Pen-and-Ink Rendering in Volume Visualisation

S. M. F. Treavett M. Chen

Department of Computer Science, University of Wales Swansea, United Kingdom

Abstract

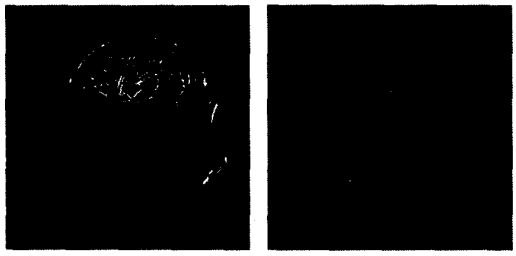
This paper is concerned with the development of non-photorealistic rendering techniques for volume visualisation. In particular, we present two pen-and-ink rendering methods, a 3D method based on non-photorealistic solid textures, and a 2^{+}D method that involves two rendering phases in the object space and image space respectively. As both techniques utilize volume- and image-based data representations, they can be built upon a traditional volume rendering pipeline, and be integrated with photorealistic methods available in such a pipeline. We demonstrate that such an integration facilitates an effective mechanism for enhancing visualisation and its interpretation.

Keywords: Volume rendering, non-photorealistic rendering, pen-and-ink rendering, 3D texture mapping

1 Introduction

The past decade has witnessed the rapid development of volume visualisation techniques, driven mainly by applications such as medical imaging and scientific computation. The work in this field has produced a collection of visualisation methods, including surface extraction [1], volume ray casting [2] and splatting [3], together with numerous acceleration techniques.

The majority of work in the field of volume rendering has been focused on the synthesis of photorealistic (PR) images to assist in the visualisation of iso-surfaces and amorphous phenomena contained in volume datasets. By simulating the way objects are viewed by human eyes, these PR techniques generally give good visual cues to the viewer and enable them to interrogate data intelligently. Figure 1 shows some typical visualisations of datasets obtained from a magnetic resonance imaging (MRI) and a computed tomography (CT) scanner respectively.



(a) (b)

Figure 1: *(a) An image generated as a visualisation of a MRI dataset, (b) a visualisation of a CT dataset showing the skin transparently to convey its relationship with the skull.*

Recently there has been a surge of effort in surface graphics for developing non-photorealistic (NPR) modeling and rendering methods. Such development has established a noticeable role in the field of graphical illustration and the entertainment industry. In the 1998 IEEE Visualisation Conference, a call was made for the extension of these techniques to the field of visualisation [4]. Meanwhile, the advances in volume visualisation, coupled with the rapid increase in computer power, also suggest that volume visualisation may be developed into volume graphics [5, 6] as a general purpose graphics technology. It is these developments that motivated our efforts in the application of NPR techniques to volume rendering particularly as effects similar to pen-and-ink had already been used to try and help visualise transparent surfaces [7].

Pen-and-ink based line drawing has been used as an effective form of abstraction for hundreds of years [8]. The artist leaves out a large amount of detail, such as texture and colour, and instead concentrates on outlines and limited 'mood' shading. These types of images have often been used in preference to more accurate images in specific circumstances. For instance a group of psychologists reported in 1996 [9] that, when asked to compare a computer-produced 'sketch' against a photorealistically rendered CAD image, architects showed a great preference for the sketch for certain purposes.

In this paper, we focus on the development of pen-and-ink techniques in a volume rendering pipeline. We will describe two different methods for rendering volume datasets with line drawing effects. We will apply these techniques to datasets with complex geometry such as medical volume datasets, while using mathematically defined scalar fields to illustrate various attributes of the techniques. We will demonstrate that not only can the techniques be used as a cost-effective tool for graphical illustrations involving volume datasets, they can also be employed to enhance the process of extracting meaningful information from volume datasets. In particular, we will show that integrated NPR and PR rendering can offer different levels of abstraction to assist in the interpretation of synthesised images for visualisation.

2 Background of Nonphotorealism

An ever increasing amount of work has been carried out in the field of non-photorealistic (NPR) rendering since its emergence around 1990 [10]. In 1995 an extensive review of the state of the art was published by Landsdown and Schofield [11]. Although not trying to exactly imitate the work of human artists, most researchers in the field of non-photorealism have modeled their expressive effects on a well known image style, including line drawings using pencil and pen [12, 13] oil painting [10] and water colour [14].

Imitation of paintings started with Strassman's detailed simulation of a brushstroke [15] and progressed into more refined and simplistic brush stroke models such as Pham's [16] and Hertzmann's [17] approaches using B-splines [18, 19]. Hertzmann's technique incorporates these strokes into complete painting-style images in the fashion pioneered by Haeberli's oil painting model [10] and continued by Curtis *et al.*'s complex model of water colour paint-

ing [14]. In Strassman's Piranesi [15] system the 2D techniques used by the others are extended by passing some spatial information from the rendering engine into image space as a z-buffer [20].

