Technical Report No. 11/2004

Seeing Between the Strokes

Tobias Isenberg, Roland Jesse, Oscar Meruvia, Thomas Strothotte

{isenberg|jesse|oscar|tstr}@isg.cs.uni-magdeburg.de

Department of Simulation and Graphics

Otto-von-Guericke University of Magdeburg

Abstract

Non-photorealistic rendering offers a wide range of rendering styles. However, when different styles—in particular, stroke-based techniques—are combined with others to create hybrid renditions often a see-through-effect is caused where background objects can be seen through a newly drawn foreground object. This effect can either be used intentionally to present more than one layer of information at the same time or has to be avoided because it distracts from the rendering.

In this work, we attempt an initial discussion of the see-through-effect in illustrations. Thereby, a specific focus is put on stroke-based NPR renditions. In support of this, both ends of the effect are addressed: the intentional use of see-through characteristics as well as avoiding it. By discussing examples for both, we show how to visualize multiple layers of information in a model as well as approaches of avoiding the see-through-effect for stroke-based NPR rendering.

1. Introduction

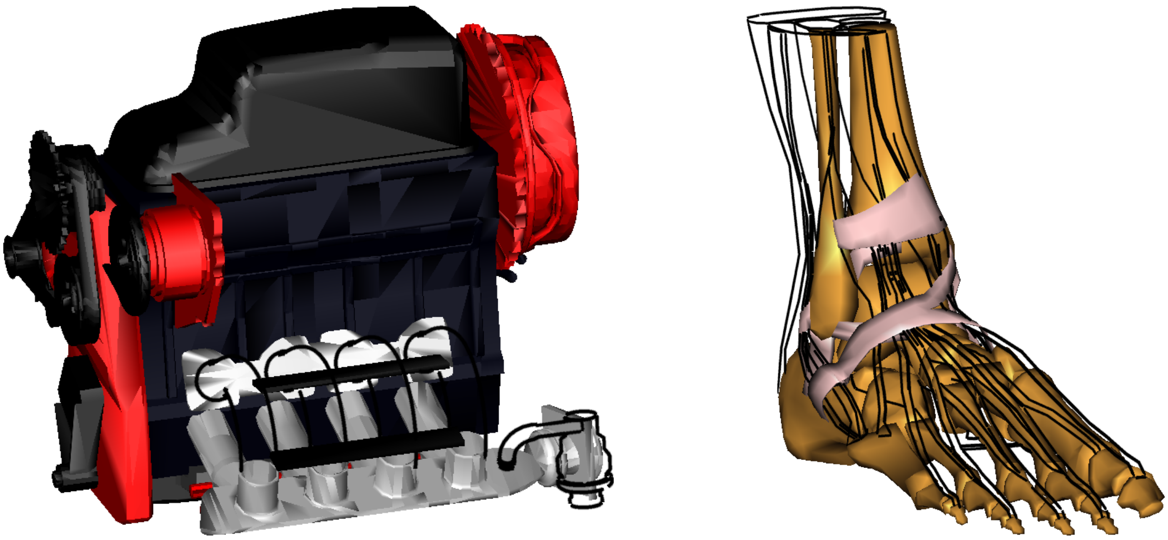
Today's computer graphics offer a great variety of ways to generate images. Especially the area of non-photorealistic rendering (NPR) has created a great number of different rendering styles. In NPR, various rendering styles and line stroke parameterizations can be used to achieve effects that abstract elements of the rendered objects. This helps to to emphasize specific scene components.

However, most rendering techniques were designed to be used exclusively. This typically leads to problems in cases when multiple rendering techniques or styles are composed to create hybrid renditions. An example is in combining certain non-photorealistic techniques with photorealism. This is due to the fact that, in contrast to photorealistic rendering, not all NPR techniques fill the entire presentation space covered by objects to which the techniques are applied. In particular, stroke-based techniques[1] that are characterized by applying isolated marks onto a surface use the background that shines through the marks as part of the created effect. For example, dark hatch marks require a light background to create a certain perceived brightness of the rendered shape. In contrast, other NPR techniques and most photorealistic methods do cover the background completely.

When using only one of these techniques, this usually does not pose a problem. However, when combining two of these stroke techniques or a stroke method with a photorealistic shading, problems may occur. If the stroke method is applied on top of another stroke method or on top of photorealistic shading, these structures in the background can be seen through the strokes in the foreground. There are even cases where the strokes in the foreground seem to disappear. This happens if the background has nearly the same color as the strokes in the foreground. In general, we define this as the *see-through-effect*.

Two specific examples of the see-through-effect as it might occur in hybrid non-photorealistic renditions are shown in Figure 1.1. In case of the engine as illustrated in Figure 1.1(a), tubes of the cooling system are emphasized by use of a stroke-based line style, that is not used for the rendering of the remainder of the model. However, since the color of the lines is close to the one of the engine's body the tubes seem to almost disappear. A similar although not as severe version of the see-through-effect is demonstrated for the medical model in Figure 1.1(b). In both of these cases, the see-through-effect needs to be avoided in order to achieve the envisioned visualization goal (see the two respective examples in Figure 1.2).

