *Proceedings SPIE Human Vision and Electronic Imaging '03*.

FREUDENBERG, B., MASUCH, M., ROEBER, N., AND STROTHOTTE, T. 2001. The Computer-Visualistik-Raum: Veritable and Inexpensive Presentation of a Virtual Reconstruction. In *In Proceedings VAST 2001: Virtual Reality, Archaelogy, and Cultural Heritage*, 97–102.

GLASSNER, A. 2003. Everyday computer graphics. *IEEE Comput. Graph. Appl. 23*, 6, 76–82.

GOOCH, B., COOMBE, G., AND SHIRLEY, P. 2002. Artistic vision: painterly rendering using computer vision techniques. In *Proceedings of the second international symposium on Non-photorealistic animation and rendering*, ACM Press, 83–ff.

HAEBERLI, P. E. 1990. Paint by numbers: Abstract image representations. In *Proceedings of SIGGRAPH 90*, 207–214.

HALLER, M., AND SPERL, D. 2004. Real-time painterly rendering for mr applications. In *Graphite, International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*, to be published.

HALLER, M., DRAB, S., HARTMANN, W., AND ZAUNER, J. 2003. A real-time shadow approach for an augmented reality application using shadow volumes. In *ACM Symposium on Virtual Reality Software and Technology*, 56–65.

HERTZMANN, A., AND PERLIN, K. 2000. Painterly rendering for video and interaction. In *Proceedings of the first international symposium on Non-photorealistic animation and rendering*, ACM Press, 7–12.

KAPLAN, M., GOOCH, B., AND COHEN, E. 2000. Interactive artistic rendering. In *Proceedings of the first international symposium on Non-photorealistic animation and rendering*, ACM Press, 67–74.

KATO, H., BILLINGHURST, M., BLANDING, B., AND MAY, R. 1999. ARToolKit.

KEN-ICHI ANJYO, AND KATSUAKI HIRAMITSU. 2003. Stylized Highlights for Cartoon Rendering and Animation. *IEEE Computer Graphics and Applications 23*, 4 (July/August), 54–61.

LANDER, J. 2000. Shades of disney: Opaquing a 3d world. *Game Developers Magazine* (March).

MADSEN, C. B., SØRENSEN, M. K. D., AND VITTRUP, M. 2003. The importance of shadows in augmented reality. In *Proceedings: 6th Annual International Workshop on Presence, Aalborg, Denmark*, (4 pages). To appear.

MAJUMDER, A., AND GOPI, M. 2002. Hardware accelerated real time charcoal rendering. In *Proceedings of the second international symposium on Non-photorealistic animation and rendering*, ACM Press, 59–66.

MARKOSIAN, L., KOWALSKI, M. A., TRYCHIN, S. J., BOURDEV, L. D., GOLDSTEIN, D., AND HUGHES, J. F. 1997. Real-time nonphotorealistic rendering. In *Proceedings of SIGGRAPH 97*, 415–420.

MCCOOL, M. D. 2000. Shadow volume reconstruction from depth maps. *ACM Transactions on Graphics 19*, 1, 1–26.

MEIER, B. J. 1996. Painterly rendering for animation. In *Proceedings of SIGGRAPH 96*, 477–484.

NAEMURA, T., NITTA, T., MIMURA, A., AND HARASHIMA, H. 2002. Virtual Shadows - Enhanced Interaction in Mixed Reality Environment. In *IEEE Virtual Reality (VR'02)*.

NAEMURA, T., NITTA, T., MIMURA, A., AND HARASHIMA, H. 2002. Virtual shadows in mixed reality environment using flashlight-like devices. *Trans. Virtual Reality Society of Japan 7*, 2, 227–237.

NAVAB, N. 2003. Industrial augmented reality (iar): Callenges in design and commerzialization of killer apps. In *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR 2003)*, IEEE Computer Society, 2–6.

PRAUN, E., HOPPE, H., WEBB, M., AND FINKELSTEIN, A. 2001. Real-time hatching. In *Proceedings of SIGGRAPH 2001*, 581.

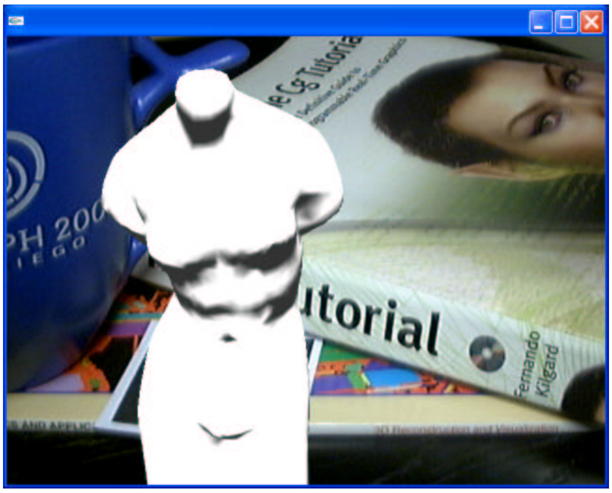
ROUSSOU, M., AND DRETTAKIS, G. 2003. Photorealism and non-photorealism in virtual heritage representation. In *VAST 2003 and First Eurographics Workshop on Graphics and Cultural Heritage*, A.Chalmers, D.Arnold, and F. Niccolucci, Eds., Eurographics.

