Computational Aesthetics in Graphics, Visualization and Imaging (2005)
L. Neumann, M. Sbert, B. Gooch, W. Purgathofer (Editors)

Breaking the Pixel Barrier

Tobias Isenberg   M. Sheelagh T. Carpendale   Mario Costa Sousa

Department of Computer Science, University of Calgary, Canada
{isenberg | sheelagh | mario}@cpsc.ucalgary.ca

Abstract

The majority of the current advances in computer graphic rendering strive for fast and realistic creation of pixel images, e. g., for the film and gaming industry. This development, unfortunately, leads to various problems due to limitations of pixel images, in particular, when they are not used for screen viewing. Thus, in this paper we argue for the placement of greater emphasis on the generation of vector graphics. Vector graphics offer the best approach for achieving effectiveness for both media simulation and illustration techniques. We discuss advantages of using vector graphics, pose a number of questions in this context, and evaluate directions of further research.

Categories and Subject Descriptors (according to ACM CCS): I.3.0 [Computer Graphics]: General

1. Introduction

The overall computational quest for speed, particularly as driven by the film and gaming industries, has lead to an increasing research emphasis on rendering techniques that produce pixel images. Although these results are effective when viewed on a computer screen, they have limitations in terms of storage space, zooming, and other forms of presentation such as printing. In these cases better results can be achieved when images are represented in an analytic form, i. e., as vector graphics. The development towards pixel rendering stands in contrast to the long tradition of image production for print media, mainly in form of illustrations. Even very early computer graphics dealt almost exclusively with vector output. However, with the invention of the frame buffer, research in this area has been considerably reduced and only a few recent publications address high-quality vector output.

Since these issues are particularly important for images designed for illustrative purposes that are frequently viewed in print, we will focus on the newly emerged field of non-photorealistic rendering (NPR) which strives to produce renditions that are more comprehensible, illustrative, or artistic. However, we will discuss how increased use of vector graphics will also support a more general improvement in visual quality. We argue that development in this area of NPR should be intensified and more techniques for creating appealing vector images should be devised.

NPR draws from traditional artwork and illustration and thus, it has a rich vocabulary of expression and diverse applications areas. Many NPR algorithms have been introduced that simulate various artistic and traditional techniques. In addition to creating new rendering styles, NPR research also aims to improve existing techniques in terms of speed and quality. In the last few years, this quest for speed has resulted in many methods for using computer graphics hardware in NPR algorithms. Techniques such as vertex or pixel shaders that were originally devised to speed up photorealistic rendering mainly for the games sector are now employed in NPR but have only enforced pixel output.

2. State of the Art in Vector Rendering

Recent achievements in non-photorealistic vector graphic rendering include the generation of line drawings in two dimensions using *skeletal strokes* [HL94] and from three-dimensional models using object-space silhouette and feature extraction methods (see survey in [IFH*03]). These can be combined with Gouraud shaded versions of the same object using a non-photorealistic illumination model (see Figure 1). In addition, techniques for producing hatching [SABS94, DHR*99, HZ00, RK00, JEGPO02, ZISS04] and stippling renditions [DHvS00, Sec02] have been conceived in both two and three dimensions (also see Figures 2 and 3). Also, certain ornaments such as floral, Islamic, and Celtic patterns have been reproduced as vector graphics [Ost98, WZS98, Gla99b, Gla99a, Kap02, KC03] (also see Figure 4).