1 By the term *stroke-based techniques* we also mean, for example, stippling and cross-hatching.



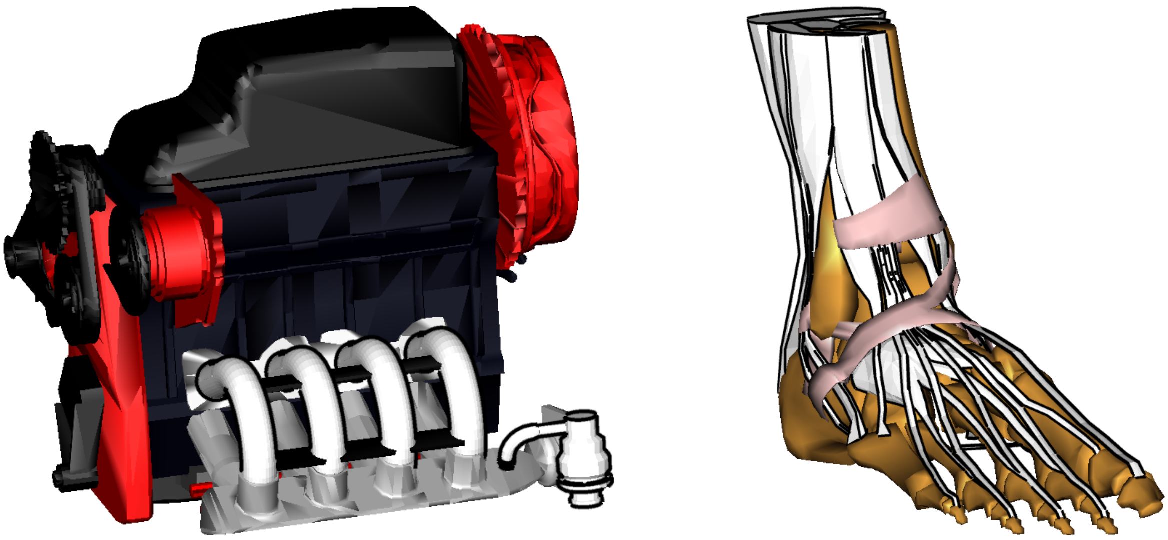
(a) (b)

Figure 1.1.: When the shading of an object is exchanged with a line-based NPR technique, the photorealistically rendered remainder of the model is seen in the background, which is very distracting for many applications.

However, there are situations where the see-through-effect may be desired. In these cases it could be used, for example, to visualize two layers of information instead of just one (see example in Figure 1.3). In order to distinguish both layers, the different rendering styles should enable the viewer to be able to tell them apart very easily. Ways of achieving this include use of two clearly distinct methods or using animation with different motion patterns for each of the two methods. In the example in Figure 1.3, two different densities of stippling were used combined with a silhouette and objects rendered in black.

In the remainder of this paper, we will discuss strategies necessary to either profit from the effect (i. e., use it to convey something) or avoid it (i. e., so that it does not disturb the rendition). First, we discuss some related work in the area of hybrid rendering in Section 2. This is followed by an analysis of the see-through-effect in Section 3. Therein, the notion of the effect is pointed out as well as some causes of it. Strategies of making use of the effect or avoiding it are outlined in Sections 4 and 5, respectively. Finally, Section 6 concludes this work.

Our main contribution lies in introducing the notion of the see-through-effect and in its initial discussion. A selection of possible causes of this effect are illustrated. This is accompanied by ways of making use of the it as well as a presentation of some techniques for avoiding it.



(a) (b)

Figure 1.2.: Using a shading to enhance the appearance of the line-based NPR style. In contrast to Figure 1.1, hidden surfaces do not shine through the non-photorealistically rendered parts of the models.

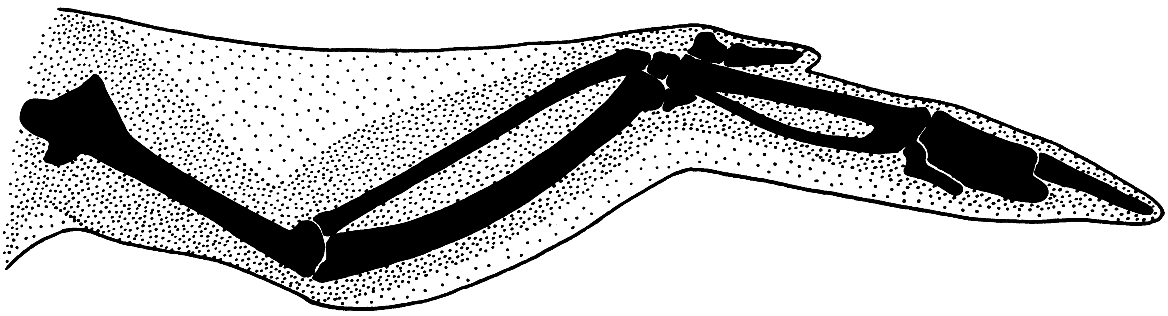


Figure 1.3.: Example for the use of the see-through-effect in a traditional illustration to visualize different layers of information. From [Hod89, , page 341].

2. Previous and Related Work on Hybrid Rendering

In recent years, non-photorealistic rendering has received a great deal of attention. Two books have been published that give a comprehensive overview about the subject [GG01, SS02]. In the following, a selection of approaches that combine several styles to create hybrid renditions is covered because those are the most relevant ones for the topic of this paper.

Saito and Takahashi were the first to add non-photorealistic elements—such as silhouettes—to photorealistically shaded objects in order to create more expressive renditions [ST90]. Gooch et al. combine an adapted photorealistic shading with non-photorealistic elements such as silhouettes to create a non-photorealistic lighting model [GGSC98]. Their main application domain is the automatic technical illustration. In [GSG+99], they explore techniques to speed up the computation of the lighting model in order to allow for interactive technical illustration. This means that in both cases a somewhat hybrid style is created, they apply it coherently to the whole scene. The rendering itself only uses one style. It is, thus, not hybrid in the sense of this work.

In contrast, Masuch and Strothotte combine different styles in one image [MS01]. They use a photorealistic rendering or photographic image as a background and add a non-photorealistically rendered object as foreground in order to visualize the uncertainty and the degree of trust in the reconstruction of ancient architecture. When using a photo, naturally they only produce still images (see also [SPM+99]). However, when including the NPR renditions into the photorealistic background, real-time animations can be created. On the other hand, they are not able to make a continuous transition from the photorealistic style to non-photorealism or vice versa.

Ritter et al. use additional silhouettes for semi-transparent objects in interactive medical illustrations to increase the user's ability to recognize these objects [RSHS03].

Halper et al. present an interface for easy combination of various different rendering styles [HSS02]. Being conceived as a tool for designers to assemble and experiment with renditions, it makes it easy to come up with new combinations of styles including the possibility to create hybrid renditions.

A different kind of hybrid animation is used in movies such as "Who Framed Roger Rabbit" (1988) or "Space Jam" (1996). In this type of movies, real footage is combined with hand-drawn cartoon characters. Johnston improves the appearance of the cartoon characters by creating toon renditions using the light positions from the real scene [Joh02]. When combining it with real video footage, the more coherent $2\frac{1}{2}$D look of the character with correct highlights is achieved by estimating normals from the hand-drawn silhouette using a set of heuristics.

