

Hardware Accelerated Real Time Charcoal Rendering

Aditi Majumder
Department of Computer Science
University of North Carolina at Chapel Hill
majumder@cs.unc.edu

M. Gopi
Information and Computer Science
University of California, Irvine.
gopi@ics.uci.edu

Abstract

In this paper, we present simple rendering techniques implemented using traditional graphics hardware to achieve the effects of charcoal drawing. The effects include characteristics of charcoal drawings like broad grainy strokes and smooth tonal variations that are achieved by smudging the charcoal by hand. Further, we also generate the *closure effect* that is used by artists at times to avoid hard silhouette edges. All these effects are achieved using *contrast enhancement operators* on textures and/or colors of the 3D model.

Our contribution lies in unifying the methods to achieve these effects under the common framework of contrast enhancement operators. Further, since the effects have been implemented using traditional graphics hardware, a *single rendering pass* is sufficient to create different effects. Hence, we can render highly complex models with large number of triangles at interactive rates. Thus, our method is especially suited for applications like scientific visualization and preliminary sketches/animations.

CR Categories: 1.3.5 [Non Photorealistic Rendering]: Hardware Accelerated Rendering—Charcoal Rendering

Keywords: Non Photorealistic Rendering, Hardware Accelerated Rendering, Real time rendering, Charcoal Rendering

1 Introduction

In many applications – from medical texts to architectural design and mechanical illustrations – non-photorealistic rendering is often more effective than photorealism. Such illustrations perform an abstraction on the actual scene content and convey information that is most important to an observer. Further, recent research has shown that such illustrations can be easily produced, consume less storage and thus can be a better choice than photorealism in many applications.

Research in non-photorealistic rendering (NPR) has found methods to simulate the effects of various traditional *media* like pen and ink [4, 27, 28, 34, 35, 19], pencil sketches [1], water color [3] and engraving [21, 30]. There has been work on rendering different styles like impressionism [8, 10, 15, 18] and technical illustrations [6, 26, 25]. There has also been work on non-photorealistic rendering of specific objects like fur and grass [13, 14]

To the best of our knowledge, except for an interactive 2D system developed in 1988 [2], we have not seen any work specifically on computer generated 3D charcoal rendering. However, there has been some nice work on graphite pencil rendering and modeling blenders and erasers in the past [32, 33, 31]. This work achieves realistic graphite effects by using detailed models of pencils and erasers. As a result, the algorithm is non-interactive and takes many seconds to render even small models composed of a few hundred triangles. We can envision situations as in developing animation movies or visualizing large scientific data, where the user may want to get a quick look at the data from different viewpoints even before

deciding on more detailed and elaborate rendering. Further, there is also a need for interactive system for applications like artistic virtual environments. Our technique uses traditional graphics hardware to implement all the techniques and thus, is more suited for such purposes.

Our method is based on the following interesting observation about the charcoal drawings. Charcoal is extremely limited in dynamic range when compared to the dynamic range of the physical world, or even other digital and non-digital media. So, it is indeed difficult to capture many rich colors and textures using charcoal. Yet, an artist does it with amazing ease. One of the very common techniques that artists use to overcome this limitation is to exaggerate the contrast difference. A related observation was made in [6] while designing lighting model for non-photorealistic colored rendering of mechanical illustrations. This system used luminance and hue shifts to create such lighting effects. We also use contrast enhancement while creating charcoal rendering effects. We achieve this by using a contrast modifying function which we call the **contrast enhancement operator (CEO)**. Similar techniques have also been used in [32, 31, 29, 27, 28, 34, 35] where the problem is viewed as a special kind of half-toning. The contribution of our technique lies in the fact that we achieve many different effects of charcoal rendering using the common framework of contrast enhancement.

The enhancement operator operates on a uniform noise texture to generate a **contrast enhanced texture** which is texture mapped on the model to create the grainy effect of charcoal rendering. Many previous algorithms have also used texture mapping for interactive NPR. These include *Tonal Art Map* described by Praun *et. al* [23], *prioritized stroke textures* used by Winkenback [34] and Salisbury [27] and *Art Maps* used by Klein [12]. In the former two systems, one texture is used to convey a single tone. Thus, many such textures have to be blended together in varying proportions to create the varying tone of a polygon [23]. This may lead to renderings that need multiple passes. We achieve the tonal variation by using *only one* CET. Thus we do not need multi-texturing or blending to achieve the effect of charcoal rendering. Though art maps use a single texture, this texture is dependent on the scene content whereas our texture is independent of the scene content.

The contrast enhancement operator also operates on the color of the Phong shaded vertices of a model to generate what we call a **contrast enhanced model**. The color at vertices of the contrast enhanced model is used to index into the contrast enhanced texture for the final texture mapping. To create the effect of smudging the charcoal by hand, this texture mapped model is blended with the contrast enhanced model.

Finally, we create what artists call the *closure effect* where the silhouettes are not explicitly rendered, and the closure of the object is left to the interpretation of the viewer. There have been several methods that use traditional graphics hardware to achieve real time silhouettes [5, 7, 9, 16, 20, 24] and artistic silhouettes [20]. However, rendering the texture mapped model achieves the closure



Figure 1: Conventional Charcoal Drawings(From top to bottom and left to right): (a) The abstract Fitting (b) The Musician (c) Lady on Horse (d) Sleeping Dog (e) Indian Woman (f) The Girl

effect without an explicit computation of silhouettes.