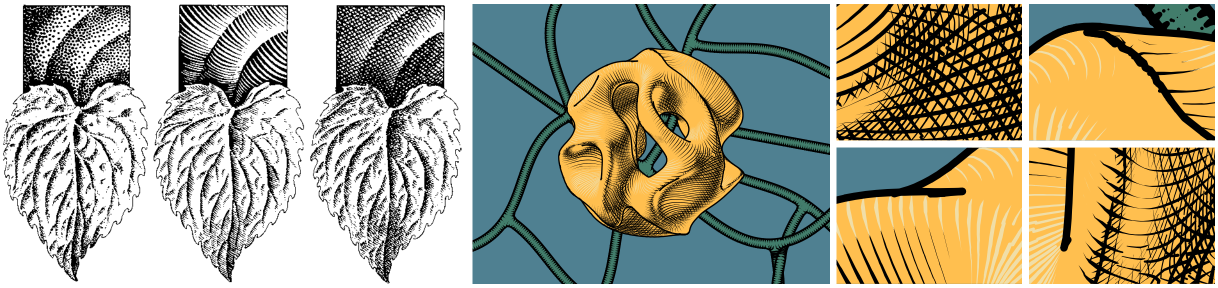


Figure 2: *Hand-made drawings (left, from [Wes83], page 85) compared to a color vector image using silhouettes and high-quality cross-hatching techniques from [ZISS04] (middle). The close-ups of the computer-generated image (right) show regions both with good approximation of cross-hatching as well as artifacts that are still present. These are caused by the 3D model, the silhouette hidden-line removal, and the hatching technique.*

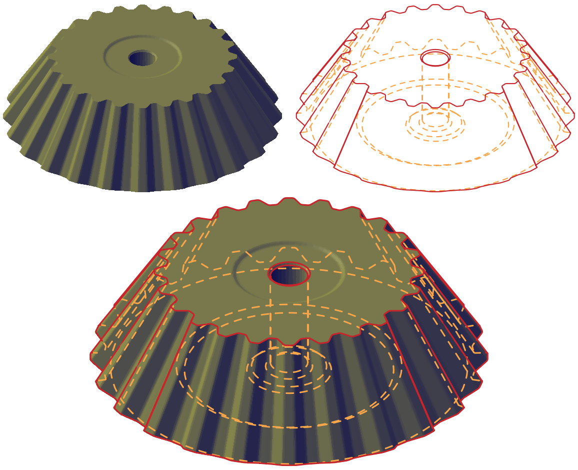


Figure 1: *Gouraud shading using the Gooch illumination model and lines in a vector graphic illustration.*

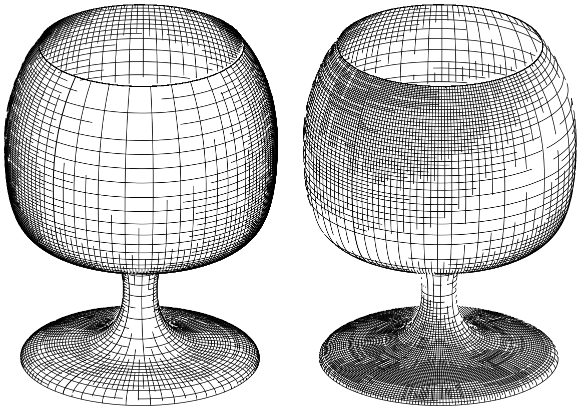


Figure 3: *Isocurve line rendering generated using Gershon* ELBER*'s* IRIT *software (http://www.cs.technion.ac.il/˜irit/).*

3. A Call for Vector Graphic Rendering

A pixel image is constructed by specifying the colors with which each pixel should be drawn. This structure is difficult to manipulate other than through image processing. In contrast, a vector graphic representation better supports further manipulations on a more meaningful level because of their analytic description of the image's content. In addition, once a pixel image has been rendered for a specific display size its resolution, as far as quality output is concerned, can be considered fixed. Pixel images can only be reduced in size and resolution while maintaining contrast at the same time—a technique used frequently for mip-mapping in texture rendering. An increase in size, on the other hand, will lead inevitably to contrast decrease and, thus, lower image quality. The resolutions required for reproduction in print (e. g., $\geq$1200 dpi) are by one or more orders of magnitude higher that those necessary for screen viewing (typically $<$100 ppi) for which most pixel renditions are produced. Even the most advanced display technology only reaches resolutions of 204 ppi at the moment (IBM T221).

Pixel images typically require a lot of storage space. This requirement grows exponentially and is coupled to the increase of image resolution rather than to the number of displayed items. Compressing digital pixel images does not significantly reduce the storage space problem because the images will have to be uncompressed into memory for processing or display. When using full color or gray scale pixel images in the print media, further limitations emerge. A readily apparent problem is that the many gray values and subtle color shades cannot be easily reproduced because the current printing technology uses only three colors and black and either applies color ink to the paper or leaves the paper blank. Thus, gray scale or multi-color images have to be half-toned to achieve the desired shades.