Jesse and Isenberg describe a hybrid rendering system that combines non-photo-realistic styles with regular photorealistic shading techniques [JI03]. Based on this type of hybrid rendering, techniques for smooth blending between photorealistic and non-photorealistic visualizations are presented in [JINS04].

The natural see-through-effect of smoke is illustrated by [FSJ01]. In this work, smoke flows around a closed-surface object like a sphere. In order to demonstrate the flowing motion, the smoke needs to provide some see-through-effect. The same principle idea of using particle-based see-through-effects as intrinsic presentation characteristic holds for fog as well.

The effectiveness of varying transparencies in a scene is analyzed by [GDF03]. The goal is to evaluate the expressive power of transparency in order to direct user attention. A see-through-effect is used here in order to save screen space and point out scene elements of specific interest for user interaction. In a related user study, Zhai et al. examine use of semi-transparency for the interaction in 3D scenes [ZBM96]. In particular, they confirm that semi-transparency supports the recognition of relative depth information in a 3D scene.

Bier et al. introduce the related concept of the *see-through interface* [BSP$^+$93, BSP$^+$94, BSF$^+$94]. This uses stroke-based interface elements that use the see-through-effect in order to enhance the human-computer interaction. These can be combined with *magic lenses* that modify the representation of the underlying object parts, a concept that has also been extended to 3D [VCWP96]. For example, the magic lenses may allow to look through a surface object in order to show underlying parts.

Diepstraten et al. discuss another technique related to the see-through-effect: the generation of interactive cutaway illustrations [DWE03]. They demonstrate how the cutaway technique frequently used in technical illustrations can be applied to interactive 3D rendering based on set of rules they extracted from hand-made illustrations.

3. The See-Through-Effect

Although there have been relatively many works on the combination of rendering styles as discussed above, little thought has been given on how these styles may interact with each other. In particular, a systematic view on this interaction is lacking. We attempt to provide such a systematic discussion based on the see-through-effect as defined above.

The see-through-effect has two major sources: the use of either

- partial coverage styles (e. g., line stippling, hatching, and silhouettes) or
- non-opaque styles (e. g., transparency of smoke or fog as well as transparency for visualization purposes).

For both cases, the effect is achieved in case they are used together with other styles in a scene as opposed to be used exclusively for the whole model. Then, the area behind a newly rendered object is not completely covered. Thus, both types of rendering styles cause items behind objects to shine through. This can either be used on purpose or should be avoided as it can create unwanted results. Therefore, whenever either type of rendering style is used one has to be aware of the resulting see-through-effect. In the following, examples for each of the above mentioned causes of the see-through-effect will be discussed.

3.1. See-Through-Effect in Partial Coverage Styles

A common source of the see-through-effect caused by partial coverage styles is the simultaneous combination of multiple techniques in non-photorealistic renderings. These NPR techniques can roughly be divided into surface shading techniques and stroke- or line-oriented methods. Usually, both variants of NPR techniques can be used individually and exclusively, e. g., a cartoon shading can be used as well as just displaying the visible silhouettes of a model. However, a combination of NPR techniques with photorealistically rendered models eventually causes the see-through-effect. For example, using a photorealistically shaded model and only displaying one object of the model using silhouettes results in the rest of the model being visible through the silhouette lines as demonstrated in Figure 1.1.

Precisely, the effect occurs in case particular stroke techniques are combined with either some sort of shading technique or with other methods that also use strokes. The stroke techniques in question are of non-filling nature with respect to their effected surfaces. Among others, this holds for strokes, lines, and stipples.

Depending on the kind of hidden surface or line removal method being possibly used, the result of the see-through-effect is intended sometimes, as in combining cartoon shading with a silhouette technique. However, severe problems occur when one object is emphasized using a stroke technique while the rest of the model is still rendered using photorealistic methods such as shown in Figure 1.1. We will discuss methods for preventing the see-through-effect in such cases in Section 5.

3.2. See-Through-Effect in Non-Opaque Styles

As noted above, the second possible cause of the see-through-effect is the use of non-opaque styles. Similar to the partial coverage styles, the transparency of non-opaque styles does not cover the presentation space occupied by an object by 100%. This way, other objects or the background behind the newly, non-opaquely rendered object stay partially visible. Here, in most of these cases this effect is intended. For the example of simulating fog, the effect that objects in the background shine through is achieved on purpose. Also, when transparency of objects is used in certain visualizations in order to see objects that would otherwise been hidden, the effect is desired (see example in Figure 3.1).

A special case of the see-through-effect in non-opaque styles occurs when trying to remove it from visualizations mentioned in Section 3.1. A simple way to accomplish this is to add additional shading. For a smooth transition between the regularly shaded visualization and the one containing strokes and NPR shading, a dynamic blending of the object's shading from one type to another would be used. An example for a smooth transition between regular photorealistic shading of an object's surface to non-photorealistic shading is in continuously de-emphasizing one style while the other style is successively emphasized. Typically, this type of transition is achieved by using α-blending. That is, transparency is used to allow for the perception of different object representations and their respective rendering styles. Besides use of classical illumination models for photorealistic rendering, this provides a transition to various NPR-shadings of a model, such as gray-scale or Gooch shading [GGSC98].

