media/rep/

Repositorium für die Medienwissenschaft

DFG Deutsche Forschungsgemeinschaft

Philipps Universität Marburg

Tobias Isenberg

A Survey of Image-Morphologic Primitives in Non-Photorealistic Rendering

Abstract

This paper presents an overview of the image-morphologic primitives used commonly in non-photorealistic rendering (NPR), a subdomain of computer graphics that is inspired by a long tradition of artistic and illustrative depiction. In particular, we survey NPR shading, stroke-based rendering, sparse line drawings, graftals, and area primitives. Such primitives usually cover larger regions on the canvas and often carry a meaning beyond the color of the image region they represent. This distinguishes them from the pixel as a primitive used in photorealistic rendering, which does not have any meaning aside from sampling the color of the image section it represents. We give examples to illustrate the individual techniques and briefly mention how they are tracked though the rendering process as well as represented in the final image.

1   Introduction

For many years one main goal of computer graphics research has been to depict reality as it can be captured by photography. This goal of creating *photorealism* has received a lot of research attention in the areas of computer games and in the film industry. In the last decade or two, a different area of research has been established within computer graphics that does not share this goal. Instead, it is inspired by a long tradition of artistic and illustrative depiction and tries to break free from the constraints that are set by photorealistic rendering. This new area, *Non-Photorealistic Rendering* (NPR), has produced a wealth of techniques that allow us to simulate many forms of traditional media. For example, techniques such as oil painting, watercolor, pen-and-ink, stippling, or comic shading can now be reproduced fully automatically or using partial computer support; also entirely new techniques have been conceived. A good overview of the area give [Gooch & Gooch 2001] and [Strothotte & Schlechtweg 2002].

The new domain greatly benefits from the same freedoms that exist in traditional artistic and illustrative depiction such as the possibilities for abstraction, exaggeration, choice of

view and projection, etc. Similarly, choices exist with respect to the selection of a tool and medium since NPR tries to emulate traditional means of depiction just as closely as photorealism tries to simulate the camera.[1] Tools and mediums in this context are, for example, brushes and watercolor for watercolor painting or copper plates and ink for pen-and-ink techniques. In a way one could also see optics as the particular and single tool of depiction in photorealism while in NPR there exists a variety of tools that can be chosen. On the one hand, this focus on tools for many NPR techniques results in creating marks that are evident in the produced images just like with traditional tools. On the other hand, these marks are also represented as primitives in the picture production process. Therefore, marks used in NPR can and usually do carry a meaning.

This constitutes a major shift from photorealistic rendering where images are rendered typically on a pixel-by-pixel basis. In photorealistic rendering, the triangles in a 3D model are traversed, and each triangle is rasterized into pixels, for which individual lighting and texturing computations are performed, and the pixel is finally stored into a buffer. In raytracing, the pixel is even more prevalent as rays are shot into the scene based on a predetermined pixel raster. This concentration on a pixel raster and the pixel as output primitive is somewhat arbitrary: the pixel raster is only determined by the overall pixel size of the output image and its resolution; it does not depend on the contents of the image itself. Pixels, therefore, do not carry a meaning beyond the color of the image section they represent.

In non-photorealistic rendering, in contrast, higher-level primitives are typically used to represent the depicted objects and scenes, even if the final image is rasterized and stored as a pixel matrix.[2] In contrast to pixels, higher-level primitives usually have a meaning beyond the essentially arbitrary measure of resolution. They normally represent the marks created by the traditional tools that are simulated in NPR. As such, NPR primitives can represent lighting conditions, similar to what pixels represent in photorealism. In contrast to photorealistic pixels, however, NPR primitives also represent properties of the depicted materials and objects. This is the case, for example, in hatching and stippling. Non-photorealistic rendering also allows us to go beyond the mere simulation of traditional techniques and make use of dynamic primitives such as *graftals* [Kowalski *et al*. 1999] that can adapt their way of rendering depending on conditions such as view and size on the screen.

We analyze the different types of image-morphologic primitives in non-photorealistic rendering in Section 2. We then explore how such primitives are typically tracked during the rendering process (Section 3), and how they are finally represented as either pixel images or vector graphics (Section 4). The final Section 5 concludes the paper.

---

[1] This could be seen as the quest for (photo)realism in NPR.
[2] This, however, does not have to be the case as NPR images can also be stored as vector graphics, depending on the rendering process.
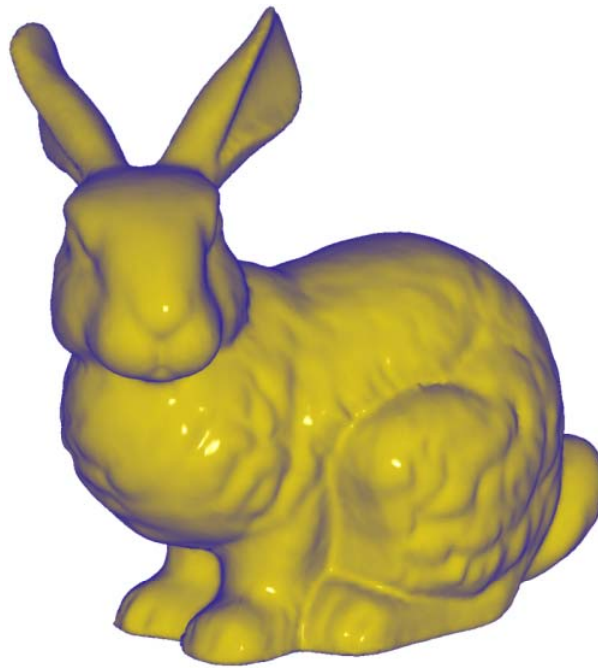
Figure 1: NPR cool-to-warm shading [Gooch et al*. 1998] can be used to better depict the surface shape. Cool colors (e.g., blue) are used in darker regions, warm colors (e.g., yellow) in lit parts*

2   Image-Morphologic Primitives in NPR

The different types of image-morphologic primitives that are being used in non-photorealistic rendering and imagery range from the pixel as in photorealistic rendering to fairly large elements such as silhouettes and feature strokes and even includes dynamic elements such as graftals. The following sections discuss these different types of primitives, grouped roughly by their size and their purpose in representing elements in the images.

2.1   Non-Photorealistic Shading

One subset of non-photorealistic rendering techniques adapts photorealistic rendering only slightly to create, for example, images that are more illustrative. This can be achieved, e.g., by using nonphotorealistic illumination models such as *Gooch cool-to-warm shading* [Gooch *et al*. 1998] (Figure 1). The goal of this technique is to introduce more richness into the transition from illuminated regions to dark regions, and to better suggest shape, a technique illustrators have been using for a long time. This is achieved by mixing the object's color properties with an additional transition from warm to cool colors, which makes color changes better visible in very bright regions as well as in very dark regions. As the traditional photorealistic rendering pipeline does not need to be changed much—only the illumination formula is modified—the same image-morphologic element is used: the pixel. As in photorealistic rendering, the pixel is employed as a means to sample the entire area of the image to be able to display and store it on digital media. Therefore, the pixels in

Figure 2: Application of Haeberli's [1990] Paint by Numbers technique to Figure 1 as the source image

these techniques are equivalent to pixels in photorealistic rendering: they do not carry any meaning aside from sampling the color of the image depending on the image size and resolution.