1.1 Main Contributions

In summary, the following are the main contributions of this paper:

- **Contrast Enhancement Framework:** We achieve many different effects of charcoal rendering using a common framework of contrast enhancement.
- **Hardware Assisted Contrast Enhancement:** We present a new way of using the conventional graphics hardware that is capable of doing Phong shading, to achieve a contrast-enhanced smoothly shaded model.
- **One Contrast Enhanced Texture:** The contrast enhanced texture captures the information of different tones in a single texture and is obtained by applying the enhancement operator on a noise texture that conveys a uniform tone.
- **Hardware Accelerated One Pass Real Time Rendering System :** We present a single pass rendering algorithm that uses several features of traditional graphics pipeline to implement all these techniques to generate a real-time charcoal rendering of 3D models.

In Section 2, we describe in details some of the effects of charcoal rendering that we simulate in our system. In Section 3, we present the algorithm overview. In Section 4, we describe our implementation using the traditional graphics hardware. Finally, we present the results in Section 5 and conclude with future work in Section 6.

2 Charcoal Rendering

It is impossible to capture an artist's creativity, rendering style and individuality using computer generated techniques. However, there are some basic rules that all artists use extensively that we can strive to achieve. In this section, we describe a few such basic techniques that artists follow while using charcoal as a medium.

2.1 Contrast Enhancement

As mentioned before, artists extensively use contrast enhancement to overcome the limited dynamic range of the charcoal medium. This helps the artist to make the brightness differences more pronounced in order to achieve a similar perception of brightness as the real world within the limited dynamic range of the medium. This technique helps to create various effects like shadows and also to effectively hide details in non-illuminated parts of the scene. This contrast enhancement is illustrated in the drawing of Figure 1(a). The specular highlight is shown as a sharp white contrast from the darker sides. In the drawing in Figure 1(e), the shadows below the chin of the woman have been accentuated using the same technique and the hair on both sides of the head are darkened to hide details. Thus, contrast exaggeration has helped the artist to create a sense of volume or 3D on the 2D canvas.

2.2 Closure

Artists may sometimes like to leave some parts of the silhouette unrendered or very soft. The observer is allowed the freedom to imagine and interpret the effect. This effect is termed by the artists as *the closure*. Closure is also used to render the effects of very bright light on the object. Notice the closure on the left side of the