Due to the inherent simplicity of line drawings many authors have focused their efforts on production of this kind of effect. The papers by Wikenbach and Salesin [21] and Salisbury *et al.* [12] in SIGGRAPH '94 paved the way for this work. Wikenbach and Salesin later extended their pen-and-ink styles to the rendering of parametric surfaces [22] and Elber made further extension for freeform surfaces [23]. Buchanan and Veryovka treated pen-and-ink drawing as a special case of halftoning [24, 25]. Further work by Salisbury, with his colleagues, provided methods of orienting [26] and scaling [27] pen-and-ink textures.

Moving away from pen-and-ink, Sousa and Buchanan have researched into the simulation of graphite pencil drawings [13, 28]. Hamel and Strothtte [29] developed a system which allows users to capture and reproduce particular NPR styles generated with the system.

Among the existing NPR techniques, many have been purely 2D techniques working in image space. Those techniques that have made use of 3D information from graphics models are almost all built upon surface-based representations [13]. Despite the 1998 call [4] for research into NPR techniques in the context of visualisation, adequate effort is yet to be made to develop such techniques for visualisation purposes, and to investigate into their effective deployment in visualisation applications.

3 Pen-and-Ink Rendering

Pen-and-ink techniques can be classified into three main categories according to the specification of pen-and-ink strokes and the use of information associated with the original 3D objects. These categories are:

- *3D drawing*, where 3D strokes are generated in the object space and are projected onto an image plane by a rendering algorithm. A common approach is to generate pen-and-ink strokes as textures and to render the textures with a purposely built renderer [23, 22].
- *2^+D drawing*, where 2D strokes are generated based on 3D or partial 3D information associated with original 3D objects. Such a technique often involves a pre-rendering stage which generates necessary 3D attributes for each pixel in the image space; and an NPR rendering stage which synthesises a pen-and-ink image based on the information which has been forwarded to image space. A typical example is the use of 'G-buffers' employed by a number of pen-and-ink algorithms [25, 30].
- *2D drawing*, where 2D strokes are generated based solely on the 2D information available in image space. Techniques in this category are fundamentally akin to many traditional halftone techniques [27].

The 2D pen-and-ink techniques have limited control over 3D effects that may be represented by strokes. Without the intervention of a user, it is unlikely such techniques can be used effectively for the purpose of visualisation. Our work has therefore focused on the other two categories.

3.1 3D Pen-and-Ink Rendering

A scalar field is the underlying mathematical definition of a volume dataset coupled with an interpolation function. This concept forms the basis of a variety of volume visualisation techniques, including isosurfacing, volume sampling and attribute mapping. It also provides a means for specifying NPR textures in volume rendering. In general, an NPR texture is essentially defined as a filter:

$$F(p, O_{att}, T_{att}) \rightarrow \{opacity, colour\}$$

where p is a point in the spatial domain of a texture, O_{att} is a set of object attributes associated with p (e.g. the corresponding value in the original volume dataset, the RGB values for a mapped colour, etc.), and T_{att} is a set of texture attributes (e.g. density, line width and length, noise level, etc.).

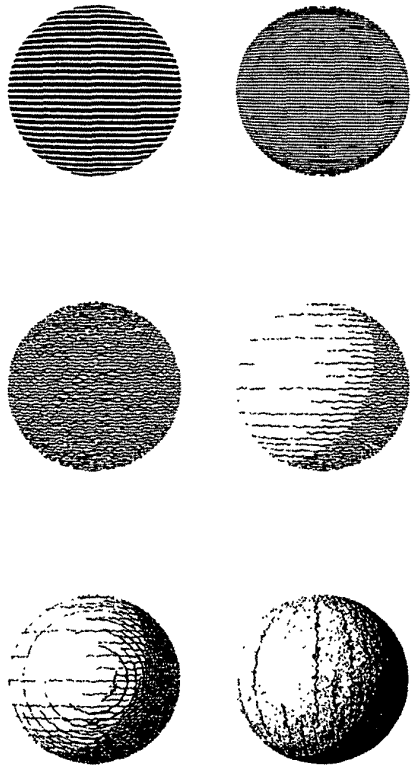


Figure 2: *pen-and-ink techniques of increasing complexity: (a) straight 'pen lines', (b) varying thicknesses, (c) noise added to lines, (d) introducing lighting effects, (e) a second set of lines, (f) advanced techniques.*

All our 3D NPR textures are defined in a normalized domain of $[0, 1] \times [0, 1] \times [0, 1]$. In our rendering pipeline [31] which was designed for multi-volume rendering, there is no extra cost for mapping an arbitrary sampling point to a point in the normalized domain.