This directly leads to the see-through-effect in α-blending: As one object is rendered in two different styles, these styles visually interact with each other by being visible through their respective transparent counterpart. This effect is well visible in the third and forth image of Figure 3.2 which shows a series of snapshots from an animation created by blending an eye's nerve continuously from a realistic rendering style to

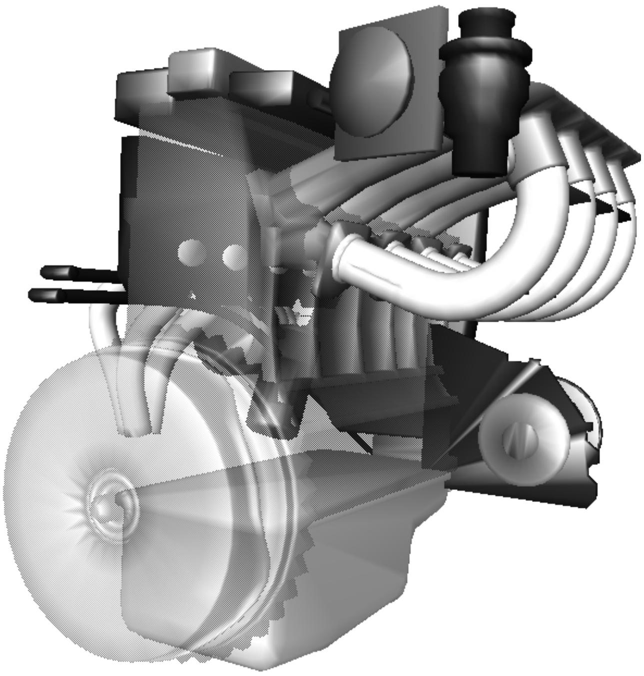


Figure 3.1.: Using semi-transparency for objects in order to be able to see objects behind them. (Courtesy of Henry Sonnet. Used with permission.)

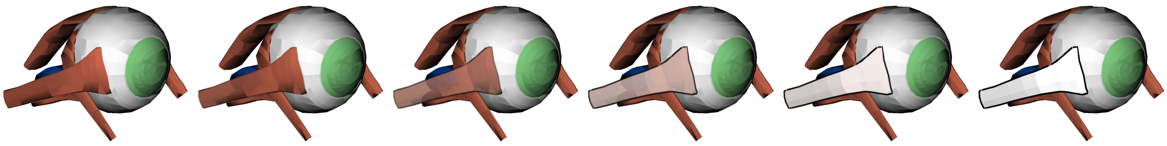


Figure 3.2.: Snapshots from an animation created by blending from a realistic rendition to a non-realistic rendition regarding a part of the model. For this purpose, α-blending is used. Note the *see-through-effect* in the third and forth image causing faces hidden by the blended object to shine through for some period of time during the animation.

silhouette line style by steady adjustments to the α-channel. This form of the see-through-effect is certainly someting that was not intended for the transition from one type of shading to another and should therefore be prevented from occurring. Approaches for avoiding this effect are discussed in Section 5.

4. Using the See-Through-Effect in Stroke Renditions

In the previous section it has already been mentioned that there are quite a few cases where the see-through-effect may be used on purpose, in particular, for visualization applications. In these cases it can be employed to show two layers of information at the same time. Two examples for this will be discussed in detail in the following subsections.

4.1. Illustration of Architectural Information

An exemplary use of a transparency-based see-through-effect in order to illustrate architectural information is presented in [SPM+99]. As illustrated in Figure 4.1, a transparent visualization of a virtual reconstruction of an ancient building over photography of its excarvations showing the foundation walls may be used to provide an impression of the former position and appearance of the building. The different degrees of transparency also give an impression as to the degree of trust into the virtual reconstruction depending on its distance from the ground. In fact, in this image both types of see-through-effect are used: the non-opaque style of the photorealistically rendered virtual reconstruction and the partial coverage style of the line drawing that is shown as well.

4.2. Stippling Transparency

A second example for using the see-through-effect for visualization purposes is animated stippling. Real-Time 3D Stippling [Mer03] is a technique which makes it possible to produce frame-coherent animations of models rendered in the stippling style. As described in [MS04], a novel application of animated stippling is to use stipples to represent a transparent layer in a model. To achieve this effect, stipples are placed over the surface of the 3D model and are rendered in place of the polygonal surface itself.

The use of a screen of stipples on an animated surface makes it possible to visualize both the transparent surface and the parts of the model which lie behind the