The vast majority of non-photorealistic rendering methods, however, are based on higher-level image-morphologic primitives such as strokes, mosaic tiles, graftals, area primitives, etc. This is partly the defined result of using abstraction, but mostly it is due to the goal of trying to simulate traditional techniques of artistic expression in which specific tools such as paintbrushes are used. These tools leave marks on the created images, giving the created images their unique character and style. Even though some techniques are implemented using pixel-by-pixel processing, these higher-level primitives are still evident in the produced images. One could argue that the tool in photorealistic rendering and non-photorealistic shading is the simulation of optical processes on a pixel-by-pixel basis resulting in no evidence of marks produced by this tool other than the pixels themselves. Therefore, the pixel as the primitive in non-photorealistic shading is the lowest-level primitive used in NPR.

2.2   Stroke-Based Rendering

A range of techniques that was very attractive for NPR researchers to attempt to replicate is the painting or drawing with strokes (e.g., brush strokes in oil and watercolor painting). The range of strokes being simulated includes, for example, pencil or ink strokes, stippling, and even the placement of decorative mosaic tiles. In their original use these marks can

represent elements of the depicted scene (e.g., a brush stroke in painting could represent a leaf or a group of leaves in a tree) or just serve as a means to sample the scene for the canvas (e.g., pointillism). Marks in some techniques are used to work around limitations of the chosen medium. For example, stippling and hatching are employed due to the difficulty of using gray scales in the printing process.[3]

The area within NPR to simulate such techniques is called *stroke-based Rendering* [Hertzmann 2003]. The idea is to compute a new image based on an example image or 3D scene by rendering shorter or longer strokes. These strokes both approximate the example image as well as abstract from it at the same time. The degree of this approximation and abstraction depends on the specific types of marks, their size, and how many marks are being used.

Example techniques in this category include painterly rendering (e.g., [Meier 1996; Hertzmann 1998]), pointillism (e.g., [Yang & Yang 2006]), stippling (e.g., [Deussen *et al*. 2000; Secord 2002; Schlechtweg *et al*. 2005]), hatching (e.g., [Salisbury *et al*. 1994; Winkenbach & Salesin 1994; Salisbury *et al*. 1996; Winkenbach & Salesin 1996; Salisbury *et al*. 1997; Deussen *et al*. 1999; Ostromoukhov 1999; Hertzmann & Zorin 2000; Praun *et al.* 2001; Zander *et al*. 2004]), and the rendering of decorative mosaics (e.g., [Hausner 2001; Elber & Wolberg 2003; Di Blasi & Gallo 2005]).

One of the earliest and most influential approaches in stroke-based rendering was Haeberli's Paint by Numbers technique [1990].[4] His system introduced the principle of non-photorealistic abstraction of an image by placing strokes onto the target image using the color sampled from the source image. This approach did not yet have the target of closely simulating a specific traditional style, but used strokes as drawing primitives that are evident in the produced image (Figure 2), thus opening up a multitude of possibilities for stroke-based rendering in simulating traditional styles as well as coming up with new ones.

The evidence of mark making in traditional depiction techniques could be seen as an artifact of using the tool. However, artists and illustrators have developed ways to place marks that represent more than just that. This is very obvious, for example, in hatching as employed in woodcuts and copperplate engravings. Here, the hatching lines not only represent the gray value of an equivalent black-and-white photograph but also portray the structure and properties of the depicted surfaces. A masterly example for this application of marks is shown in Figure 3.

---

[3] Normally in the printing process, halftoning techniques are often employed which use dot patterns of pure colors (spot colors) to represent shades of gray or color.

[4] See http://laminadesign.com/explore/impression/imphelp.html for an example implementation of Haeberli's Paint by Numbers technique [1990].

Figure 3: Example off an artist's use of hatching to portray the structure and properties of the depicted surfaces (e.g., cloths and basket). Detail from the woodcut "Life of the Virgin: 14. The Rest during the Flight to Egypt" (1504–05) by Albrecht Dürer

NPR techniques simulating these pen-and-ink styles[5] have attempted to replicate this effect. For example, Salisbury *et al*. [1994] as well as Winkenbach & Salesin [1994] used specific *prioritized stroke textures* to represent different materials. These consist of pre-recorded layers of strokes for representing a series of consecutively darker textures and are applied according to the gray value in a source image or the lighting conditions in a 3D scene. Other approaches put more emphasis on extracting a field of streamlines from a 3D surface to be able to illustrate the surface's shape (e.g., [Hertzmann & Zorin 2000] and [Zander *et al.* 2004]). These techniques portray illumination using line densities or by modulating the line parameters such as thickness or line stippling patterns (Figure 4).

In most pen-and-ink styles, strokes are used for showing the structure of surfaces by depicting ridges, creases, and other surface features (as done, for instance, by Winkenbach

---

[5] Although woodcuts and engravings are not technically pen-and-ink styles, NPR usually includes them in this group.
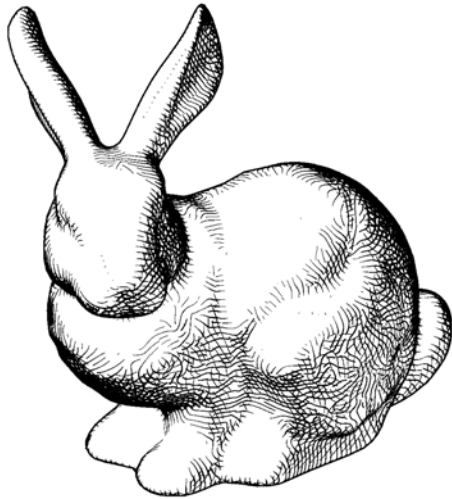
Figure 4: Hatching using Zander et al.'s technique [2004]

Figure 5: Painterly rendering: strokes represent leaves or wood shingles. Courtesy and copyright of Martin Schwarz, used with permission

and Salesin [1994]). In painterly rendering, on the other hand, strokes are typically employed to either represent whole elements of the depicted scene such as the leaves or wood shingles in Figure 5 or to convey the overall impression of painterly rendering.