Consider a simple texture, as shown in Figure 2(a), that consists of horizontal rings defined in cylindrical manner. The geometry of the texture is controlled by its density

$t_d \in [0,1]$ and ring thickness $t_t \in [0,1]$. We thus have a scalar field:

$$f_1(p,t_d,t_t) = \begin{cases} 0 & \langle t_d K \rangle = 0 \\ 0 & |P_y - \frac{2^{\langle t_d K \rangle - 1} P_y}{2^{\langle t_d K \rangle - 1}}| \geq t_t \\ 1 & |P_y - \frac{2^{\langle t_d K \rangle - 1} P_y}{2^{\langle t_d K \rangle - 1}}| \leq t_t \end{cases}$$

where $\langle\ \rangle$ denotes a *round* operation, and $K(\geq 1)$ defines the maximum number of rings as $2^{(K-1)}$. Scalar field f_1 preserves the consistency and coherence of the texture in terms of varying t_d and t_t. For instance, given a point p and a fixed t_t,

$$f_1(p,t_d,t_t) = 1 \quad \text{implies} \quad \forall d \geq t_d, f_1(p,d,t_t) = 1;$$

and

$$f_1(p,t_d,t_t) = 0 \quad \text{implies} \quad \forall d \leq t_d, f_1(p,d,t_t) = 0.$$

This feature is essential in maintaining coherence in animation even for such a simple texture. In particular, in the following discussions, we will find that many visualisation effects are controlled by t_d and t_t.

The binary output of f_1 easily leads to aliasing, which can be overcome by introducing a simple grey scale function to reduce the ring intensity gradually along their edges. However, a more dense texture could easily cause unnatural artifacts as shown in Figure 2(b). We thus add a noise function into the texture through the use of a set of pre-computed noise lattices [32, 33]. The use of such lattices, instead of a run-time random number generator, is not only for the consideration of computational speed but also for maintaining coherence in animation. As shown in Figure 2(c), the noise function removes the undesirable artifacts shown in Figure 2(b), as well as the degree of artificial perfection which is rare in real life hand drawings.

Together with the anti-aliasing and noise functions, f_1 can be implemented as the following procedural texture

```
float f1(p, td, tt, tn)
Point3  p;
float   td, tt, tn; /* density, thickness, noise */
{
   int     k, r;
   float   y1, y2, dy, fr;

   if ((k = round(td*K)) == 0)
      return(0.0);
   y1 = p.y + noise_field(p)*tn;
   r = (01)<<(k-1);      /* r = 2^(k-1) */
   y2 = round(y1*r)/r;   /* needs float division */
   dy = abs(y1 - y2);
   if (dy >= tt)
       return(0.0);
   else if ((fr = dy/tt) <= 0.5)
      return(1.0);
   else
      return(2.0-2.0*fr);
}
```

During rendering, it is quite possible to treat an NPR texture in the same way as a PR texture, that is by feeding the obtained texture colour as the object colour into an illumination model. However, in this case, the visual cues are in fact provided mainly by the traditional shading, and it somehow defeats the objective of NPR rendering.

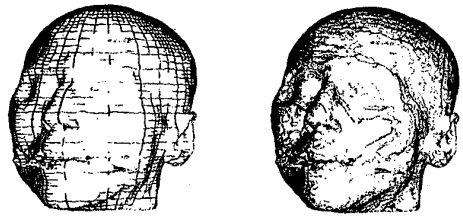
We adopt an approach that applies shading prior to the application of a pen-and-ink texture. Given a point p detected to be on a specified iso-surface, the rendering process first determines the light intensity l at p using a traditional illumination model. It then modifies the texture attributes such as t_d and t_t according l. The resultant texture value is then used to synthesis the pixel colour. The example shown in Figure 2(d) is rendered by the following sequence of operations:

```
Colour F(p, oatt, tatt)
Point3  p;
Oatt    oatt;
Tatt    tatt
{
    float   l, td, tt, tv;
    Colour   c;

    l = illumination(p, oatt);
    switch (tatt.type)
    {
    ...
    case CYLINDRICAL_RINGS:
        td = tatt.td * (1.0 - l);
        tt = tatt.tt * (1.0 - l);
        tv = f1(p, td, tt, tatt.tn);
        break;
    case ...:
    ...
    }

    c.r = (1-tv)(oatt.r*tv + BG.r);
    c.g = (1-tv)(oatt.g*tv + BG.g);
    c.b = (1-tv)(oatt.b*tv + BG.b);
}
```

The last three statements in the above procedure determine a colour at p according to the obtained texture value t_v, which also acts as an opacity value when applying an *over* operation to the colour and a background colour. Later, we will also discuss the use of t_v as an opacity value in the combination of PR and NPR rendering.



(a) (b)

Figure 3: *Pen-and-ink style rendering of a CT dataset.*

A variety of textures can be defined in a way similar to $f_1(p,t_d,t_t,t_n)$, and they can easily be combined together to give more visual cues (Figures 2(e) and 2(f)). Figure 3 shows the application of two pen-and-ink textures to a CT dataset. Both volumes were lit by a point light source in the front.