Advantages of vector graphics are numerous. Their use allows viewers to interactively zoom into images to reveal more detail where it is needed. In order for pixel images to

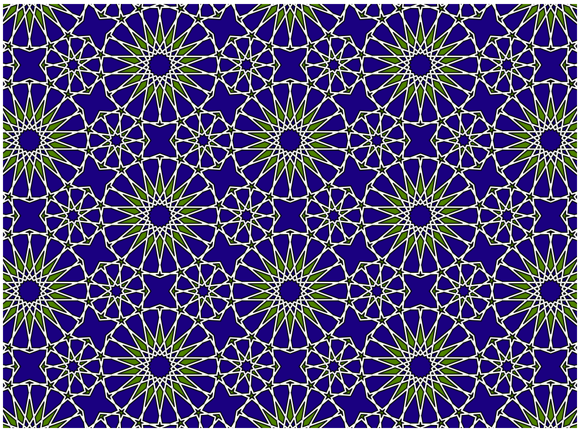


Figure 4: *Islamic pattern generated with algorithms from [Kap02] using Craig* KAPLAN*'s* TAPRATS *software.*

allow the same interaction, the pixel image would need to be sampled with high resolution everywhere to avoid the "fat pixels" which would otherwise appear. Vector graphics, in contrast, are scan-converted on demand. In addition, even if pixel images were available at a very high resolution they would only very rarely have the resolution that is exactly right for the desired zoom factor. Thus, they would have to be re-sampled for the requested view which could lead to a loss in sharpness of the depicted features. Vector graphics, in contrast, can always be rendered at the correct resolution maintaining the optimal sharpness of the depicted objects.

Traditional drawing tools usually have strokes rather than pixels as their foundation. To be able to reproduce these techniques faithfully we should take this fact into account. Moreover, artists usually think in terms of regions for placing strokes in different lengths, clusters, directional fields etc. This objective of rendering large regions of the subject stands in contrast to the very local character of pixels. In NPR, only few researchers have considered this fact [DHR*99, DS02]. Fine details can also be important. In many traditional techniques, very small artifacts and effects result from the combined interactions between the drawing/painting device, hand gestures, media, and the drawing/painting surface. For instance, in traditional engraving part of its expressive "signature" is due to those artifacts. As a result they are hard to properly simulate in a pixel-based NPR engine.

4. Print Quality Case Study

In order to compare the differences in print quality between pixel and vector images we have generated a set of pixel images at various resolutions and placed them adjacent to their vector graphic counterparts (Tables 1 and 2). We have chosen an example image with enough detail to achieve a reasonable comparison. The pixel images were produced for display at 5 cm by 3 cm at resolutions of 100 ppi (the approximate resolution of current displays), 300 ppi (which serves as an example for screenshots used frequently in technical papers), 1200 dpi (which is the state of the art for laser printers), and 2400 dpi (which represents resolutions used in commercial printing). Both the compressed and uncompressed file sizes are included in the tables. To achieve vector quality with a pixel image one must match resolutions throughout the entire process, including the resolution for which the images was created and all aspects of output resolution such as size on the page of software, printer expectation, and size of the page in the printer. A mismatch of even one pixel at any point in the process will interfere with quality. To keep our tables representative of common occurrences in practice, the pixel images are printed at 4.5 cm wide instead of the 5.0 cm for which they were created.

Table 1 shows the two smaller pixel images (100 ppi, 300 ppi) on the left and the vector graphic on the right. Note that the printed quality is fuzzy and the halftoning of the gray-scale images is clearly visible. However, the file sizes are much smaller than the ones of the vector graphic. In Table 2 we show the resolutions that were intended for print reproduction (1200 ppi and 2400 ppi). Here we can see that although the pixel images were produced for print reproduction, there are still artifacts to be observed (e. g., at the ends of the hatching lines) if compared to the vector graphic. In addition, the file sizes are in the same ballpark or even larger than the ones for the vector graphic.

This comparison shows that high quality reproduction in print is difficult to achieve with pixel images. They are inferior to vector images depicting the same objects even if produced solely for the purpose of printing. Thus, vector images are, generally, more flexible because they generate superior quality for both print reproduction and screen viewing.