In the latter case strokes are not associated with a particular object or scene element that they portray. In general, it can be difficult to algorithmically associate the placement of marks with elements of the depicted scene in a meaningful way as this assumes an understanding of the depicted scene. This can, however, be supported by allowing more interaction with the NPR rendering technique to guide the placement of marks in meaningful ways.
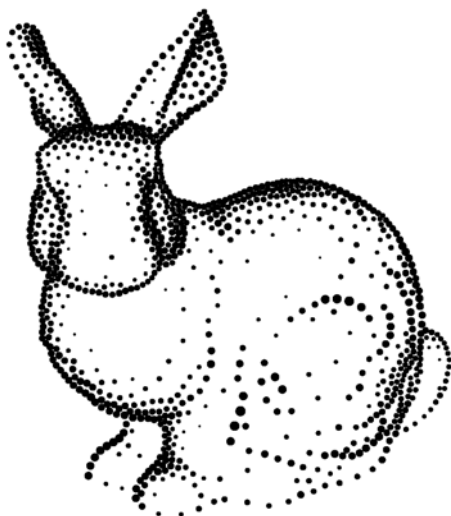
Figure 6: Stippling using Secord's technique [2002]
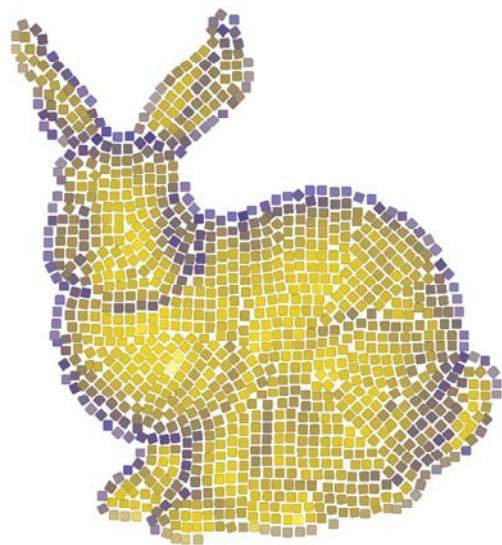
Figure 7: Decorative mosaics using Schlechtweg et al.'s Render-Bots [2005]

(a) Only silhouettes   *(b) With suggestive contours*

Figure 8: Comparison of a sparse line drawing with just silhouettes and with additional suggestive contours [DeCarlo et al. 2003]

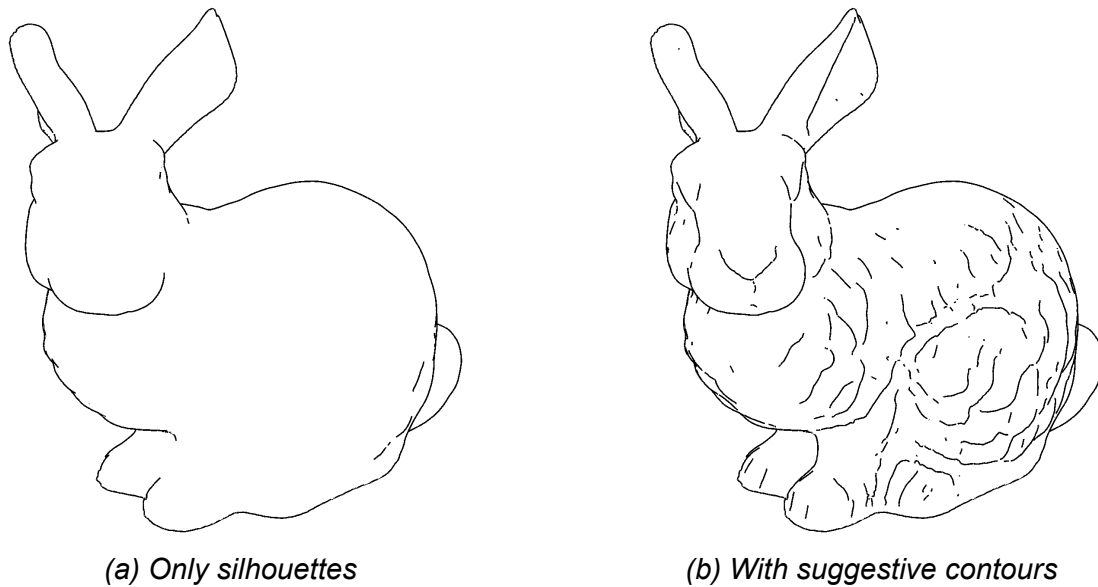In some stroke-based rendering techniques such as stippling and (usually) the simulation of traditional mosaics, the primitives do not represent meaningful elements of the image. Instead, the marks are used to carry shading information (stippling, Figure 6) or colors (mosaic tiles, Figure 7) and are the—intended—artifacts of the specific technique.

2.3 Strokes in Sparse Line Drawings

Some meaningful structures can be algorithmically extracted from, in particular, 3D scenes and constitute exceptions to the above-mentioned general rule. These are *silhouettes* and *feature lines*; elements that make up *sparse line drawings* (abstract illustrations with just a few significant lines). Such lines are a very common means of expression and are traditionally used, for example, in comics and technical drawings. They are also often used in conjunction with the previously discussed stroke-based rendering techniques to guide stroke placement or as additional elements in the images.

Silhouettes are lines on the surface of 3D objects where the visibility changes from visible to invisible, or *vice versa* [Isenberg *et al.* 2003]. As such, silhouettes are view-dependent and move on the surface as a view onto the object changes. For closed objects, the silhouette also comprises a curve that borders the object and separates it from the background: its *contour*. Feature lines consist of lines on the surface that are otherwise significant and should be drawn in a sparse line drawing. The latter group includes view-independent creases, i.e., lines of sharp bends or high local curvature of the surface, but also view-dependent lines such as *suggestive contours* [DeCarlo *et al.* 2003] that visually extend the silhouettes in an image (Figure 8).