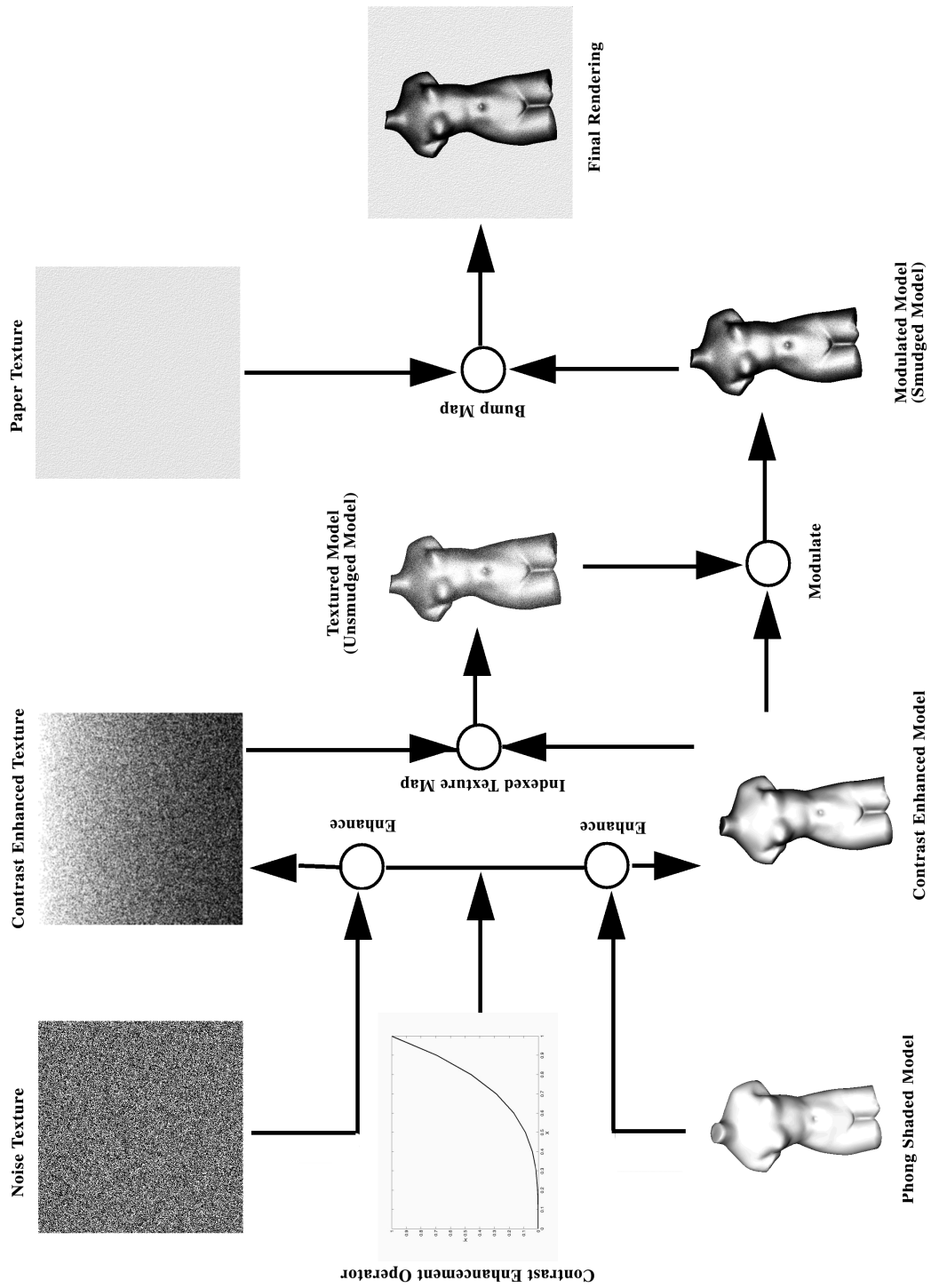


Figure 2: The Algorithm

forehead of the musician in Figure 1(b) and the left arm and leg of the lady on horse in Figure 1(c). In the picture of the sleeping dog in Figure 1(d), closure is used on the right side of the dog’s face to get the effect of bright sunlight. Also notice that this technique does not hamper our perception of the shape of the object in any way.

2.3 Smudging Effect

Artists use charcoal in different ways to create different rendering styles. Sometimes charcoal is used lightly on coarse paper. Thus, grains of charcoal left on the paper creates a grainy appearance as in the drawing of the girl in Figure 1(f). On the other hand charcoal may be pressed hard on paper to leave larger number of grains which are then smudged by hand to create the effect of varying tones of gray as in the picture of the musician, the Indian woman and the sleeping dog. The tones created likewise can be varied by using different kinds of charcoal or by varying the pressure with which the charcoal is used on the paper.

3 Algorithm Overview

In this section, we describe the design of our algorithm for charcoal rendering of 3D models at interactive rates that creates the artistic effects explained in the previous section. This algorithm uses contrast enhancement techniques to achieve these effects. These techniques are accelerated using several features of traditional graphics hardware in our implementation.

The steps in our algorithm are shown in Figure 2. We apply a contrast enhancement operator (“operator”) on a noise texture (conveying an uniform tone) to generate the charcoal grain texture, which we refer to as contrast enhanced texture (“texture”). We use the same contrast enhancement operator to modify the gray scale Phong shaded model. The model thus modified is called the contrast enhanced model (“enhanced model”). In Section 4, we will describe how we use the conventional graphics hardware to render contrast enhanced model.

We use the enhanced model for two purposes. The contrast enhanced gray value of the vertices of the enhanced model is used to index into the texture for the mapping of the grainy texture onto the original 3D model. The second purpose of this contrast enhanced model is to create the smudging effect that is achieved by blending the textured model with the enhanced model. Finally, we bump map a paper-texture in 2D on the rendered frame to impart the coarseness of the background paper.

3.1 Creation of Contrast Enhancement Operator, Texture, and Model

In this section, we define contrast enhancement operators, contrast enhanced textures and enhanced models.

3.1.1 Contrast Enhancement Operators

A contrast enhancement operator C works on a gray scale texture T or a gray shaded model M and modifies them in such a way so as to increase the perceived contrast. Thus, C is a function that maps the gray value x , $0.0 \leq x \leq 1.0$, to another gray value y , $0.0 \leq y \leq 1.0$. For example, $y = x^\gamma$ is a contrast enhancement function for $\gamma > 1.0$. A set of such contrast enhancement operators are shown in Figure 3 which vary in the value of γ .

3.1.2 Contrast Enhanced Texture

In this step, we apply a contrast enhanced operator C on a noise texture which conveys the appearance of a uniform grainy tone. The

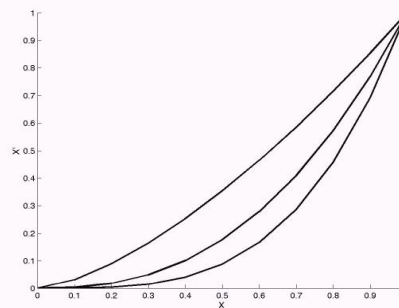


Figure 3: Family of contrast enhancement curves of the form $y = x^\gamma$. Here we show the curves with γ equal to 1.5, 2.5 and 3.5.

contrast enhanced texture thus generated helps us to convey the information of several tones in the same texture.

A noise texture is usually generated by placing black grains at random locations on a square white image. The density of the grains simulates the graininess of the charcoal which can vary depending on the type of charcoal and the pressure with which it is used on the paper. We can vary grain density by changing the number of black particles on the white image.

Contrast enhanced texture is generated the same way a noise texture is generated, except that we apply the contrast enhancement operator on the random y -coordinate generated for every black grain. Finally, we blur the texture to create the final contrast enhanced texture. Figure 4 shows two contrast enhancement operators and the corresponding enhanced textures of varying grain densities.

3.1.3 Contrast Enhanced Model

We apply the contrast enhancement operator on the gray values of the Phong shaded model to create the contrast enhanced model. The Phong shading gives a perception of the curvature of the surface as it is derived from the surface normal and the light position. Applying the operator on the colors of this model pronounces the shadows and curves and thus enhances the perceived sense of ‘volume’ and 3D shape.