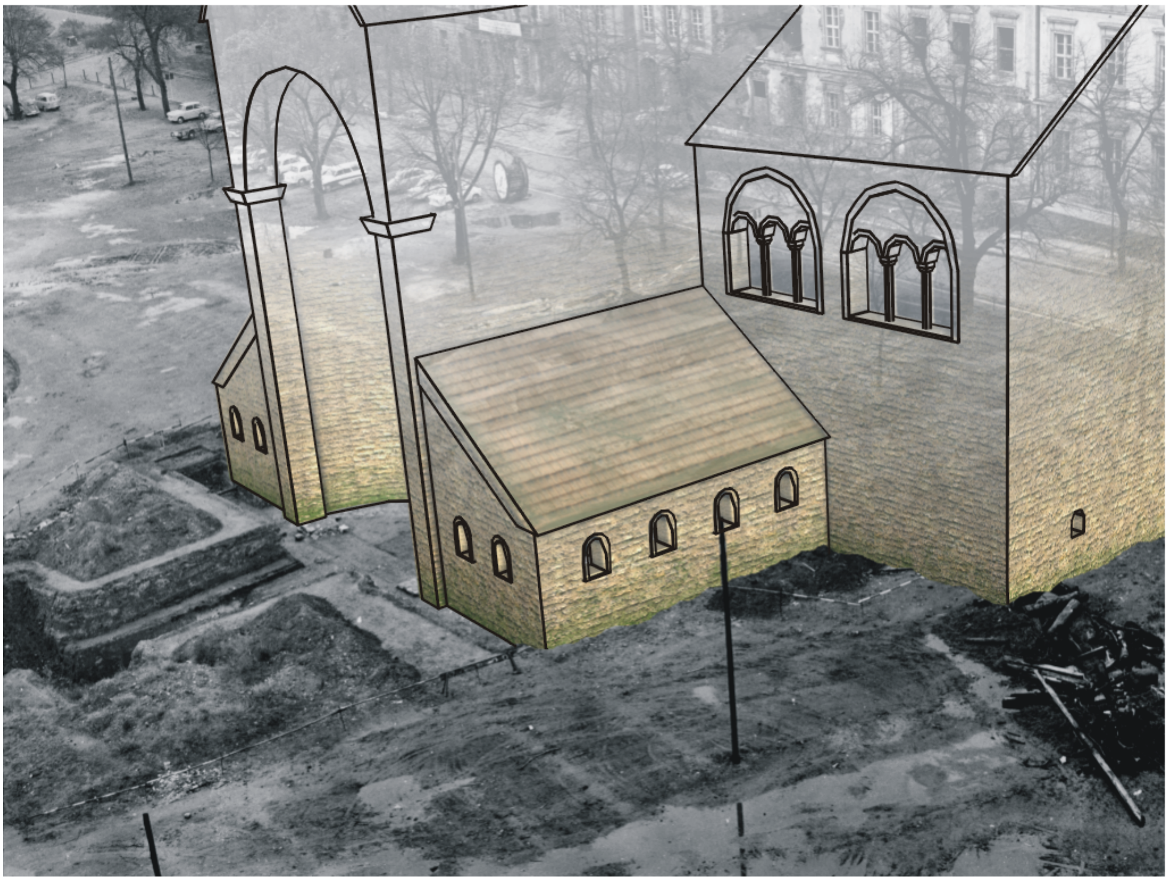


Figure 4.1.: Use of strokes combined with a transparent visualization of a reconstruction over its foundation walls [SPM+99].

stipple set (see Figure 4.2). In addition, for animated surfaces it is possible to visualize the stretching of the surface where the stipples are located. The images in Figure 4.2 illustrate this effect in the case of a heart beat. The heart shell has been replaced with a stippled layer and it is possible to observe the inner animation as well.[1] As the heart beat occurs, it is possible to see how the outer layer stretches and contracts while the inner parts are also animated. The effect is best observed in an animation as opposed to still imaged. This effect cannot be achieved if non-textured semi-transparent surfaces are used rendered with α-blending or a similar approach. An alternative to the use of a stipple cloud is to place a non-photorealistic texture on the semi-transparent surface. In this case, the approaches of Praun et al. [PHWF01] and Freudenberg et al. [FMS02] can be employed. However, a disadvantage of textures is that they are meant to be applied on static polygonal models. Nevertheless, there are a number of models that do not involve stretching surfaces as part of the animation, and which could benefit from use of textures on transparent surfaces. In addition, non-photorealistic transparent surfaces can be illustrated in real-time using

1 The accompanying video was produced using standard 3D animation software and was produced only to illustrate this effect. It does not necessarily follow the actual physics of the heart.

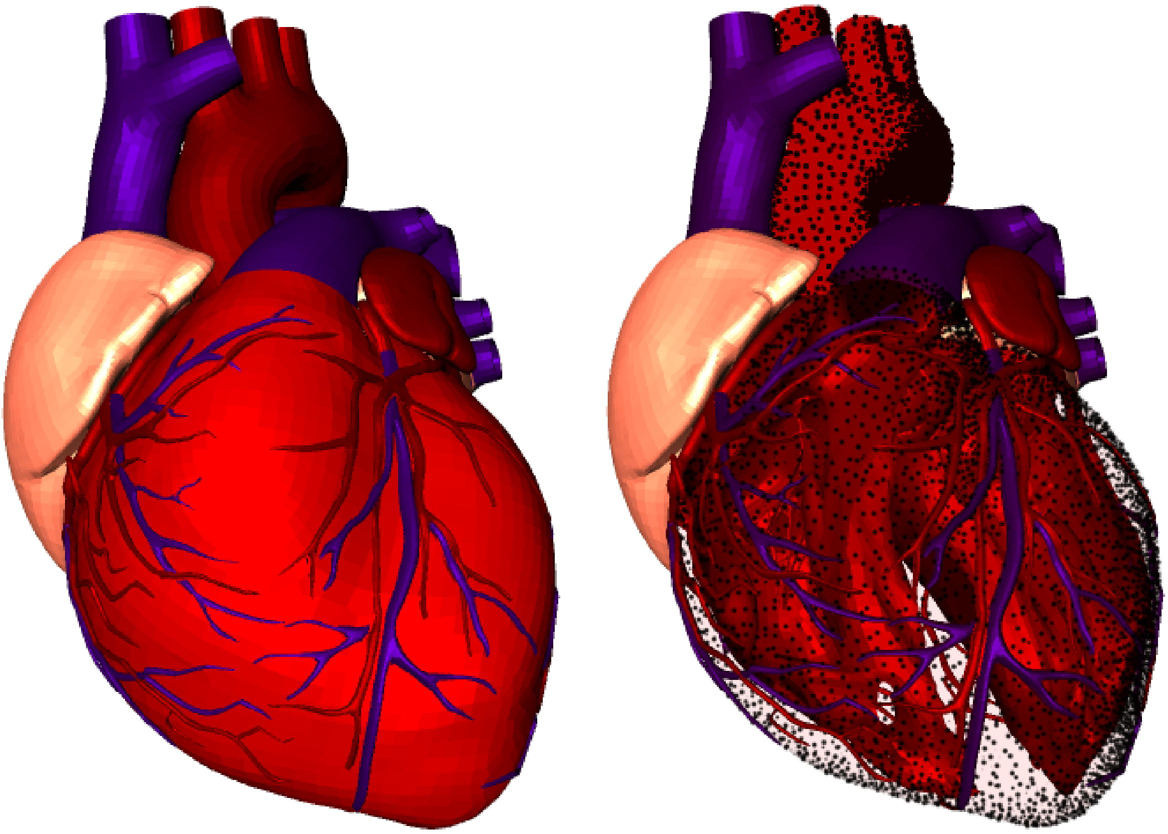


Figure 4.2.: Renderings of a heart model. The images illustrate the effect of providing a view of the inner model parts through transparent stipples. This way, two layers of information are shown. The effect is even more emphasised in case of the animation accompanying this paper.

vertex programs. Both Freudenberg [Fre04] and Meruvia [Mer03] make use of vertex programs to produce real-time non-photorealistic renditions.

This transparency effect has a particular advantage with respect to the standard transparency effect achieved through α-blending: when the model is animated, the stipples on the surface of the transparent layer behave like an elastic texture, showing how this surface changes during the animation. If the inner parts of the model are also animated, it is possible to view the animation of these parts as well.