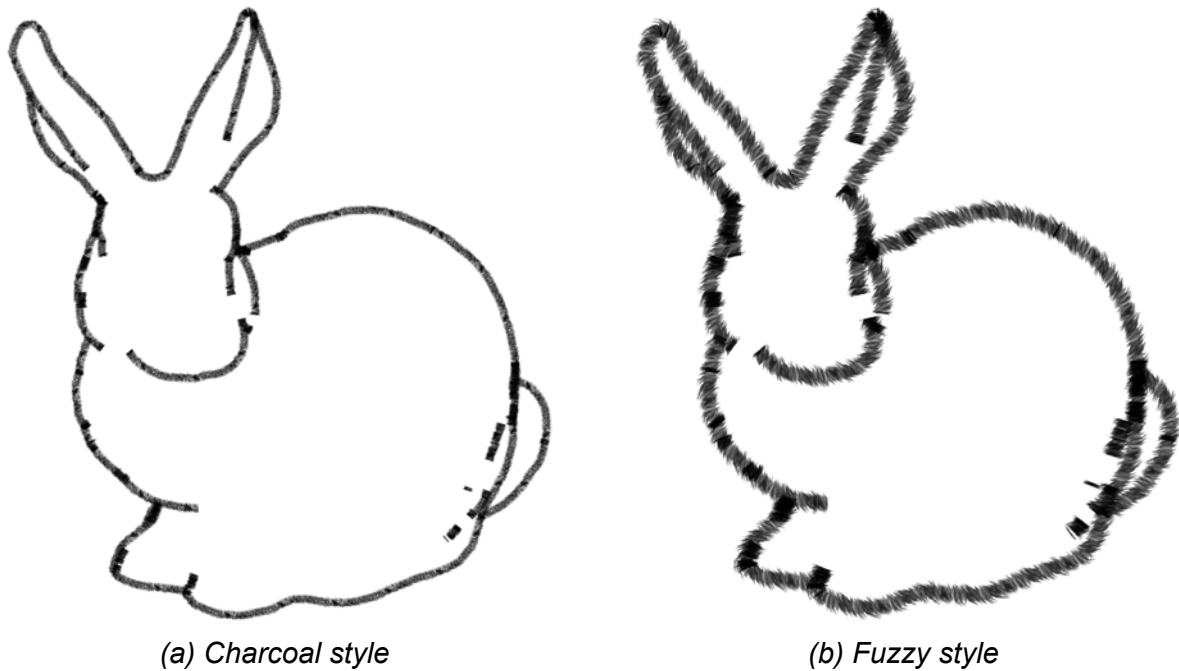*(a) Charcoal style*   *(b) Fuzzy style*

Figure 9: Applying line styles in form of textures to extracted silhouette strokes

After having been extracted from a 3D scene, silhouette and feature line segments are concatenated to form longer strokes. This character of forming long strokes solely as meaningful elements of the image distinguishes the lines in sparse line drawings from the strokes used in stroke-based rendering where strokes are also and probably mainly a means of sampling the image space, i.e., are used for portraying color and/or shading.

The silhouette or feature strokes can now either be drawn directly or be modified by applying a *line style*. The latter can simulate a specific traditional drawing tool such as a pencil, watercolor, chalk, or charcoal by applying an appropriate texture (Figure 9). Line styles can also disturb the path of a stroke, for example, to simulate the appearance of sketchiness.

2.4   Graftals

Graftals are a special form of stroke used in non-photorealistic rendering. Introduced to computer graphics by Smith [1984], in NPR the term has developed to comprise primitives that can algorithmically change their visual representation depending on parameters that can vary over the course of an animation or simulation [Kowalski *et al.* 1999; Kaplan *et al.* 2000; Markosian *et al.* 2000]. Similar to some elements in stroke-based rendering, they represent meaningful elements in a scene. In contrast to stroke-based rendering, however, the dynamic and procedural character allows graftals to change the visual representation depending on, e.g., the orientation or distance to the viewer (Figure 10). For example, a tuft of grass in the background may be shown with just one or two black strokes on a
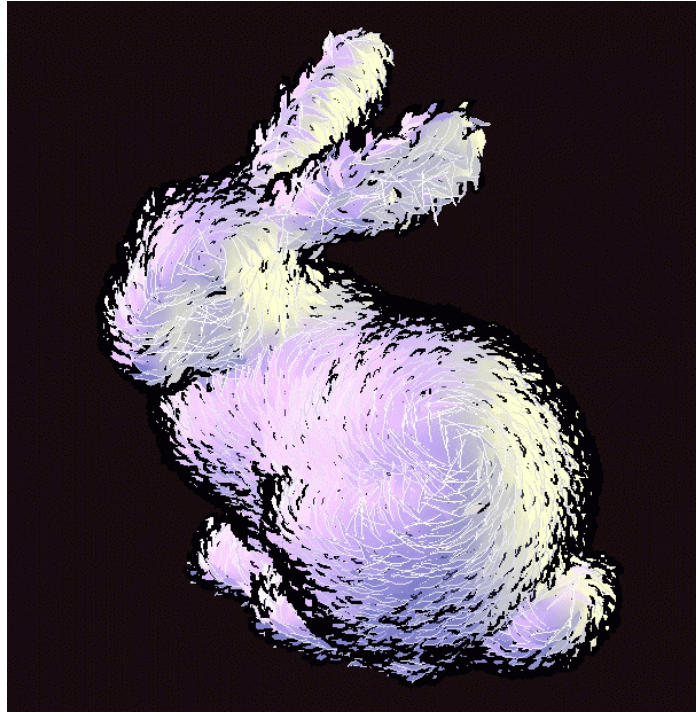
Figure 10: Graftals used to simulate artistic fur [Kaplan et al. 2000]. Note the different appearance of the graftals facing the viewer and those on surface parts that are perpendicular to the viewing direction. Image courtesy and copyright of Matthew Kaplan, used with permission

green background. As the camera gets closer, more strokes will appear and eventually a bush of triangular leaves will become visible.

Initially, graftals were used in an image-space stroke-based rendering manner [Kowalski *et al*. 1999]. This, however, leads to frame-incoherence and, thus, to flickering images in an animation because of the frame-by-frame processing and each frame being treated independently. In other approaches, graftals are placed into the scene during the modeling phase, directly associating them to the specific surfaces or objects they represent [Kaplan et al. 2000; Markosian *et al*. 2000]. This way their locations remained fixed on the respective surfaces, which makes the rendering coherent over time. In a way, this contrast between frame-incoherence vs. the maintenance of primitives over time constitutes a temporal equivalent of the difference between (very local) pixel processing and higher-level primitives.