Notice that in Figure 4 the planar textures impart a perception of 3D. They look curved and the perceived curvature varies with different operators and grain densities. This is exactly the technique artists use to create what they call a sense of ‘3D volume’. Applying the operator to the Phong shaded model imparts a similar effect. Figure 2 shows the model before and after the operator is applied.

3.2 The Grainy Stroke Effect

The effect of grainy strokes are brought about by texture mapping the contrast enhanced grainy texture on to the original 3D model. The gray values of the contrast enhanced model is used to find these texture coordinates. Figure 5 shows the effect of using the gray values of the Phong shaded model as texture coordinates instead of the gray values of the contrast enhanced model. This illustrates the importance of the operator in rendering the grainy strokes.

3.3 The Smudging Effect

The rendering at this point, with only the grainy texture on the model, lacks the effect of smudging the charcoal with hand to create smooth tonal variations. Interestingly, the blending of this textured model with the contrast enhanced model brings about this effect. Refer to Figure 2 to see this effect.

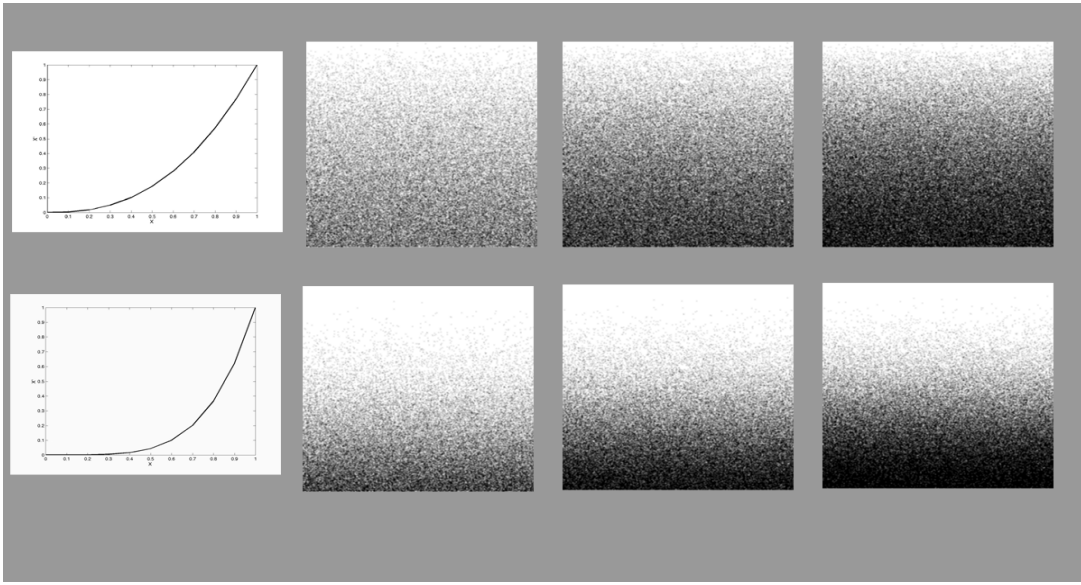


Figure 4: CETs of varying grain density generated by applying different CEOs

3.4 The Paper-Texture Effect

Finally, to create the effect of a coarse paper canvas, we bump map a paper texture on the rendered frame in 2D. The paper texture is created by embossing a noise image. Notice that, this is different from the 3D bump mapping used in [1] to create the same effect. In [1], the paper texture has to be scaled appropriately if the model is scaled, to maintain coherence. Since we use a 2D bump map on the rendered frame, the paper texture remains the same even when the model is zoomed in or out. Further, a 2D bump map simulates the paper/canvas appropriately since the coarseness of the paper does not change with the changing content of the rendering.

4 Real Time Rendering

In this section, we describe our single pass rendering technique to implement the algorithm presented in the previous section which generates charcoal rendering of 3D models on traditional graphics pipeline at interactive rates. We use n to denote the normal at a vertex and L to denote the light vector from the light position to the vertex. We use the operators of the form $C(x) = x^\gamma$, $\gamma > 1.0$. We can also specify a piecewise linear operator.

4.1 Contrast Enhanced Texture

We generate a set of contrast enhanced textures using different contrast enhancement operators and grain densities as shown in Figure 4, as a part of preprocessing. At run-time, we use *one resident texture* at a time to texture map the whole model. The textures are switched in order to change the rendering style. For example, if we want to render the effects of a softer charcoal, we use an enhanced texture of a greater grain density.

4.2 Contrast Enhanced Model

The way we create a contrast enhanced model is interesting since we use the conventional graphics hardware for this purpose and we do not explicitly compute the contrast enhancement. One way to enhance the shading of the model is to calculate the gray values at the vertices of the polygons using the Phong shading model, and

then apply the operator on these gray values to enhance the shade at each vertex. Then Gouroud shading can be employed to interpolate the values of the enhanced shading within the polygon. But the Gouroud shading introduces banding effects due to discretization and rasterization process. But the same result without the banding artifacts could be achieved using Phong shading. The following is the equation of the color we would like to achieve in the interior of the triangle, but without much of artifacts. This uses barycentric coordinates α and β , the vertex normals N_1 , N_2 , and N_3 , and the light vector (at infinity) L .

$$\alpha C(N_1 \cdot L) + \beta C(N_2 \cdot L) + (1 - \alpha - \beta) C(N_3 \cdot L) \quad (1)$$

In our application, we use no specular lighting component and hence we are concerned only about the diffused lighting. We assume a white colored model and white colored light with a coefficient of diffused lighting of 1.0. Thus, the diffused lighting at any point is exactly equal to the dot product of the normal and the light vector.