3.2 2⁺D Pen-and-Ink Rendering

While a 3D pen-and-ink rendering method may benefit from the availability of a variety of 3D information associated with graphics objects, it faces a degree of difficulties in appreciating the information in the context of the final image. For instance, during the rendering of a volume dataset, it is trivial for a renderer to locate information near a sampling point or a voxel. However it is not that straightforward for the renderer to know if such information would ever be used to determine the colour of any pixel in the image plane. This leads to the desire to a two-phase approach, with which the *first phase rendering* in the object space identifies all necessary 3D information relevant to what is to be synthesised in the image plane, and the *second phase rendering* in the image space applies NPR effects, often to more than one pixel, with a global view of all relevant information. We refer such techniques as 2^+D NPR techniques, as the

NPR renderer has access to a restricted collection of spatial information.

Different information from the object space can be used to influence the pen-and-ink drawing in different ways. An outline of an object, such as the one shown in Figure 4(a) may be determined by the variation of the distance of neighbouring pixels in the image plane to the visible surface or, alternatively, by the angle between the surface normals and the view plane normals. The length, thickness and density of strokes may be determined by normals in relation to a light source, as shown in Figure 4(b) where all strokes follow the same direction. Direction of strokes can be controlled by the curvature of the surface (Figure 4(c)). The final image shown in 4(d) is generated by an amalgamation of the interior shading (Figure 4(c)) and outline (Figure 4(a)) images.

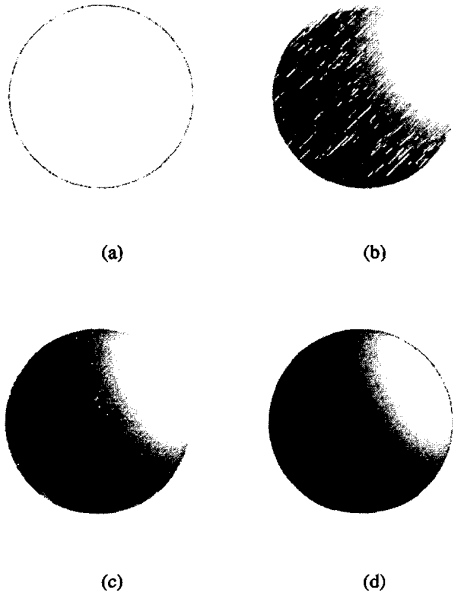


Figure 4: *The sphere rendered in pen-and-ink as: (a) outline only, (b) interior shading with lines in oue direction only, (c) interior shading with control of line direction (d) (a) and (c) together.*

In 2^+D pen-and-ink rendering, the first phase forwards more than just colour information into the image space. This information can be passed on as a set of n image buffers $\{IB_1, IB_2, ..., IBn\}$. The choice of information to pass on very much depends on the NPR effects one wishes to achieve, but obviously the more information, the more flexibility and control can be exercised in image space. In our 2^+D rendering system we forward the following set of image buffers into image space:

- IB_{shade}: A greyscale image synthesized using a PR shading method as shown in Figure 5(b), where any light source or sources can be introduced as required.
- IB_{dist}: A buffer containing the distances from the image plane, following the individual projection lines, to the visible part of a specific iso-surface in the dataset. Figure 5(a) shows a visualisation of this IB where the distances are mapped to a grey-scale range of $0 - 255$ and the background designated as not hitting the surface in the dataset is coloured red for clarity.
- IB_{norm}: A buffer that forwards, for each pixel, a vector represents the normal estimated for the corresponding point on the visible surface. In Figure 5(c), IB is visualized as the angles between the surface normal and the view plane normal. An angle $\in [0, \pi/2]$ maps onto an intensity $\in [0, 255]$, and red once again represents the background.
- IB_{curve}: Actually split into two IBs, namely IB_{hcurve} (Figure 5(e)) and IB_{vcurve} (Figure 5(d)), which contain the curvature of the surface in the horizontal and vertical directions respectively. In this case, curvature is taken as the rate of change of normals across the surface.