5. New Problems and Challenges

With one of the advantages of analytic vector rendering techniques—the ability to interactively zoom into increasing detail—comes new research challenges. The enhanced ability to see detail makes it more possible to see small imperfections. Figure 2 shows examples for details of both hatched hand-drawn images (on the left) and hatched computer-generated images (on the right). In the details of the computer-generated image several artifacts are visible: some lines have considerable noise, some hatching lines are interrupted, some silhouette lines end abruptly, and some lines double back. These types of problems with the line placement and line rendering remain as challenges in our efforts to produce higher quality images.

One possible research direction is to develop closer collaborations with graphic designers and illustrators. We believe artists' lifelong training to produce quality renderings makes it possible for them to provide advice for improving

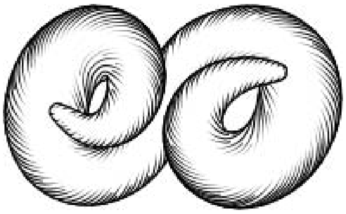
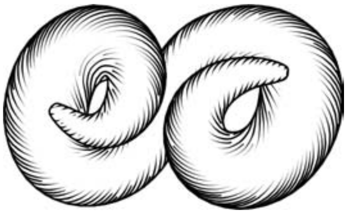
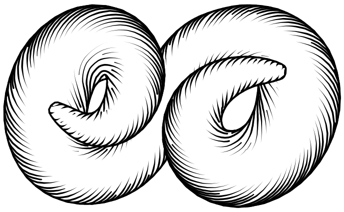
	197 x 119, 100 ppi, 8 bit	591 x 356, 300 ppi, 8 bit	vector graphic
uncompr.	23 kB	205 kB	417 kB
PNG/PDF	11 kB	52 kB	117 kB
image			
detail			

Table 1: *Comparison images: The first example reflects the screen sizes and resolutions used in today's displays, the second corresponds to images often used for figures in scientific papers, the third one is the vector graphic image. The pixel images were produced for a width of 5 cm. Note the halftoning artifacts resulting from printing a gray-scale image.*

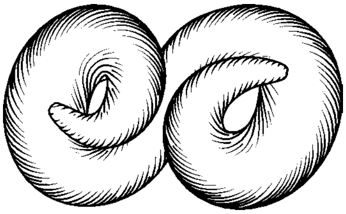
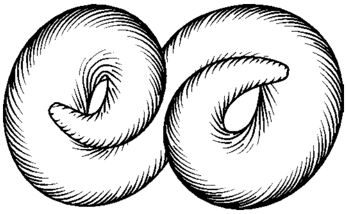
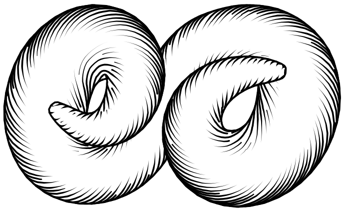
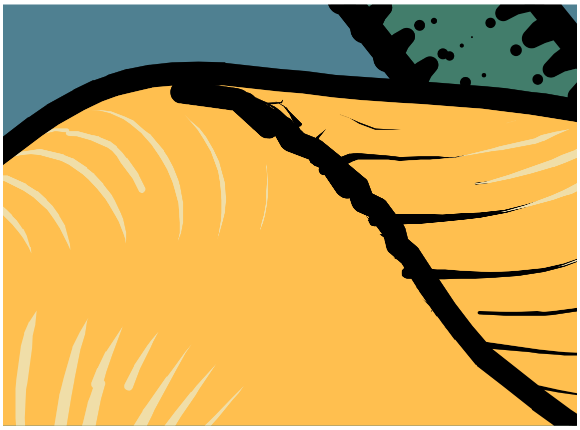
	2362 x 1423, 1200 dpi, 1 bit	4724 x 2846, 2400 dpi, 1 bit	vector graphic
uncompr.	410 kB	1,641 kB	417 kB
PNG/PDF	68 kB	166 kB	117 kB
image			
detail			

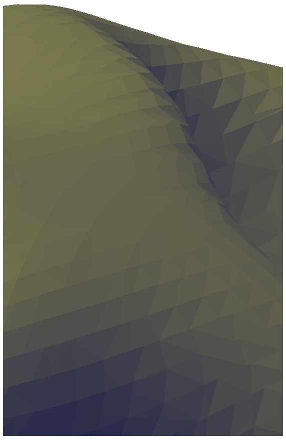
Table 2: *Comparison of pixel images for print with vector graphic. Note the fuzziness of the pixel images, in particular, at the ends of the hatching lines. These occur because the pixel images are not printed at exactly the size they were created for (4.5 cm instead of 5 cm), a case that occurs often in practice. Also, the file sizes are similar or bigger than the vector graphic.*

our techniques. In particular, their input can help to improve the placement of strokes by using the non-local character of vector rendering and by considering whole regions of the canvas at a time.