Let us try to approximate the effect of Equation 1 to some reasonable accuracy using Phong shading so that we get a continuous shading. We achieve this by scaling the normals in such a way that Equation 1 is satisfied at the vertices. (At the vertices of the triangle, one of α , β and $1 - \alpha - \beta$ is 1 and the others are 0). The scale factor thus computed is $\frac{C(n \cdot L)}{n \cdot L}$. Hence at the vertices of the triangle we get the following shading expression as required by Equation 1.

$$\left(\frac{C(n \cdot L)}{n \cdot L} n \right) \cdot L = \frac{C(n \cdot L)}{(n \cdot L)} (n \cdot L) = C(n \cdot L),$$

Then we render this model using Phong shading to achieve the gray scale enhancement within the polygon. In the interior of the triangle, we get

$$t \times (\alpha C(N_1 \cdot L) + \beta C(N_2 \cdot L) + (1 - \alpha - \beta) C(N_3 \cdot L))$$

where t is a varying scale factor to normalize the normal vector. Thus, we maintain the required results at the vertices of the triangle, and the Phong shading gives us results that are good in the interior of the triangle also.

In summary, this method helps us in two ways. First, we are able to use the hardware to achieve the contrast enhanced colors. Second, we get rid of banding artifacts caused by the Gouroud shading.

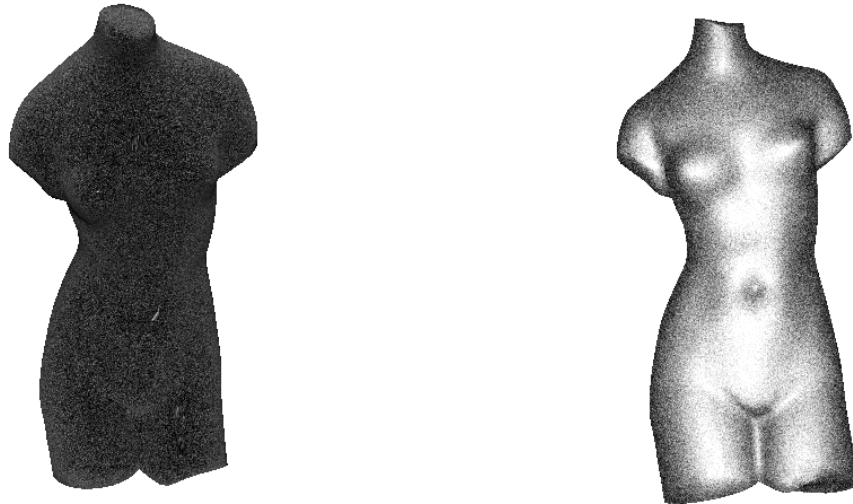


Figure 5: Left : The model, texture mapped using the Phong shade gray values as an index to the CET Right : The model, texture mapped using the gray values of the contrast enhanced model as an index into CET.

4.3 Smudging Effect

To create the *smudging*, we again use the graphics hardware capability of blending colors with texture mapped polygons. We modulate the textured model by the enhanced gray values of the enhanced model generated by the method mentioned above. This produces nice smudging effects as shown in the Figures 6 and 7.

4.4 Paper Texture

We generate paper textures of varying coarseness by embossing a noise texture and then compressing its range near the white region of the gray scale. After rendering the textured model, we use the hardware accelerated alpha blending to bump map the paper texture on the rendered frame.

With the above system implementation, we can interactively change between different operators, grain density factors, paper coarseness, and smudged or unsmudged rendering styles.

5 Results and Discussions

In Figures 6 and 7 we present a few of our results using different contrast enhancement operators, different grain density factors, and smudged and unsmudged styles. These are generated using unoptimized code on SGI O2 R10000 processor. Notice the closure effect attained on the left teapot, the hump of the left pig and the missile. The closure effect is attained when we use low grain density factors, high contrast, and have the smudging off. Also notice that we can produce different drawing styles from light sketchy drawings (left teapot, right plane) and lightly shaded drawings (missile, pig) to highly silhouetted smudged drawings (bunny, venus and others). Further, we can handle models with complex curves and bends, starting from 1600 polygons (teapot) to 65000 polygons (bunny) at interactive rates of 30 frames per second.

We have ported our algorithm on to regular PCs having NVidia Geforce graphics cards. Figure 8 shows our results on PC when using our techniques on large scientific data of thousands of polygons. The models shown here have more than one million triangles each.

We have also investigated using different contrast enhancement operators for generating the enhanced texture and the contrast enhanced model. We have used a gamma function as mentioned in Section 3 to create the enhanced noise texture while applying a user-defined piecewise linear operator to generate the contrast enhanced model. Once these two models are blended, the result creates a different style of charcoal rendering as illustrated by the left Venus model of Figure 6 and the head in Figure 7.

In the accompanying video we illustrate the interactive rendering of various models. Since we use a single noise texture, we do not see any temporal popping artifacts while zooming in and out. Since our enhanced textures are uniform in one dimension and not in the other, we see some spatial discontinuities when we zoom very close to the model. We are investigating if we can use ‘lapped texture’ [22, 23] to achieve better spatial coherency. We are also investigating geograftal/particle techniques [11, 17] to avoid any temporal artifacts whatsoever.