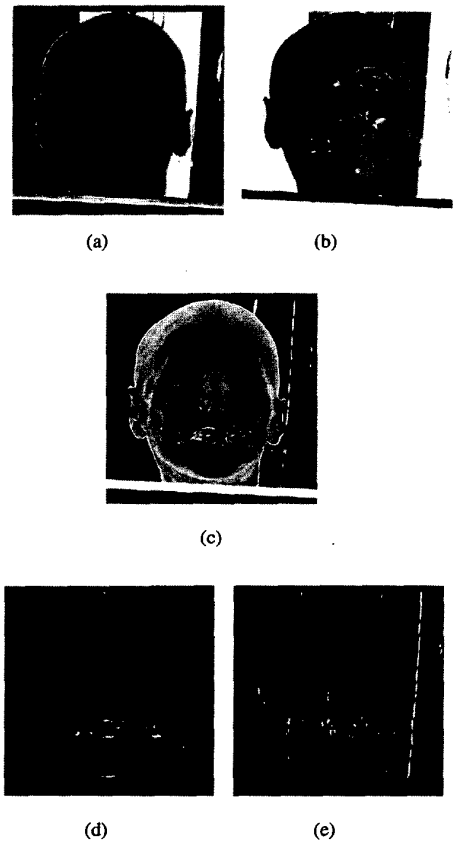


Figure 5: Control image buffers containing (a) distance, (b) shade, (c) normals, (d) vertical curvature and (e) horizontal curvature.

Having a set of IBs containing the necessary spatial information, the second phase renderer can apply different pen-and-ink filters to synthesise an NPR image. For example, an outline filter

$$f_{outline}(IB_{dist}) \rightarrow IM_1$$

determines the pixel values in IM_1 according to the values in IB_{dist}. If a pixel neighbours others of significantly different distance values, it is interpreted as being on the outline, and a short, oriented pen stroke is added in the direction of this section of the outline. Figure 6(a) shows an

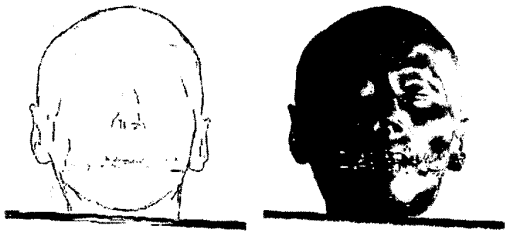
outline image generated based on the information contained in Figure 5(a).

A more complicated filter

$$f_{shade}(IB_{shade}, IB_{dist}, IB_{norm}, IB_{curve}) \rightarrow IM_2$$

is used to generate sketch drawings inside the boundary, as shown in Figure 6(b), taking into a variety of information forwarded by the first phase rendering. f_{shade} acts by the drawing of short lines emanating from randomly sampled points in the IB set. IB_{shade} enables us to make the decision of whether a line is needed at each pixel in the image plane, IB_{dist} is used to ensure lines do not traverse more than one surface, IB_{curve} allows us to decide which direction a stroke should be drawn in and IB_{norm} constrains the length of strokes.

A simple filter is then invoked, which combines two images, IM_1 and IM_2, into a final image as shown in Figure 6(c).



(a) (b)

(c)

Figure 6: *The application of 2^+D pen-and-ink rendering to a CT dataset: (a) the outline, (b) the interior shading and (c) the final image.*

4 Integration with Photorealistic Rendering

Both pen-and-ink rendering techniques discussed above were implemented in a volume-based pipeline using ray casting, with a focus on the evaluation of single or multiple iso-surfaces. As the pipeline also incorporates traditional PR rendering techniques, it intrinsically facilitates the integration between PR and NPR effects.

Figure 7: Using pen-and-ink rendering to display the skin around the skull gives good visual cues to its thickness.

One of our approaches is to apply PR and NPR rendering to different iso-surfaces as shown in Figure 7, where the bone surface is rendered using traditional PR effects while the skin surface is rendered in the same way as in Figure 3(a). However, the main algorithmic difference between the two figures is that the skin surface in Figure 7 does not have a uniform opacity, its opacity in fact varying according to the pen-and-ink texture. In this particular example, we have

$$opacity = t_v = f_1(p, t_d, t_t, t_n),$$

which transforms the surface to a net full of holes. In traditional volume visualisation, we often visualise multiple isosurfaces by making the outer surface transparent as in Figure 1(b). Our integrated method provides an alternative mechanism for such needs. Comparing Figure 7 with Figure 1(b), the former shows more visual cues to the relationship between the skin and bone, giving a greater insight into the variations of skin depth around the head, where the latter effectively allows the translucent outer surface to obscure the real colour of the skull.

We can also introduce some minimal opacity to a surface rendered with a pen-and-ink texture. As shown in Figure 8(a), the pen-and-ink effects on the outer surface provide necessary 3D visual cues to an otherwise constant sphere in terms of both colour and opacity. On the other hand, many of us would experience a degree of uncertainty about the actual geometry when given a visualisation similar to the one shown in Figure 8(b). The merits of combined PR and NPR effects are clearly demonstrated in Figures 8(c) and 8(d) where more than two nested surfaces are considered.