2.5 Area Primitives and Patterns

One final group of primitives that are used in non-photorealistic rendering is not primarily based on short strokes, long strokes, or procedural graftals. We call this group *area primitives and patterns* because they cover larger areas of the produced images, sometimes filling them with patterns. NPR techniques that focus on the use of real primitives simulate, for example, traditional ornaments, modern art styles, and cell animation. As they do not
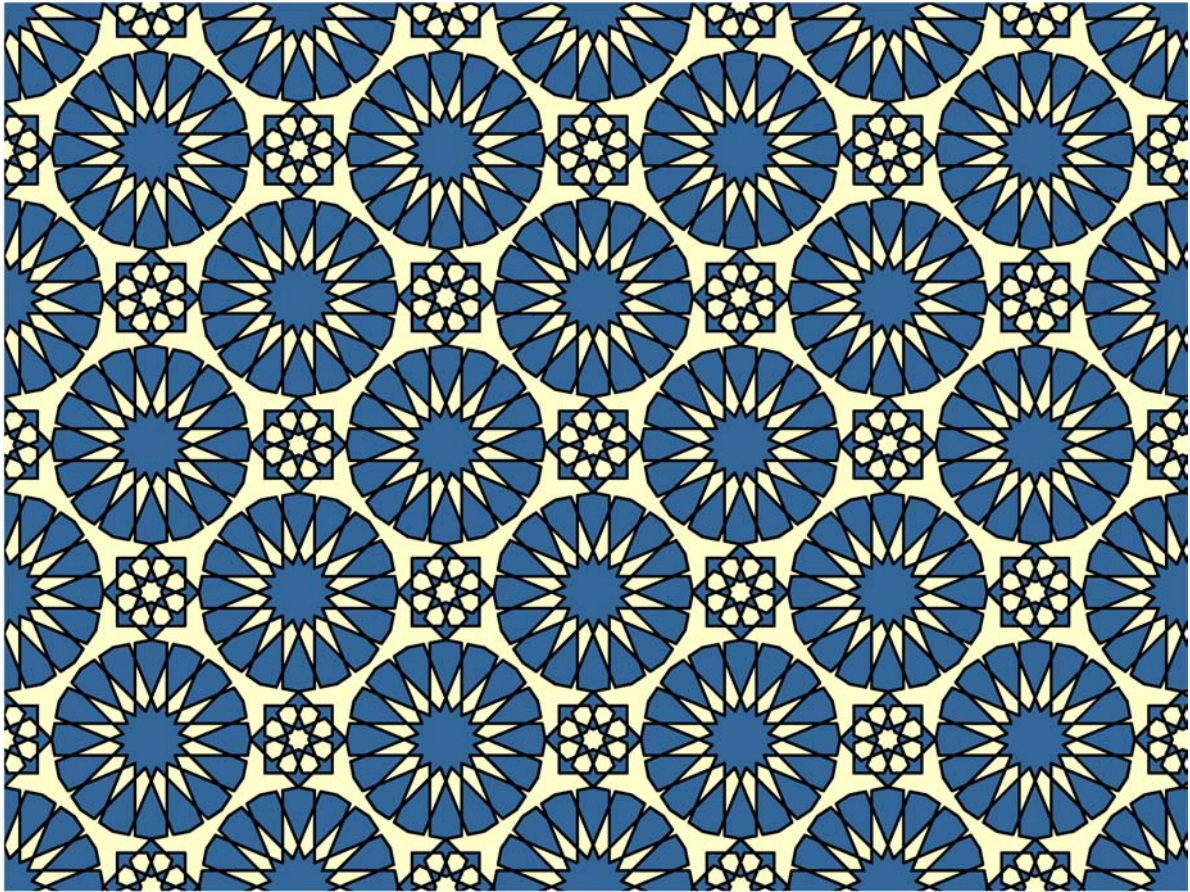
Figure 11: Oriental pattern generated using the method by Kaplan & Salesin [2004]

necessarily have a common algorithmic background, we restrict ourselves here to a few examples. Ornaments are algorithmically produced in form of floral, oriental, and Celtic patterns as inspired by the traditional examples [Ostromoukhov 1998; Wong *et al*. 1998; Kaplan & Cohen 2003; Kaplan & Salesin 2004], usually based on a mathematic scheme underlying the ornament (Figure 11). Related approaches reproduce ornamental effects such as the ones demonstrated by the works of M. C. Escher where the plane is entirely filled by repeating tiles of a given input image [Kaplan & Salesin 2000] or where two tiles are used which are morphed into one another [Ostromoukhov & Hersch 1995]. Other techniques are inspired by works by Piet Mondrian or the Japanese Seigaiha style and employ multi-agent systems and coalition forming to generate the elements in an image such as lines, colored tiles, and other patterns [Mason *et al.* 2005].

Cel shading can be thought of as a special case of area primitives. While it is technically an NPR shading technique (as discussed in Section 2.1), it also generates area features that can be regarded as non-photorealistic image-morphologic primitives. Here, the typical Phong shading technique of surfaces is changed such that regions with solid colors are created (Figure 12). This is inspired by traditional cel animation where foreground figures were drawn using silhouettes and feature lines on celluloid, and the regions then filled-in with color; finally the figure was recorded on a background. In NPR, the effect of a few
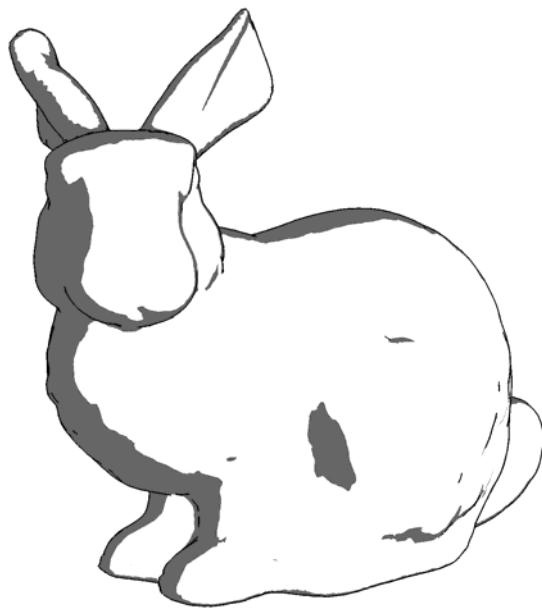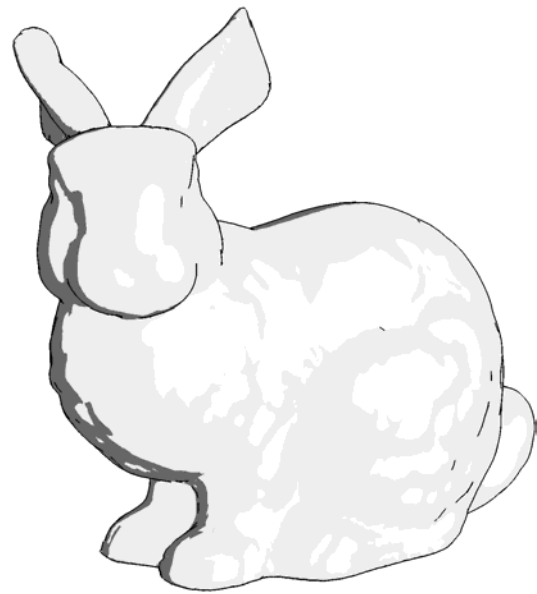
(a) Two colors   (b) Three colors

Figure 12: Cel shading with two or three color steps, combined with silhouettes

shading levels (e.g., shadow, regular colors, and highlights) can be created by defining thresholds for the illumination of a surface, and then coloring all points that lie between two thresholds with the same color. There are also techniques to track highlights specifically, and to give them distinct shapes [Anjyo & Hiramitsu 2003].