6 Summary and Future Work

In summary, we presented an algorithm that generates different effects of charcoal drawing of 3D models using the common framework of contrast enhancement. The contrast enhancement operators can be applied on textures and colors to achieve non-photorealistic effects. This also helped us to achieve the closure effect. Further, we implemented this algorithm as a single pass rendering technique to render charcoal effects at interactive rates using available features in the traditional graphics pipeline. We hope to continue this research in the following directions.

Different Contrast Enhancing Operators: We believe that contrast enhancement operators are powerful enough to generate different NPR effects. For example, a step function contrast enhancement operator can be used to generate cartoon renderings similar to those described in [1]. It would be worth determining if several NPR effects can be unified under the umbrella of the contrast enhancement.

More Sketch Strokes: Our charcoal rendering lacks rich strokes like the ones in the drawing of the girl in Figure 1. It is interesting to ponder how we might use different textures (instead of noise

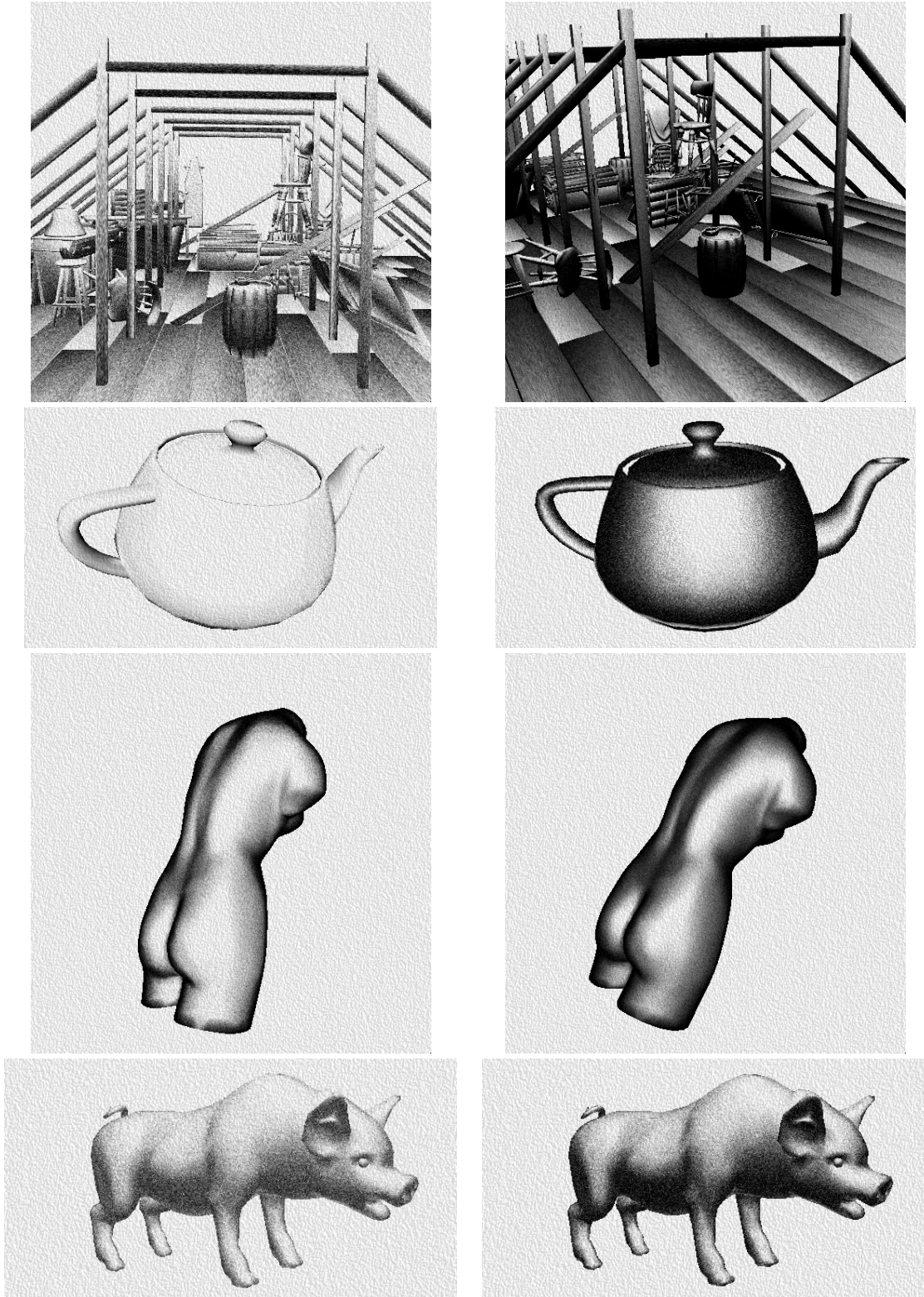


Figure 6: Results (Row wise from top to bottom): (a) The attic model (50,000 triangles) rendered with and without smudging. (b) The Teapot (1600 triangles) rendered using two different contrast enhancement operators. (c) The Venus Model (6000 polygons) rendered using two different contrast enhancement operators. (d) Babe, the pig (3000 polygons) rendered using two different grain density factors.