5. Avoiding the See-Through-Effect in Stroke Renditions

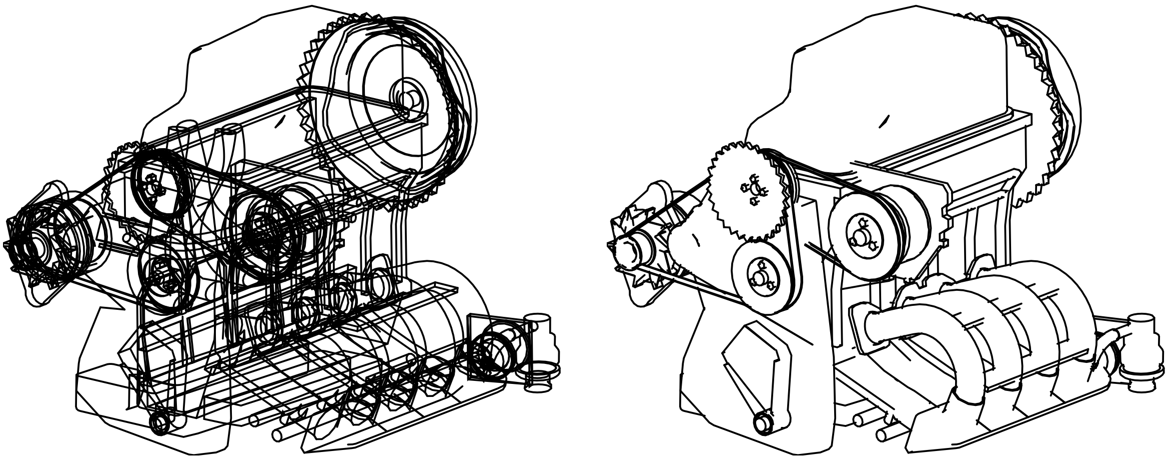
In contrast to using the see-through-effect for, e. g., showing different layers of information as discussed in the previous section, there are also cases where it should be avoided. In particular, when using stroke techniques there are often cases where the effect creates unwanted results. Two examples for this will be discussed in the following.

5.1. Hidden Line Removal

An very common unwanted see-through-effect that is usually not considered to be one occurs in plain silhouette rendering. Silhouettes can be generated by explicitly computing the silhouette edges, e. g., by brute-force testing for edges that share one front-facing and one back-facing polygon. When directly rendering these silhouette edges, the part of the silhouette that is theoretically occluded still shows up in the rendering as expressed in Figure 5.1(a). This is by reason of a common hidden surface removal method such as z-buffering not working for stroke primitives. Thus, some extra type of hidden line removal has to be employed to remove the invisible part of the silhouette strokes as shown in Figure 5.1(b). There are several techniques that can be employed in this context (see, e. g., [IFH$^+$03]).

5.2. Combining Strokes With Shading

More serious problems are caused when it is desired to use a NPR stroke technique as a means of highlighting certain objects of an otherwise protorealistically shaded model (as mentioned in Section 1). In these circumstances, it is often not desired to see objects behind the highlighted object. It is therefore required to add an additional shading to the object visualized using strokes. This way of visualizing objects refers to the drawing-on-paper metaphor that most stroke-based techniques rely on. This results from examples these techniques have in real artwork in that hand-drawn stroke renditions are typically presented on some kind of mostly white paper. Therefore, a similar shading should be employed in hybrid renditions as well in order to avoid



(a) Silhouettes without occlusion culling. (b) Silhouettes with occlusion culling.

Figure 5.1.: See-through-effect in silhouette rendering when no occlusion culling is used.

the see-through-effect shown in the examples of Figure 1.1. In order to be able to generate smooth transitions between a regularly shaded scene and one with selected objects being visualized using a combination of strokes and NPR shading, two methods may be used (for more details on these approaches see [JINS04]).

A naïve approach is to render the NPR shading non-transparently first, followed by an opaque rendering of the realistic shading style. The latter is then continuously made more transparent using α-blending until it is completely blended out. This guarantees a smooth transition from a realistic rendering style to a non-realistic style. Unfortunately, this technique again leads to a different type of see-through-effect caused by α-blending used for the transition as mentioned in Section 3.2 and shown in Figure 3.2.

An alternative to modifying the α-channel of a material is in adjusting the emissive color of objects in a scene. As discussed in [JINS04], this is achieved by manipulating the object's emissive color field. By applying extreme values to this parameter (i. e., either 1 or -1 for all color channels) the object is displayed as completely white or black without affecting the neighboring objects. This way the manipulated objects form a canvas for strokes that are rendered on top of them using the drawing-on-paper metaphor. In addition, this technique can easily be animated to produce a smooth transition between photorealistic and non-photorealistic rendition without the see-though-effect in α-blending (see Figure 5.2).

A series of snapshots illustrating this idea is shown in Figure 5.3 at the example of a foot presentation. Therein, dynamic changes of hybrid rendering styles are used to communicate the spatial relationship of the bones of the toes. Beginning at a cuneiform bone, a silhouette line rendering is used in order to gradually emphasize the individual bones forming a toe and associated bones. Precisely, the order of bones as

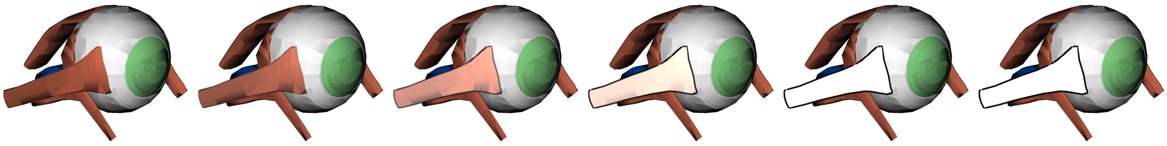


Figure 5.2.: NEW: Snapshots from an animation created by changing the emissive color RGB values of a nerve continually from 0.0 to 1.0. Note that no shading artifacts are visible for this specific nerve and that the *see-through-effect* from Figure 3.2 is avoided.

pointed out in the figure is: *Os cuneiformie I*, *Os metatarsale I*, *Phalanx proximalis I*, and *Phalanx distalis I*.