3   Tracking Image-Morphologic Primitives

While the algorithms to create the primitives described above are as plentiful as there are primitives and NPR effects, there are generally three distinct categories they can be attributed to. These are image-space, hybrid, and object-space techniques. Depending on the category an algorithm belongs to, the generated primitives are explicitly represented during the rendering process or not. Even if primitives are not explicitly represented in the rendering process they may still be present in the final image.

We briefly discuss the three groups here using the example of silhouette and feature line extraction (for more detail see [Isenberg *et al*. 2003]). Image-space or pixel-based silhouette extraction depends on additional G-buffers[6] storing depth (z-buffer) and/or normal vector[7] information. This data can be processed using an edge detection filter [Saito & Takahashi 1990], resulting in purely local edge elements being detected where discontinuities of the depths or surface orientations are in these G-buffers, i.e., at silhouettes and fea-

---

[6] G-buffers are pixel images of the same size as the rendered image. They are generated during the rendering process and store geometric information about the rendered scene such as local depth values, normal vectors, ID values of triangles or objects, etc.

[7] The normal vector of a point on the surface of an object is perpendicular to the tangential plane that touches the surface point.

ture lines. Thus, even though during the process no silhouette or feature line is explicitly represented, they are nevertheless created through the local pixel processing.

Hybrid techniques do not rely on a pixel-by-pixel processing directly but arrange the actual rendering in a smart way so that silhouettes are created. For example, they select polygons on the backside of an object first, enlarge them, and render them in black. Afterwards, the polygons on the front side are rendered normally; but at the silhouettes the previously rendered black polygons stick out, forming silhouette lines. Again, the silhouettes are not explicitly represented but exist in the final image.

In contrast, object-space techniques do explicitly extract and represent the intended primitives in the rendering process. In the case of silhouette extraction, visibility information is determined, for example, on a polygonal mesh, and line segments are then identified where this visibility changes (from facing a viewer/camera to facing away, or *vice versa*). These line segments are concatenated to form long strokes based on connectivity information from the mesh and can then be rendered.

As a second example, the cel shading technique discussed in Section 2.5 could be thought of as an image-space technique (because it applies a threshold to the computed pixel colors) or a hybrid method (because it changes the lighting computation during rendering). Nevertheless, the image-morphologic primitives generated by this technique— regions with a constant color—only become apparent in the final image and are not present during the rendering process. However, cel shading may be also created in an object-space manner as recently demonstrated by Stroila *et al.* [2008]. Here, the curves bordering the regions of uniform color are explicitly extracted from the scene, and shapes representing the regions are created from these borders.

The characteristic of primitives that they are explicitly represented in the rendering process is essential where the primitives have to undergo further processing. Primitives that are only present in the resulting image in form of pixel colors usually cannot be altered on a meaningful primitive-by-primitive level; only methods such as image-processing filters that work on a pixel-by-pixel basis can be applied. Therefore, in particular in domains such as sparse line drawings where image elements need to be stylized, object-space techniques are used to extract the primitives. Several methods have been created to aid this stylization process through capturing and maintaining additional properties needed for it.

Grabli *et al.* [2004] capture information such as extracted strokes, their type, their visibility, and a number of other data items in a graph data structure called *view map* (Figure 13(a)) that lets them algorithmically determine which strokes to select, chain, and stylize in a wide variety of ways. These complex styles can be stored in style sheets in order to enable easy re-using. This approach allows them to create more complex and elaborate stylized
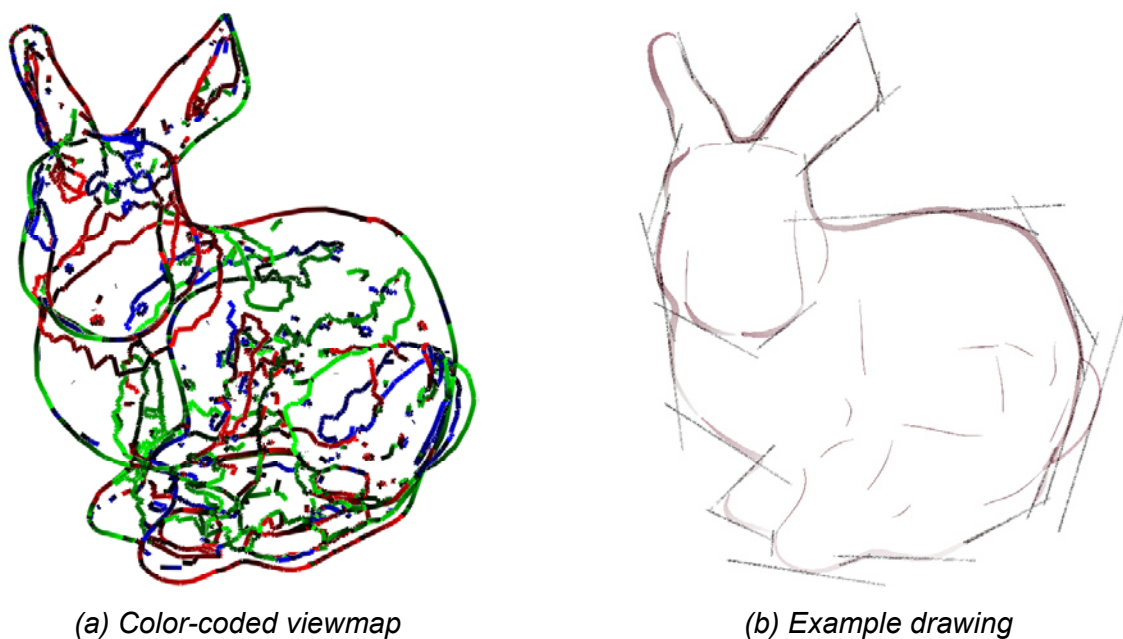
(a) Color-coded viewmap   *(b) Example drawing*

Figure 13: Images generated with the system by Grabli et al. *[2004]*

line drawings as still images (Figure 13(b)). Isenberg & Brennecke [2006] introduced their *G-strokes* approach, which also captures stroke properties but stores them as information tracks parallel to the stroke's geometry data. As the geometric stroke data is processed in the rendering pipeline, so is the additional G-stroke data. For example, when a stroke's visibility is determined, a segment may be found that needs to be split because the visibility changes along its path. In that case an additional visibility G-stroke can be used to capture the visibility information. This new G-stroke and all others G-strokes are adapted during rendering to reflect the necessary changes. In the case of the visibility change this requires splitting the geometry of the segment as well as all its other G-stroke data. This way it is possible to create complex stroke pipeline networks to stylize sparse line drawing at interactive rates (Figure 14).

4   Image Representation

The distinction of whether NPR rendering primitives are explicitly represented in the rendering process or not also plays a role in how the produced images are output and stored. This can occur in one of two forms: as *pixel images* or as *vector graphics*. The type of image determines whether primitives can be explicitly represented in the image as well.