We can also assign different surface attributes to different volume objects as seen in Figure 9. One application of pen-and-ink visualisation is to convey to the viewer a less finished, or less real, appearance for the object in question. Figure 9(a) shows a visualisation of an actual head with the introduction of a new object, possibly a schematic for intended surgery or such like. Varying rendering styles through a visualisation can also be used to direct attention as seen in Figure 9(b) where the attention is immediately drawn to the rod rather than the head, as it is in Figure 9(a). In the case of Figure 9(c) emphasis is very clearly given to the photorealistic brain. The pen-and-ink rendering is used

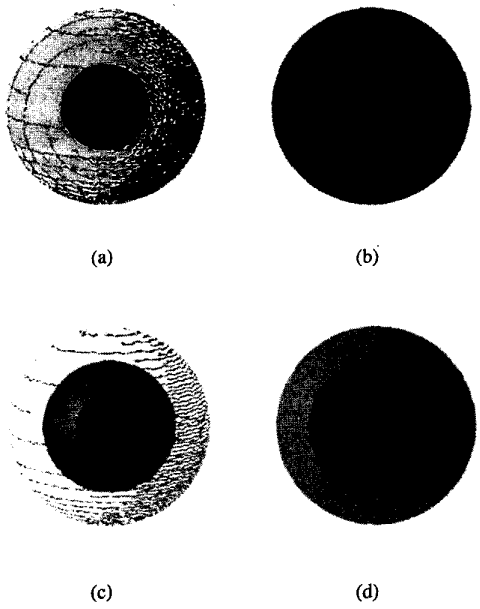


Figure 8: *NPR gives better visual cues for concentric spheres.*

to aid visualisation of the brain by showing the relative geometry of the face. This brings the brain in context without distracting the viewer too much from the intended focus of the visualisation.

5 Other Results and Discussion

One of the great strengths of pen-and-ink visualisation is that it can be used to leave out details that may be considered a little too much or even 'gloss' over missing details in an initial presentation. Our techniques can be applied to any volume dataset and we have produced extremly powerful images from a number of sources. One major source of volume data is medical machinery output very often showing body parts such as skulls, brains, or other internal organs. Although a photorealistic visualisation may be of great benefit to a surgeon in diagnosing what problems exist and deciding what course of action to take it may sometimes be desirable to present this data to a patient or their relatives by means of explanation. In this case a photorealistic visualisation may be too vivid and shocking whereas a pen-and-ink rendering could be used to as a less 'gory' visual aid. Figure 10 shows two examples of non-photorealistic medical data visualisation: Figure 10(a) being a monkey's skull and Figure 10(b) being the skull from the CT data set shown in previous images in this paper (e.g. Figures 5 and 6).

As mentioned above another use for pen-and-ink visualisations could be as an initial presentation of a preliminary design to a client. In the real world it is often the case that expensive CAD tools may be laid aside in favour of an artist's hand at the initial stages of the design process. Figure 11 shows a 'sketch' of a plane, in this case generated from a volume dataset, although presentable to a client as a designer's first ideas.

The visualisation can be further enhanced with animations, for which the 3D pen-and-ink techniques ensure the coherence between consecutive frames. Figure 12 shows two sequences of frames extracted from an animation with varying lighting and camera positions respectively.

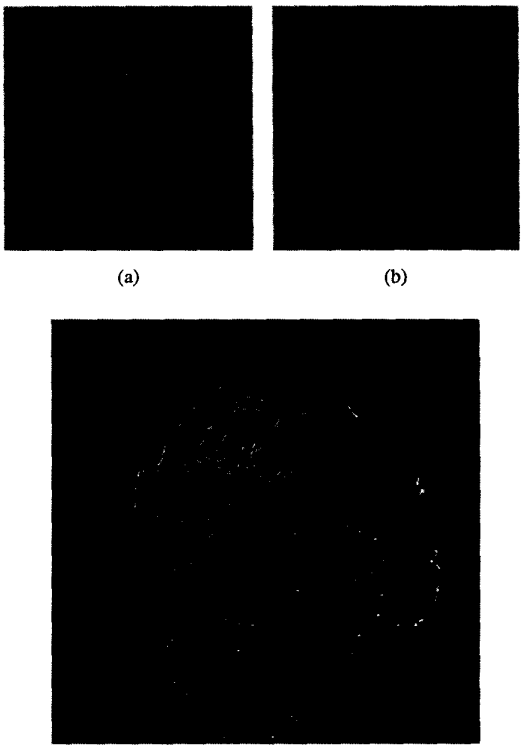


Figure 9: *Non-photorealism can be integrated with photorealism to provcide certain emphasis and/or direct the viewer's attention.*

6 Conclusions

There is no doubt that NPR techniques in general, and pen-and-ink effects in particular, are effective tools for graphical illustrations. In this paper, we have shown that pen-and-ink rendering techniques can also be implemented in a volume-based graphics pipeline, which facilitates a consistent representation of NPR textures as scalar fields, an intrinsic integration with traditional PR techniques, and a rich collection of information for image space pen-and-ink rendering. The 2^{+}D method discussed in this paper provides pen-and-ink drawings close to human hand painting in both style and quality with very little user interaction- the same parameter set was used for all the images shown in this paper. For visualisation, the 3D method, in conjunction with PR rendering, can also be used to enhance 3D visual cues for translucent surfaces, and to highlight or de-emphasise objects. In many scenarios, such combined techniques can be more effective than the use of PR techniques on their own.