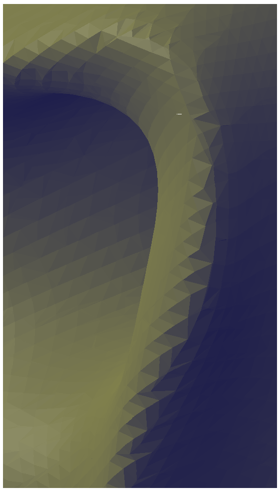
Other artifacts that should be avoided are caused by problems with the 3D models on which the rendering is based. This is due to the fact that most models used in analytic non-photorealistic rendering were originally conceived for photorealistic pixel rendering. Certain problems or inaccuracies in the models that do not show up in pixel rendering can become apparent in vector rendering. For example, the model in Figure 5 shows how the placement of a 3D model's surface triangles can be less than perfect and then lead to the creation of a noisy silhouette line.



(a) Detail from Figure 2 with noisy silhouette line.



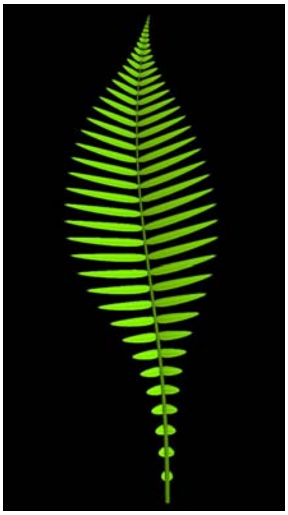
(b) Same region as in (a) shown using Gooch shading.



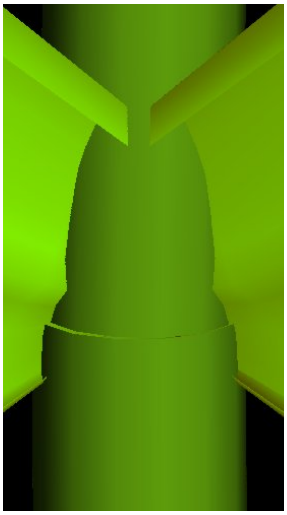
(c) Different region of the model with same problem.

Figure 5: *Model artifacts (a) caused by noisy surface (b),(c). Note the non-optimal triangle placement in (b) and (c).*

A problem common in 3D models for photorealistic pixel rendering is illustrated by Figure 6: non-connected model parts. The figure shows that this usually does not pose a problem in photorealistic rendering as long as the errors are smaller than one pixel. However, in analytic rendering used for generating vector graphics even small errors in the model show up as separate lines that depict non-meaningful features. Also, other problems such as double triangles, small folds, or even missing triangles may lead to more problems than in photorealistic rendering.



(a) Whole fern leaf.



(b) Close-up.

Figure 6: *Model artifacts due to non-connected model parts. In analytic silhouette rendering the stem would generate lines across at each point where leaves connect because it is modeled as several non-connected cylinders. In photorealistic rendering this is not a problem (see (a)). Images courtesy of Przemyslaw Prusinkiewicz, used with permission.*

While, as just discussed, some artifacts are unwanted, there are desirable artifacts. For instance, when examining the enlargements of the hand-drawn image in Figure 2 lines can be seen that are not necessarily all entirely straight but each do contain some character due to hand movements and drawing tool interaction with the canvas. Also, the lines are not all exactly parallel to each other and do not all have the same length. These types of artifacts make a drawing more interesting and less sterile and, therefore, should be included in computer-generated drawings.