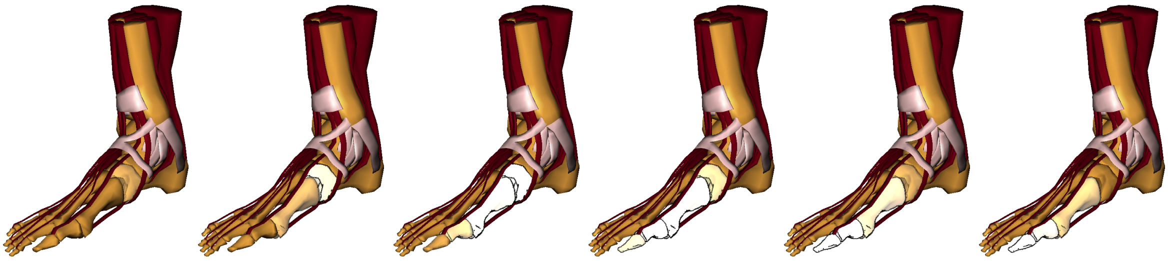


Figure 5.3.: Snapshots of an animation pointing out specific parts of an anatomical example. The bones *Os cuneiformie I*, *Os metatarsale I*, *Phalanx proximalis I*, and *Phalanx distalis I* are emphasized in order to visualize their functionality.

This animation can be run cyclical at varying speeds. Now, a sophisticated learning environment could possibly provide multiple parameterizations of the animation cycle to the students. By spanning the whole spectrum of the presented stimulus intervals and analyzing the learning effect on the students, studies about respectively modified learning curves can be derived.

Figure 5.4 shows a series of snapshots from an animation presenting the utilization of the engine model. The current item of interest is its cooling system. Four tubes combine to this system and are supported by a cooling aggregate. In order to examine the utilization of individual tubes, the consecutive usage of all individual tubes is visually represented with a smooth transition from one tube to the next. The work load of the small extra tube connected to the cooling aggregate is emphasized incrementally while going through the main tubes. A trained design engineer is now in charge of interpreting the scenario in order to decide on a potential lack of capacity for the bandwidth of this connection tube.

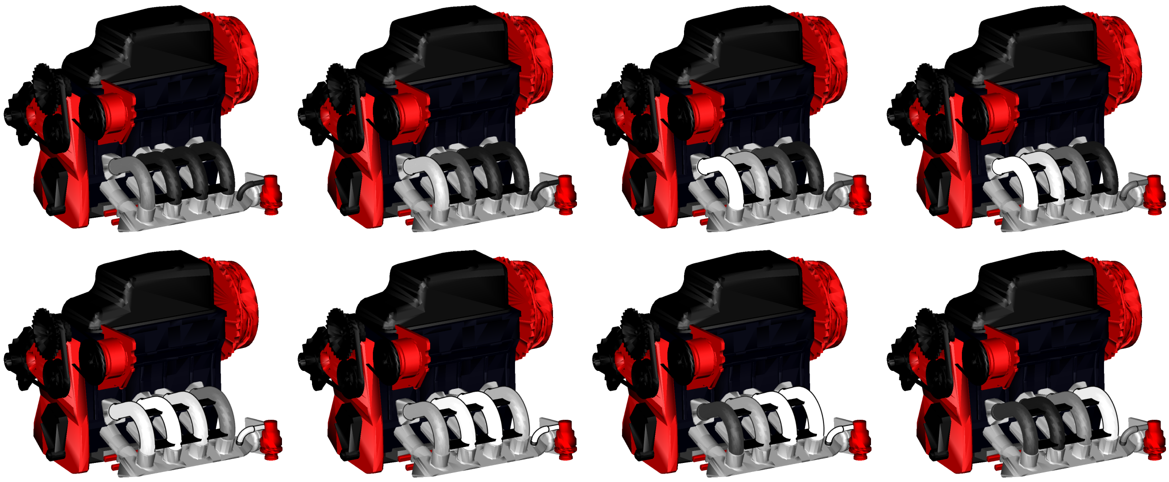


Figure 5.4.: NEW: Snapshots of an engine model. The tubes of the engine's cooling system are gradually emphasized by being rendered in a line shaped silhouette style. The style changes reflect the possible utilization of the single tubes and their interplay with the cooling aggregate located ad the bottom right side.

6. Conclusion

In this paper we have presented the notion of the see-through-effect. We have discussed that it commonly occurs in non-photorealistic rendering when different techniques are combined in order to create hybrid renditions. We have furthermore shown that it is not limited to NPR since it can also be found in, e. g., photorealistic renditions involving fog or smoke. However, most of our discussion focused on the specifics of this effect with regard to non-photorealistic rendering.

We have shown that the effect is caused by using either one of two types of rendering techniques in combination with other rendering styles: partial coverage styles (such as line stippling or hatching) or non-opaque style (as, for example, transparency found in smoke or fog). In both cases a newly rendered object does not cover the items behind it completely. Therefore, these items can be seen through the newly drawn objects.

We have argued that the see-through-effect can either be used intentionally to visualize more than one layer of information at the same time or it has to be avoided where it would lead to distracting renditions. In the subsequent discussion we gave examples for both of these situations. We have shown how transparency and stroke techniques can be used to visualize two layers of information at the same time. In particular, we have shown how stippling can be used in hybrid rendering to show the internal structure of an object and how this effect can be emphasized by employing animation.

In addition, we sketched methods for avoiding the see-through-effect when seeing a second layer of information is not desired. In such cases non-photorealistic stroke styles are used to emphasize certain objects in an otherwise photorealistically shaded scene. In this case a NPR shading relating to the drawing-on-paper metaphor is necessary to avoid seeing the background through the strokes. We have demonstrated how a dynamic transition between photorealistic visualization to the hybrid image can be achieved. We also discussed that this leads to another form of the see-through effect and how this can be avoided as well. Details of the specific algorithms involved are outlined in [JINS04].

Our discussion of the see-through-effect is intended to serve as a starting point for further research. We are well aware that there are many more possibilities for where the effect can either be intentionally used or has to be avoided. Therefore, the examples given in this paper illustrate the existing diversity of relevant situations.