To follow this work, we intend to conduct a human-factor study into the effectiveness of pen-and-ink rendering techniques in volume visualisation.

References

[1] W. Lorensen and H. Cline, "Marching cubes: a high resolution 3d surface construction algorithm," *Proc. SIGGRAPH '87*, vol. 21, no. 4, pp. 163–169, July 1987.

[2] M. Levoy, "Display of surfaces from volume data," *IEEE Computer Graphics and Applications*, vol. 8, no. 3, pp. 29–37, May 1988.

[3] L. Westover, "Footprint evaluation for volume rendering," *Proc. SIGGRAPH '90*, vol. 24, no. 4, pp. 367–376, Aug. 1990.

[4] D. Laidlaw (Organiser), "Art and visulaization: Oil and water?," in *IEEE Visualisation*, 1998, pp. 507–509, Panel Session.

[5] A. Kaufman, D. Cohen, and R. Yagel, "Volume graphics," *IEEE Computer*, vol. 26, no. 7, pp. 51–64, 1993.

[6] M. Chen, A. Kaufman, and R. Yagel, Eds., *Volume Graphics*, Springer, London, 2000.

[7] V. Interrante, H. Fuchs, and S. Pizer, "Illustrating transparent surfaces with curvature-directed strokes," *Visulaisation '96*, pp. 211–218, 1996.

[8] W. Luzadder and J. Duff, *Fundamentals of Engineering Drawing*, Prentice-Hall International Editions, 1989.

[9] J. Schumann, T. Strothotte, A. Raab, and S. Laser, "Assessing the effect of non-photorealistic rendered images in cad," *CHI 96*, pp. 35–41, Apr. 1996.

[10] P. Haeberli, "Paint by numbers: Abstract image representation," *Proc. SIGGRAPH '90*, vol. 24, no. 4, pp. 207–214, 1990.

[11] J. Lansdown and S. Schofield, "Expressive rendering: A review of nonphotorealistic techniques," *IEEE Computer Graphics and Applications*, pp. 29–37, May 1995.

[12] M. Salisbury, S. Anderson, R. Barzel, and D. Salesin, "Interactive pen-and-ink illustration," *Proc. SIGGRAPH '94*, vol. 28, no. 2, pp. 101–108, 1994.

[13] M. Sousa and J. Buchanan, "Computer-generated graphite pencil rendering of polygonal models," *Proceedings of Eurographics*, vol. 19, no. 3, 1999.

[14] C. Curtis, S. Anderson, J. Seims, K. Fleischer, and D. Salesin, "Computer-generated watercolour," *Proc. SIGGRAPH '97*, 1997.

[15] S. Strassman, "Hairy brushes," *Proc. SIGGRAPH '86*, pp. 225–232, 1986.

[16] B. Pham, "Expressive brush strokes," *Computer Vision, Graphics, and Image Processing*, vol. 53, no. 1, pp. 1–6, 1991.

[17] A. Hertzmann, "Painterly rendering with curved brush strokes of multiple sizes," *Proc. of SIGGRAPH '98*, pp. 453–460, July 1998.

[18] R. Bartels, J. Beaty, and B. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*, Morgan-Kaufmann, 1987.

[19] C. De Boor, "On calculating with b-splines," *Journal of Approx Theory*, vol. 6, pp. 50–62, 1972.

[20] E. Catmull, "A hidden surface algorithm with anti-aliasing," *Proc. SIGGRAPH '78*, pp. 6–11, 1978.

[21] G. Winkenbach and D. Salesin, "Computer-generated pen-and-ink illustration," *Proc. SIGGRAPH '94*, vol. 28, no. 2, pp. 91–100, 1994.

[22] G. Wikenbach and D. Salesin, "Rendering parametric surfaces in pen and ink," *Proc. SIGGRAPH '96*, pp. 469–476, 1996.

[23] G. Elber, "Interactive line art rendering of freeform surfaces," *Proceedings of Eurographics*, vol. 18, no. 3, 1999.

[24] J. Buchanan, "Special effects with half-toning," *Computer Graphics Forum*, vol. 15, no. 3, pp. 97–108, 1996.

[25] O. Veryovka and J. Buchanan, "Comprehensive halftoning of 3d scenes," *Proceedings of Eurographics*, vol. 18, no. 3, 1999.

[26] M. Salisbury, M. Wong, J. Huges, and D. Salesin, "Orientable textures for image-based pen-and-ink illustration," *Proc. SIGGRAPH '97*, 1997.