6. Open Questions and Future Work

Although vector graphics are more flexible than pixel images in terms of interactive magnification, there is still a limit to this interaction. If the magnification factor gets too big, only large areas of the same color are visible on the screen. In this context it would be good to have some type of vector graphic mip-mapping that automatically adds or removes detail when zooming is applied. SALISBURY et al. suggested such a technique for their prioritized stroke textures [SALS96]. It would be interesting to develop similar techniques for other images rendering styles.

Simulating traditional tools as vector graphics also forms an interesting challenge. Instead of using pixel textures, as

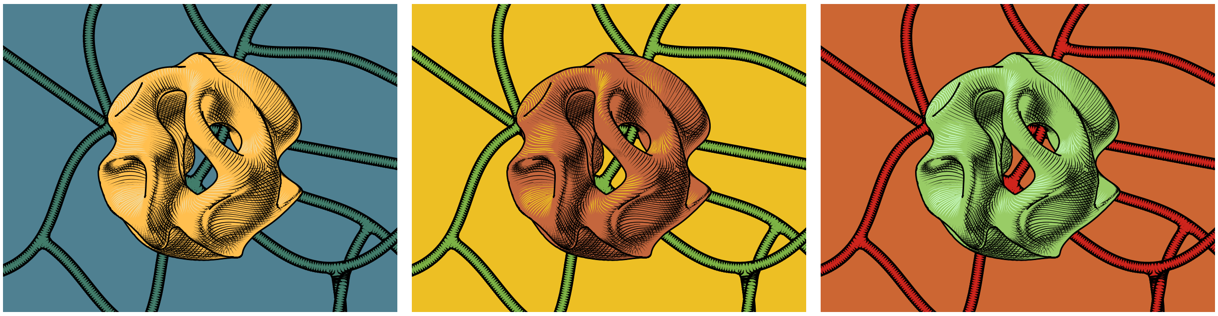


Figure 7: *Vector graphics are well suited for multi-color printing by using several spot colors for the different layers to avoid the necessity of halftoning. The figure shows three examples with different spot colors applied to the five layers (four colors and black). The different colors applied to hatching, silhouettes, and backgrounds produce shading effects that simulate colors not actually present in the images. Note that this effect could only be simulated due to lack of spot color printing.*

is now done, this would require reproducing the characteristics of these tools as vector graphic strokes. HSU and LEE showed a system that uses such vector textures [HL94]. While these were specified manually, it may be possible to create such vector textures for various tools automatically.

Vector graphics easily support various layers of data. This fact is used in printing to avoid the otherwise necessary halftoning which is required by the commonly used CMYK color model. The image is segmented into several areas that are printed independently using *spot colors* that are especially produced for the specific graphic. In this way several colors (possibly more than three) can be combined to achieve the desired effect without halftoning artifacts. Using a smart arrangement of colored primitives in the image it is, thus, possible to achieve shading effects that simulate perceived colors which are not actually present in the printed image (see Figure 7 and [ZISS04]). This raises the question whether it is possible to formulate a color model that can predict these perceived colors. In other words, is it possible to derive an arrangement of lines using specific spot colors that is necessary to simulate a desired color?

Another interesting question is whether there is a specific vector graphic pipeline and how it is different from the traditional pixel rendering pipeline. It is possible that the layer aspect of vector graphics could be usefully reflected in the vector rendering pipeline. However, it is not clear whether such a pipeline could be sufficiently general to support the many different current and future vector rendering techniques. Related to this problem is whether it is possible to use current graphics hardware to speed up a vector graphic pipeline's rendering process and the rasterization of vector graphics since both steps tend to be fairly slow. Potentially, this might require new hardware to support these tasks. However, with future versions of Microsoft Windows® supporting vector graphic user interfaces (Windows Vista™ will have a vector-based rendering engine) this may be worth investigating.

In general, the creation of a greater variety of high quality non-photorealistic vector rendering techiques can provide exciting challenges for many years to come. For example, techniques that were originally conceived for printing pixel images (i. e., halftoning and artistic screening) could be used to generate vector graphics, e. g., [OH95, OH99, Ost99b, Ost99a]. These techniques were originally presented using pixel images but vector graphic generation seems to be possible. Interestingly enough, artistic screening provides an intriguing level-of-detail/mipmapping effect through showing the image at low resolutions and text or ornaments at high resolutions. Additional techniques that are well suited for vector rendering are comic and cartoon rendering as well as artistic mosaics. Also, as suggested by David SALESIN in his keynote talk at NPAR 2002, ornaments and calligraphy may lend themsleves to vector graphic rendering ([Sal02], Challenges 1.2 and 1.3). Other techniques include the generation of stylized initial letters for fonts or book decorations to reproduce the beautiful art pieces of early printing (see Figure 8).