Bibliography

[BSF+94] Eric A. Bier, Maureen C. Stone, Ken Fishkin, William Buxton, and Thomas Baudel. A Taxonomy of See-Through Tools. In *Proc. of CHI'94*, pages 358–364, New York, 1994. ACM Press.

[BSP+93] Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony D. DeRose. Toolglass and Magic Lenses: The See-Through Interface. In *Proc. of SIGGRAPH'93*, pages 73–80, New York, 1993. ACM Press.

[BSP+94] Eric A. Bier, Maureen C. Stone, Ken Pier, Ken Fishkin, Thomas Baudel, Matt Conway, William Buxton, and Tony D. DeRose. Toolglass and Magic Lenses: The See-Through Interface. In *Proc. of CHI'94*, pages 445–446, New York, 1994. ACM Press.

[DWE03] Joachim Diepstraten, Daniel Weiskopf, and Thomas Ertl. Interactive Cutaway Illustrations. *Computer Graphics Forum*, 22(3):523–532, September 2003.

[FMS02] Bert Freudenberg, Maic Masuch, and Thomas Strothotte. Real-Time Halftoning: A Primitive For Non-Photorealistic Shading. In *Proc. Workshop on Rendering Techniques*, pages 227–231. Eurographics, 2002.

[Fre04] Bert Freudenebrg. *Real-Time Stroke-Based Halftoning*. PhD thesis, University of Magdeburg, 2004.

[FSJ01] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. Visual Simulation of Smoke. In *Proc. of SIGGRAPH 2001*, pages 15–22. ACM Press, 2001.

[GDF03] Carl Gutwin, Jeff Dyck, and Chris Fedak. The Effects of Dynamic Transparency on Targeting Performance. In *Proc. Conf. on Graphics Interface*, 2003.

[GG01] Bruce Gooch and Amy Gooch. *Non-Photorealistic Rendering*. A K Peters, Ltd., Natick, 2001.

[GGSC98] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. A Non-Photorealistic Lighting Model for Automatic Technical Illustration. In *Proc. of SIGGRAPH'99*, pages 447–452, New York, 1998. ACM SIGGRAPH.

[GSG+99] Bruce Gooch, Peter-Pike J. Sloan, Amy Gooch, Peter Shirley, and Richard Riesenfeld. Interactive Technical Illustration. In *Symp. on Interactive 3D Graphics*, pages 31–38, New York, 1999. ACM.

[Hod89] Elaine R. S. Hodges, editor. *The Guild Handbook of Scientific Illustration.* Van Nostrand Reinhold, New York, 1989.

[HSS02] Nick Halper, Stefan Schlechtweg, and Thomas Strothotte. Creating Non-Photorealistic Images the Designer's Way. In *Proc. NPAR 2002*, pages 97–104, New York, 2002. ACM Press.

[IFH+03] Tobias Isenberg, Bert Freudenberg, Nick Halper, Stefan Schlechtweg, and Thomas Strothotte. A Developer's Guide to Silhouette Algorithms for Polygonal Models. *IEEE Computer Graphics and Applications*, 23(4):28–37, July/August 2003.

[JI03] Roland Jesse and Tobias Isenberg. Use of Hybrid Rendering Styles for Presentation. In *Poster Proceedings of WSCG 2003*, 2003.

[JINS04] Roland Jesse, Tobias Isenberg, Bernd Nettelbeck, and Thomas Strothotte. Dynamics by Hybrid Combination of Photorealistic and Non-Photorealistic Rendering Styles. Technical Report 5/2004, Department of Computer Science, University of Magdeburg, Germany, 2004.

[Joh02] Scott F. Johnston. Lumo: Illumination for Cel Animation. In *Proc. NPAR 2002*, pages 45–52, New York, 2002. ACM Press.

[Mer03] Oscar E. Meruvia. *Frame-Coherent 3D Stippling for Non-Photorealistic Computer Graphics.* PhD thesis, Univ. of Magdeburg, 2003.

[MS01] Maic Masuch and Thomas Strothotte, editors. *Virtuelle Zeitreise: Der Computervisualistikraum in der Ausstellung "Otto der Große, Magdeburg und Europa".* Department of Simulation and Graphics, Univ. of Magdeburg, 2001. ISBN: 3-9804874-1-5.

[MS04] Oscar E. Meruvia and Thomas Strothotte. Graph-Based Point Relaxation for 3D Stippling. In *Submitted to Eurographics 2004*, 2004.

[PHWF01] Emil Praun, Hugues Hoppe, Matthew Webb, and Adam Finkelstein. Real-Time Hatching. In *SIGGGRAPH 2001 Conference Proceedings*, pages 581–586. ACM Press, 2001.

[RSHS03] Felix Ritter, Henry Sonnet, Knut Hartmann, and Thomas Strothotte. Illustrative Shadows: Integrating 3D and 2D Information Displays. In *Proc. IUI 2003*, pages 166–173, New York, 2003. ACM.

[SPM+99] Thomas Strothotte, Matthias Puhle, Maic Masuch, Bert Freudenberg, Sebastian Kreiker, and Babette Ludowici. Visualizing Uncertainty in Virtual Reconstructions. In *Proc. of Electronic Imaging & the Visual Arts, EVA Europe '99*, page 16, Berlin, 1999. VASARI, GFaI.

[SS02] Thomas Strothotte and Stefan Schlechtweg. *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation*. Morgan Kaufmann, San Francisco, 2002.

[ST90] Takafumi Saito and Tokiichiro Takahashi. Comprehensible Rendering of 3-D Shapes. In *Proc. of SIGGRAPH'90*, pages 197–206, New York, 1990. ACM Press.

[VCWP96] J. Viega, M. J. Conway, G. Williams, and R. Pausch. 3D Magic Lenses. In *Proc. of ACM Symp. on User Interface Software and Technology*, pages 51–58, New York, 1996. ACM Press.

[ZBM96] Shumi Zhai, William Buxton, and Paul Milgram. The Partial-Occlusion Effect: Utilizing Semitransparency in 3D Human-Computer Interaction. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 3(3):254–284, September 1996.