Most commonly, images are stored in pixel raster form. This means that for a given size and resolution a raster is mapped onto the image, and each pixel samples a color, which is then stored in the image. The rendering process itself may already determine this raster, for instance when shading techniques are used. Because of the pre-defined sampling on the pixel raster, and as already noted in Sections 1 and 2.1, pixel images do not represent NPR primitives explicitly. Therefore, it is difficult to identify them algorithmically. However,
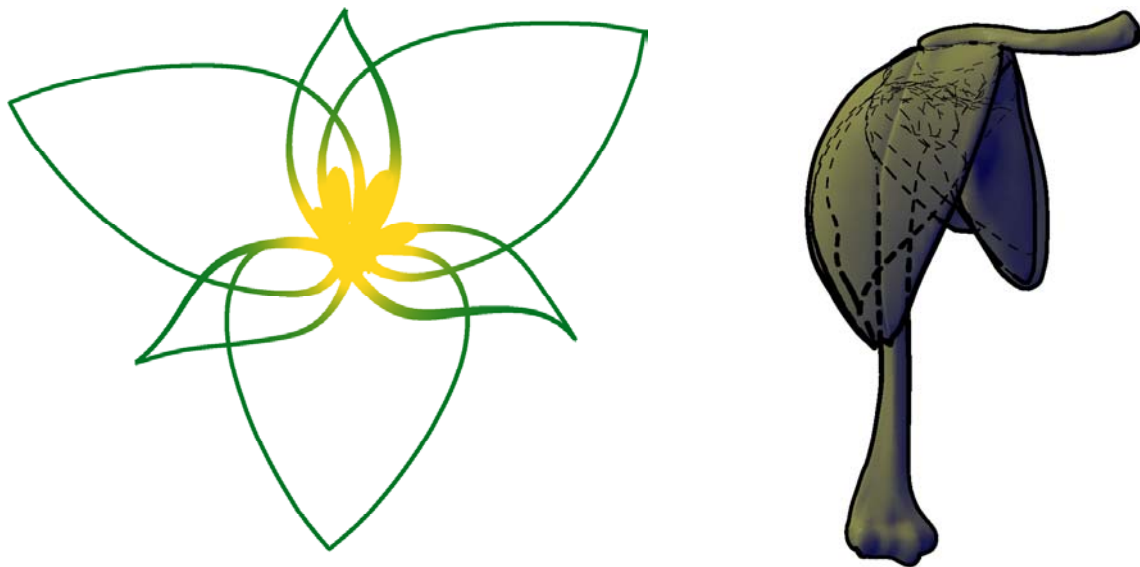
Figure 14: Using G-Strokes by Isenberg & Brennecke [2006] in conjunction with NPR Lenses [Neumann et al. 2007] to apply local style changes to line drawings

pixel images are the by far most often used form of image representation and are supported by virtually all systems where images may be needed.

The second class of image representations, vector graphics, does represent primitives as separate structures in the stored files. Therefore, it is easier to maintain primitives as separate entities in the final image as well and to allow further processing (e.g., changing stroke paths or selecting subsets of primitives). Vector images also do not have an inherent resolution and are rasterized to the resolution needed for a specific case which in most cases leads to a higher quality in the representation. It also results in their data volume usually being smaller than the equivalent pixel image, depending on the number and complexity of elements stored and the resolution of the pixel image [Isenberg *et al*. 2005]. However, as vector images store elements in analytic form, they also have to be interpreted every time they are displayed, thus requiring more time for this process than pixel images.

5 Conclusion

In this paper we have compared groups of image-morphologic primitives in non-photorealistic rendering. These include NPR shading techniques, stroke-based rendering, the generation of sparse line drawing, graphtals, and area primitives. We have shown that NPR techniques tend to work with elements that are larger than an individual pixel, usually inspired by the tools used traditionally in artistic depiction and the marks created by them. We have discussed that some techniques add such artifacts to images to give the impression of the traditional technique but that there are also a number of techniques where marks actually correlate to meaningful structures in images. We showed that the primitives may or may not be explicitly represented in the rendering process, resulting in more or less freedom for subsequently changing the appearance of the elements. Finally, we briefly

touched on that this explicit representation might be carried over to the stored image in form of a vector graphic, which allows higher quality reproduction as well as post-processing on the primitive level.

The overview that is given in this paper presents a morphologic (i.e., syntactic) view of the primitives used in non-photorealistic rendering. This naturally does not touch on the important issues of semantics and, in particular, pragmatics of the images produced in such processes.

Acknowledgments

References

Anjyo, K.-I. & Hiramitsu, K.: Stylized Highlights for Cartoon Rendering and Animation. IEEE Computer Graphics and Applications 23, 4 (July/Aug.), 2003, 54–61.

DeCarlo, D. & Finkelstein, A. & Rusinkiewicz, S. & Santella, A.: Suggestive Contours for Conveying Shape. ACM Transactions on Graphics 22, 3 (July), 2003, 848–855.

Deussen, O. & Hamel, J. & Raab, A. & Schlechtweg, S. & Strothotte, T.: An Illustration Technique Using Hardware-Based Intersections and Skeletons. In Proc. of GI'99, Morgan Kaufmann, San Francisco, 1999, 175–182.

Deussen, O. & Hiller, S. & Van Overveld, C. W. A. M. & Strothotte, T.: Floating Points: A Method for Computing Stipple Drawings. Computer Graphics Forum 19, 3 (Aug.), 2000, 40–51.

Di Blasi, G. & Gallo, G.: Artificial Mosaics. The Visual Computer 21, 6 (July), 2000, 373–383.

Elber, G. & Wolberg, G.: Rendering Traditional Mosaics. The Visual Computer 19, 1 (Mar.), 2003, 67–78.

Gooch, B. & Gooch, A. A.: Non-Photorealistic Rendering. A K Peters, Natick, 2001.

Gooch, A. A. & Gooch, B. & Shirley, P. & Cohen, E.: A Non-Photorealistic Lighting Model for Automatic Technical Illustration. In Proc. of ACM SIGGRAPH'98, ACM Press, New York, 1998, 447–452.

Grabli, S. & Turquin, E. & Durand, F. & Sillion, F. X.: Programmable Style for NPR Line Drawing. In Rendering Techniques 2004, Proc. of the EG Symp. on Rendering'04, EUROGRAPHICS Association, Aire-la-Ville, Switzerland, 2004, 33–44, 407.

Haeberli, P.: Paint By Numbers: Abstract Image Representations. ACM SIGGRAPH Computer Graphics 24, 3 (Aug.), 1990, 207–214.

Hausner, A.: Simulating Decorative Mosaics. In Proc. of ACM SIGGRAPH'01, ACM Press, New York, 2001, 573–580.