7. Why Are Vector Graphics Little Used?

One final question that we would like to address is why do people make so little use of vector rendering despite its many advantages? There may be several reasons contributing to this. One of the reasons may be that technology for vector graphic processing is not as readily available as pixel processing tools. On every new computer there are pixel tools either pre-installed or there are free tools available on the Internet (such as GIMP). However, tools for vector graphics are not as easily accessible. There are certainly tools such as Corel Draw® and Adobe Illustrator® but they have to be purchased and are fairly expensive.

In addition, it seems that more effort is necessary to produce vector output from programs. In many development environments there is support for OPENGL rendering but not for producing vector graphic output. For vector graphics a programmer has to specifically install libraries such as

Figure 8: *Two pages with examples for book decorations and stylized initial letters and ornaments from one of the first printed books—the Gutenberg bible. Courtesy and copyright of Göttingen State and University Library, Germany (http://www.gutenbergdigital.de/bibel.html), used with permission.*

GL2PS to redirect OPENGL calls or ClibPDF to directly output vector primitives. Also, with fast graphics hardware being readily available in modern PCs and that it is frequently used in many applications such as computer games, the dominance of the hardware-accelerated pixel pipeline is being fostered. Another reason is that the necessity and the know-how for rendering vector graphics is seldom taught in computer graphics courses. It seems that high quality results are not encouraged. Potentially due to pressing deadlines either for student projects or conference submissions, people tend to make a quick screenshot rather than produce a high-quality vector image, even if this were possible. However, people may also be afraid of their work being stolen since vector images contain the image in its highest possible quality. Pixel images may be downsampled for use in freely available electronic documents but this is not as easy with vector graphics. The fact that errors in the used 3D models and in the rendering techniques show up much more readily in vector graphics may also be a deterrent. Moreover, technical problems also appear with documents containing vector graphics. For example, some documents may not only print slower but also seem to cause more errors if they contain vector images. In addition, vector images are not always displayed correctly depending on how the vector graphic primitives are specified (display, e. g., the PDF of [Sec02] using Adobe® Reader® 7.0 and zoom in and out). Also, since vector graphics are an interpreted representation of an image each tool has to be able to understand all parts of this representation. However, some tools do not support the entire syntax possible in some vector graphic standards which leads to compatibility problems. For example, the vector graphic Gouraud shading used in Figure 1 that is supported by the PDF format cannot successfully be imported into Corel Draw®.

8. Conclusion

The generation of high quality vector graphic images is a field that needs to be explored in much more detail. Vector graphics closely relate to the drawing and painting techniques known from traditional art since both rely on the primitive of a stroke rather than a pixel. In addition, we can draw from an enormous amount of experience in both traditional artistic techniques and scientific illustration. Vector graphics are often much better suited for reproduction in print than pixel images, in particular, for scientific illustrations which have been used in the print media for centuries. This lead us to several new research challenges. These include avoiding bad artifacts and introducing good ones, looking more closely at necessary characteristics of 3D models to yield good vector renditions, how to realize a vector graphic mipmapping to introduce detail as it is needed, how to simulate traditional tools for vector rendering, using vector graphic layers effectively to avoid color halftoning and a possible innovative color model for this purpose, and a potential vector graphic rendering pipeline to support many of these aspects.

Acknowledgments

We would like to thank Petra Neumann for her help with some of the images and for suggesting the title of the paper. We also thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for their support.

References

[DHR*99] DEUSSEN O., HAMEL J., RAAB A., SCHLECHTWEG S., STROTHOTTE T.: An Illustration Technique Using Intersections and Skeletons. In *Proc. of Graphics Interface '99*, San Francisco, 1999, Morgan Kaufmann, pp. 175–182.

[DHvS00] 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.