[27] M. Salisbury, C. Anderson, D. Lischinski, and D. Salesin, "Scal-dependent reproduction of pen-and-ink illustrations," *Proc. SIGGRAPH '96*, pp. 461–468, 1996.

[28] M. Sousa and J. Buchanan, "Observational model of blenders and erasers in computer-generated pencil rendering," *Proceedings of Graphics Interfaces '99E*, 1999.

[29] J. Hamel and T. Strothotte, "Capturing and reusing rendition styles for non-photorealistic rendering," *Proceedings of Eurographics*, vol. 18, no. 3, 1999.

[30] T. Saito and T. Takahashi, "Comprehensible rendering of 3d shapes," *Proc. SIGGRAPH '90*, vol. 24, no. 4, pp. 197–206, 1990.

[31] M. Chen, J. Tucker, and A. Leu, "CROVE - a rendering system for constructive representations of volume environments," in *International Workshop on Volume Graphics*, Mar. 1999, pp. 275–294.

[32] D. Peachy, "Solid texturing of complex surfaces," *Proc. SIGGRAPH '85*, vol. 19, no. 3, pp. 279–286, July 1985.

[33] K. Perlin, "An image synthesiser," *Proc. SIGGRAPH '85*, vol. 19, no. 3, pp. 287–296, July 1985.

[34] M. Šrámek and A. Kaufman, "Object voxelization by filtering," in *IEEE Symposium on Volume Visualization*, 1998, pp. 111–118.

[35] M. Šrámek and A. Kaufman, "`vxt`: a c++ class library for object voxelization," in *International Workshop on Volume Graphics*, 1999, pp. 295–306.

(a)

(b)

Figure 10: *The technique can be used to visualise different volumes in the same style.*

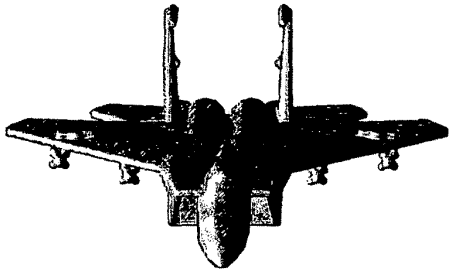


Figure 11: *A 'preliminary sketch' of an aircraft design. (generated from a plane data set used by kind permission of Arie Kaufman [34, 35]).*

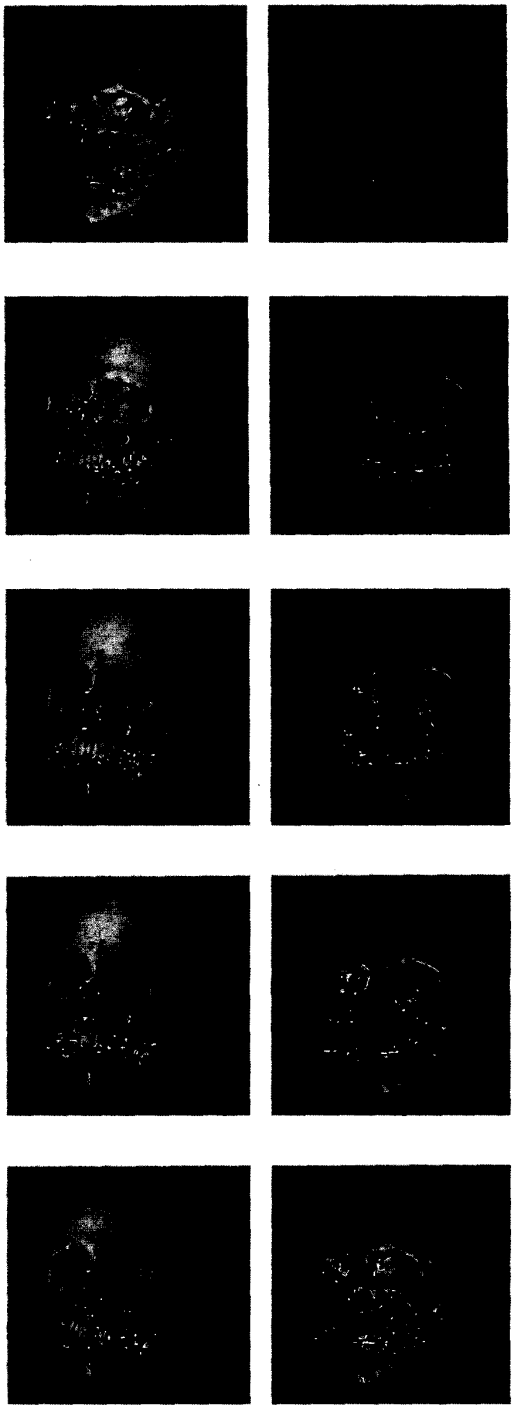


Figure 12: *Exerts from two animation sequences: (left) lightsource animation, (right) viewpoint animation.*