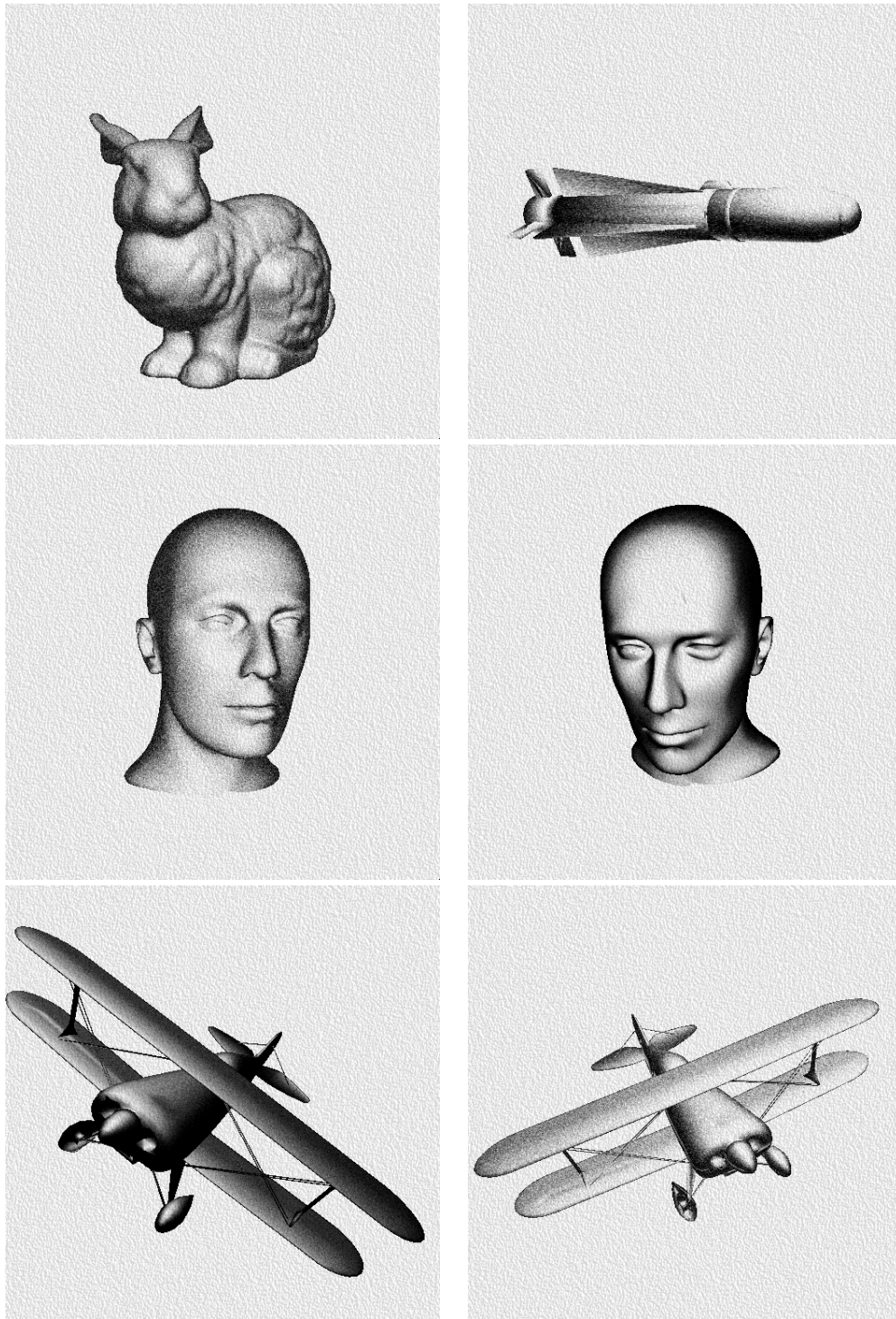


Figure 7: Results (Left to Right and Top to Bottom): (a) The Stanford Bunny (65000 polygons) (b) The Missile (3000 polygons) (c)The Face (13000 polygons) with smudging off (d) The Face with smudging on (e)The Plane (9000 polygons) with smudging on (f) The Plane with smudging off