[DS02] DECARLO D., SANTELLA A.: Stylization and Abstraction of Photographs. *ACM Transactions on Graphics 21*, 3, July 2002, 769–776.

[Gla99a] GLASSNER A.: Celtic Knots, Part II. *IEEE Computer Graphics and Applications 19*, 6, 1999, 82–86.

[Gla99b] GLASSNER A.: Celtic Knotwork, Part I. *IEEE Computer Graphics and Applications 19*, 5, 1999, 78–84.

[HL94] HSU S. C., LEE I. H. H.: Drawing and Animation Using Skeletal Strokes. In *Proc. of SIGGRAPH 94*, New York, 1994, ACM Press, pp. 109–118.

[HZ00] HERTZMANN A., ZORIN D.: Illustrating Smooth Surfaces. In *Proc. of SIGGRAPH 2000*, New York, 2000, ACM Press, pp. 517–526.

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

[JEGPO02] JODOIN P.-M., EPSTEIN E., GRANGER-PICHÉ M., OSTROMOUKHOV V.: Hatching by Example: a Statistical Approach. In *Proc. of NPAR 2002*, New York, 2002, ACM Press, pp. 29–36.

[Kap02] KAPLAN C. S.: *Computer Graphics and Geometric Ornamental Design*. PhD thesis, University of Washington, Seattle, 2002.

[KC03] KAPLAN M., COHEN E.: Computer Generated Celtic Design. In *Proc. of Eurographics Workshop on Rendering 2003*, Aire-la-Ville, Switzerland, 2003, Eurographics Association, pp. 9–19.

[OH95] OSTROMOUKHOV V., HERSCH R. D.: Artistic Screening. In *Proc. of SIGGRAPH 95*, New York, 1995, ACM Press, pp. 219–228.

[OH99] OSTROMOUKHOV V., HERSCH R. D.: Multi-Color and Artistic Dithering. In *Proc. of SIGGRAPH 99*, New York, 1999, ACM Press, pp. 425–432.

[Ost98] OSTROMOUKHOV V.: Mathematical Tools for Computer-Generated Ornamental Patterns. In *Electronic Publishing, Artistic Imaging and Digital Typography*. Springer-Verlag, Berlin, 1998, pp. 193–223.

[Ost99a] OSTROMOUKHOV V.: Artistic Halftoning: Between Technology and Art. In *Proc. SPIE Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts V*, 1999, pp. 489–509.

[Ost99b] OSTROMOUKHOV V.: Digital Facial Engraving. In *Proc. of SIGGRAPH 99*, New York, 1999, ACM Press, pp. 417–424.

[RK00] RÖSSL C., KOBBELT L.: Line Art Rendering of 3D-Models. In *Proc. of Pacific Graphics 2000*, Los Alamitos, 2000, IEEE Computer Society Press, pp. 87–96.

[SABS94] SALISBURY M. P., ANDERSON S. E., BARZEL R., SALESIN D. H.: Interactive Pen-and-Ink Illustration. In *Proc. of SIGGRAPH 94*, New York, 1994, ACM Press, pp. 101–108.

[Sal02] SALESIN D. H.: Non-Photorealistic Animation & Rendering: 7 Grand Challenges. Keynote talk at NPAR 2002, June 2002.

[SALS96] SALISBURY M. P., ANDERSON C., LISCHINSKI D., SALESIN D. H.: Scale-Dependent Reproduction of Pen-and-Ink Illustration. In *Proc. of SIGGRAPH 96*, New York, 1996, ACM Press, pp. 461–468.

[Sec02] SECORD A.: Weighted Voronoi Stippling. In *Proc. of NPAR 2002*, New York, 2002, ACM Press, pp. 37–44.

[Wes83] WEST K.: *How to Draw Plants: The Techniques of Botanical Illustration*. The Herbert Press Limited, 1983.

[WZS98] WONG M. T., ZONGKER D. E., SALESIN D. H.: Computer-Generated Floral Ornament. In *Proc. of SIGGRAPH 98*, New York, 1998, ACM Press, pp. 423–434.

[ZISS04] ZANDER J., ISENBERG T., SCHLECHTWEG S., STROTHOTTE T.: High Quality Hatching. *Computer Graphics Forum 23*, 3, Sept. 2004, 421–430.