SALISBURY, M. P., ANDERSON, S. E., BARZEL, R., AND SALESIN, D. H. 1994. Interactive pen-and-ink illustration. In *Proceedings of SIGGRAPH 94*, 101–108.

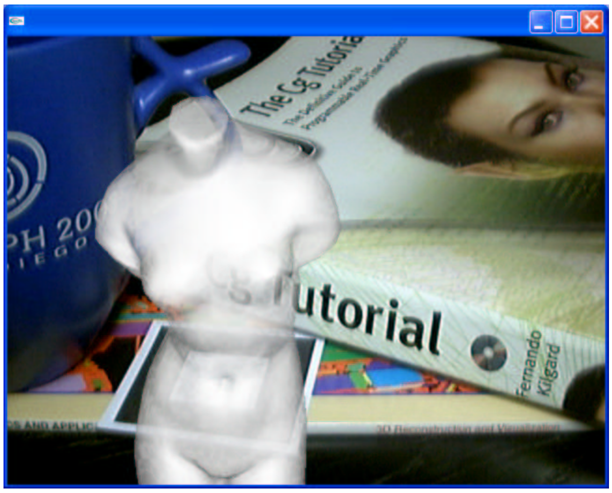
SPERL, D. 2003. *Kuenstlerisches Rendering fr Echtzeit-Applikationen*. Master's thesis, Upper Austria University of Applied Sciences, Media Technology and Design.

SUGANO, N., KATO, H., AND TACHIBANA, K. 2003. The effects of shadow representation of virtual objects in augmented reality. In *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR 2003)*, IEEE Computer Society, 76–83.

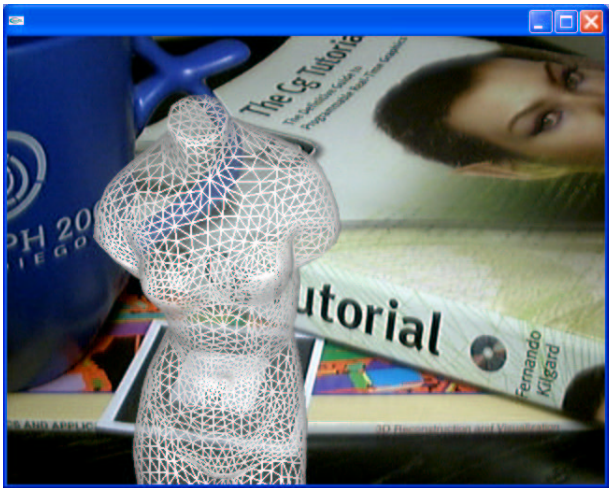
WINKENBACH, G., AND SALESIN, D. H. 1994. Computer-generated pen-and-ink illustration. In *Proceedings of SIGGRAPH 94*, 91–100.



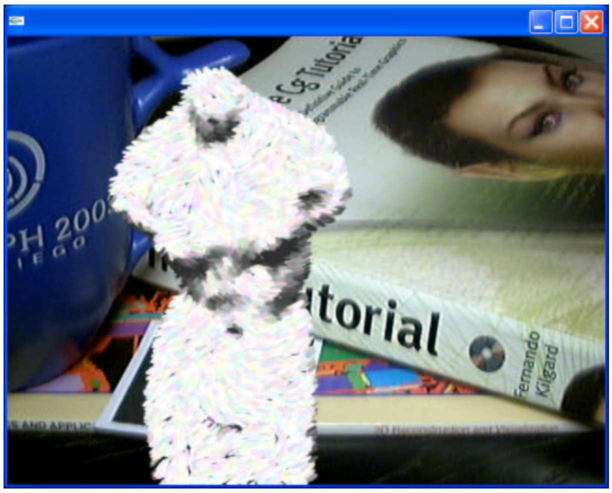
(a) This is the most frequently used technique in AR: a Gouraud shaded sculpture.



(b) Using transparency allows more artistic freedom. This technique can be used overlapping the real objects.

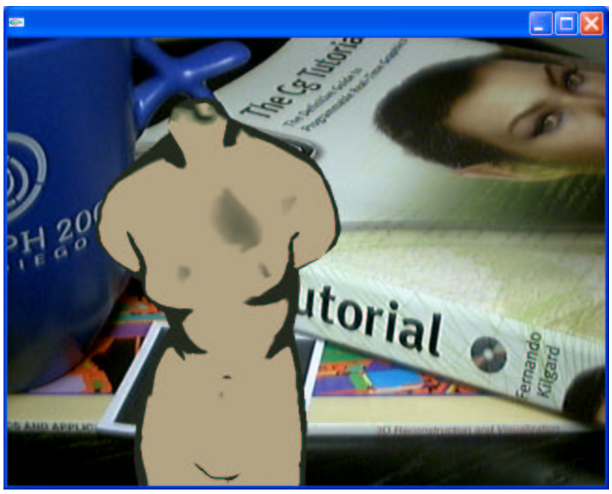


(c) A wireframe rendered sculpture can be very useful to enhance the augmented object if the performance of the system is too bad.



(d) This sculpture is rendered with painterly algorithm described in this paper. The oriented brush strokes in dependence of the surface normal can lead to a realistic painting effect.

(e) A cartoon rendered sculpture.



(f) Another cartoon rendered sculpture with a different texture for the step function.

Figure 11: A sculpture rendered with different algorithms shows the different possibilities we can achieve in Augmented Reality.