Hertzmann, A. & Zorin, D.: Illustrating Smooth Surfaces. In Proc. of ACM SIGGRAPH'00, ACM Press, New York, 2000, 517–526.

Hertzmann, A.: Painterly Rendering with Curved Brush Strokes of Multiple Sizes. In Proc. of ACM SIGGRAPH'98, ACM Press, New York, 1998, 453–460.

Hertzmann, A.: A Survey of Stroke-Based Rendering. IEEE Computer Graphics and Applications 23, 4 (July/Aug.), 2003, 70–81.

Isenberg, T. & Brennecke, A.: G-Strokes: A Concept for Simplifying Line Stylization. Computers & Graphics 30, 5 (Oct.), 2006, 754–766.

Isenberg, T. & Carpendale, M. S. T. & Sousa, M. C.: Breaking the Pixel Barrier. In Proc. of CAe'05, Eurographics Association, Aire-la-Ville, Switzerland, 2005, 41–48.

Isenberg, T. & Freudenberg, B. & Halper, N. & Schlechtweg, S. & Strothotte, T.: A Developer's Guide to Silhouette Algorithms for Polygonal Models. IEEE Computer Graphics and Application 23, 4 (July/Aug.), 2003, 28–37.

Kaplan, M. & Cohen, E.: Computer Generated Celtic Design. In Rendering Techniques'03, Eurographics Association, Aire-la-Ville, Switzerland, 2003, 9–19.

Kaplan, C. S. & Salesin, D. H.: Escherization. In Proc. of ACM SIGGRAPH'00, ACM Press, New York, 2000, 499–510.

Kaplan, C. S. & Salesin, D. H.: Islamic Star Patterns in Absolute Geometry. ACM Transactions on Graphics 23, 2 (Apr.), 2004, 97–119.

Kaplan, M. & Gooch, B. & Cohen, E.: Interactive Artistic Rendering. In Proc. of NPAR'00, ACM Press, New York, 2000, 67–74.

Kowalski, M. A. & Markosian, L. & Northrup, J. D. & Bourdev, L. & Barzel, R. & Holden, L. S. & Hughes, J. F.: Art-Based Rendering of Fur, Grass, and Trees. In Proc. of ACM SIGGRAPH'99, ACM Press, New York, 1999, 433–438.

Markosian, L., Meier & B. J., Kowalski, M. A. & Holden, L. S. & Northrup, J. D. & Hughes, J. F.: Art-based Rendering with Continuous Levels of Detail. In Proc. of NPAR'00, ACM Press, New York, 2000, 59–64.

Mason, K. & Denzinger, J. & Carpendale, S.: Negotiating Gestalt: Artistic Expression by Coalition Formation between Agents. In Proc. of Smart Graphics'05, Springer, Berlin, vol. 3638 of LNCS 2005, 103–114.

Meier, B. J.: Painterly Rendering for Animation. In Proc. of ACM SIGGRAPH'96, ACM Press, New York, 1996, 477–484.

Neumann, P. & Isenberg, T. & Carpendale, S.: NPR Lenses: Interactive Tools for Non-Photorealistic Line Drawings. In Proc. of Smart Graphics'07, Springer, Berlin, Heidelberg, vol. 4569 of LNCS, 2007, 10–22.

Ostromoukhov, V. & Hersch, R. D.: Artistic Screening. In Proc. of ACM SIGGRAPH'95, ACM Press, New York, 1995, 219–228.

Ostromoukhov, V.: Mathematical Tools for Computer-Generated Ornamental Patterns. In Electronic Publishing, Artistic Imaging and Digital Typography. Springer, Berlin, 1998, 193–223.

Ostromoukhov, V.: Digital Facial Engraving. In Proc. of ACM SIGGRAPH'99, ACM Press, New York, 1999, 417–424.

Praun, E. & Hoppe, H. & Webb, M. & Finkelstein, A.: Real-Time Hatching. In Proc. of ACM SIGGRAPH'01, ACM Press, New York, 2001, 581–586.

Saito, T. & Takahashi, T.: Comprehensible Rendering of 3-D Shapes. ACM SIGGRAPH Computer Graphics 24, 3 (Aug.), 1990, 197–206.

Salisbury, M. P. & Anderson, S. E. & Barzel, R. & Salesin, D. H.: Interactive Pen-and-Ink Illustration. In Proc. of ACM SIGGRAPH'94, ACM Press, New York, 1994, 101–108.

Salisbury, M. P. & Anderson, C. & Lischinski, D. & Salisin, D. H.: Scale-Dependent Reproduction of Pen-and-Ink Illustration. In Proc. of ACM SIGGRAPH'96, ACM Press, New York, 1996, 461–468.

Salisbury, M. P. & Wong, M. T. & Hughes, J. F. & Salesin, D. H.: Orientable Textures for Image-Based Pen-and-Ink Illustration. In Proc. of ACM SIGGRAPH'97, ACM Press, New York, 1997, 401–406.

Schlechtweg, S. & Germer, T. & Strothotte, T.: RenderBots—Multi Agent Systems for Direct Image Generation. Computer Graphics Forum 24, 2 (June), 2005, 137–148.

Secord, A.: Weighted Voronoi Stippling. In Proc. of NPAR'02, ACM Press, New York, 2002, 37–44.

Smith, A. R.: Plants, Fractals, and Formal Languages. ACM SIGGRAPH Computer Graphics 18, 3 (July), 1984, 1–10.

Stroila, M. & Eisemann, E. & Hart, J. C.: Clip Art Rendering of Smooth Isosurfaces. IEEE Transactions on Visualization and Computer Graphics. 14, 1 (Jan./Feb.), 2008, 135--145.

Strothotte, T. & Schlechtweg, S.: Non-Photorealistic Computer Graphics. Modeling, Animation, and Rendering. Morgan Kaufmann, San Francisco, 2002.

Winkenbach, G. & Salesin, D. H.: Computer-Generated Pen-and-Ink Illustration. In Proc. of ACM SIGGRAPH'94, ACM Press,New York, 1994, 91–100.

Winkenbach, G. & Salesin, D. H.: Rendering Parametric Surfaces in Pen and Ink. In Proc. of ACM SIGGRAPH'96, ACM Press, New York, 1996, 469–476.

Wong, M. T. & Zongker, D. E. & Salesin, D. H.: Computer-Generated Floral Ornament. In Proc. of ACM SIGGRAPH'98, ACM Press, New York, 1998, 423–434.

Yang, H.-L. & Yang, C.-K.: A Non-Photorealistic Rendering of Seurat's Pointillism. In Advances in Visual Computing, Part 2, Springer, Berlin, 2006, 760–769.

Zander, J. & Isenberg, T. & Schlechtweg, S. & Strothotte, T.: High Quality Hatching. Computer Graphics Forum 23, 3 (Sept.), 2004, 421–430.