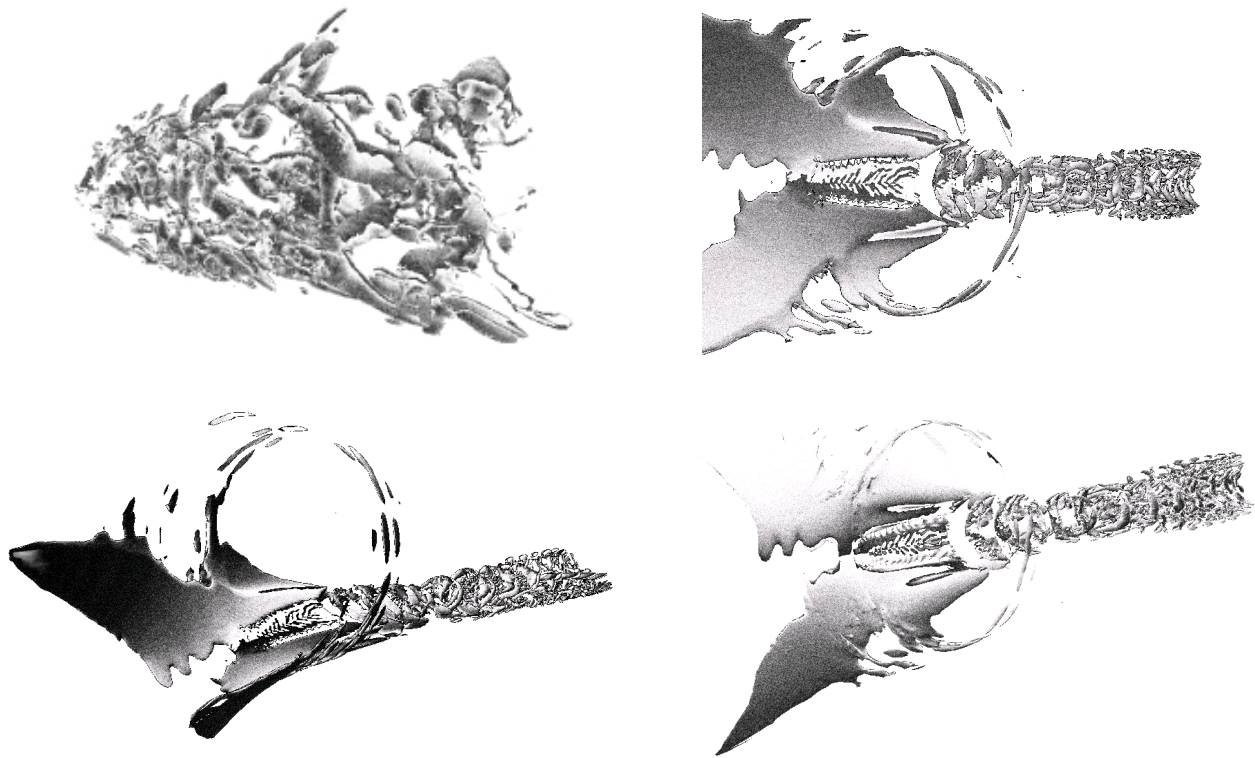


Figure 8: Visualization of large complex scientific model using charcoal rendering

texture) to create a stroke texture and use it in addition to the grain texture to create richer strokes.

Temporal and Spatial Coherence: We would like to investigate the temporal and spatial coherency of the interactive charcoal rendering in detail so that the algorithm can be used for animations.

Radiating Silhouettes: A rendering style quite commonly used by artists is to draw radiating strokes moving away from the object silhouette in the background as in Figure 1(a). It requires human creativity and aesthetics to use this style effectively. However, the question is, can we find a way to create this effect using computers?

Different styles in the same drawing: Artists also use different styles in the same drawing. For example, in the drawing of the Indian Woman in Figure 1e, some parts of the drawing are smudged by hand while others are not. Though an artist uses his creativity to decide on this, we would like to find an automatic way to switch between styles within the same frame to create artistically vibrant pictures.

Acknowledgements: We acknowledge Lauren Kroiz and Michael Papka of Argonne National Laboratories for using our algorithm on large scientific data and sharing the results with us.

References

- [1] A.Lake, C.Marshall, M.Harris, and M.Blackstein. Stylized Rendering Techniques For Scalable Real-Time 3D Animations. In *Non-Photorealistic Animation and Rendering*, 2000.
- [2] Teresa W. Bleser, John L. Sibert, and J. Patrick McGee. Charcoal sketching: Returning control to the artist. In *ACM Transactions on Graphics, Vol 7, No 1*, Jan 1988.
- [3] Cassidy J. Curtis, Sean E. Anderson, Joshua E. Seims, Kurt W. Fleischer, and David H. Salesin. Computer-Generated Watercolor. In *SIGGRAPH 97 Conference Proceedings*, pages 421–430, August 1997.
- [4] O. Deussen and T. Strothotte. Computer generated pen and ink illustration of trees. In *ACM Siggraph 2000 Conference Proceedings*, Aug 2000.
- [5] Gershon Elber. Interactive Line Art Illustration of Freeform Surfaces. In *Computer Graphics Forum 18*, pages 1–12, Sep 1999.
- [6] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. A Non-photorealistic Lighting Model for Automatic Technical Illustration. In *ACM Siggraph 1998 Conference Proceedings*, Jul 1998.
- [7] Bruce Gooch, Peter-Pike Sloan, Amy Gooch, Peter Shirley, and Richard Riesenfeld. Interactive Technical Illustration. In *Interactive 3D Conference Proceedings*, April 1999.
- [8] Paul E. Haeberli. Paint By Numbers: Abstract Image Representations. In *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 207–214, August 1990.
- [9] A. Hertzmann and D. Zorin. Illustrating smooth surfaces. In *SIGGRAPH 2000 Conference Proceedings*, pages 517–526, July 2000.
- [10] Aaron Hertzmann. Painterly Rendering with Curved Brush Strokes of Multiple Sizes. In *ACM SIGGRAPH 98 Conference Proceedings*, pages 453–460, July 1998.
- [11] M. Kaplan, B. Gooch, and E. Cohen. Interactive artistic rendering. In *ACM Siggraph 2000 Proceedings*, July 2000.

- [12] A. Klein, W. Li, M. Kazhdan, W. Corrêa, A. Finkelstein, and T. Funkhouser. Non-photorealistic virtual environments. In *ACM Siggraph 2000 Conference Proceedings*, Aug 2000.
- [13] Michael A. Kowalski and Lee Markosian et al. Art-Based Rendering of Fur, Grass, and Trees. In *SIGGRAPH 99 Conference Proceedings*, August 1999.
- [14] Jerome E. Lengyel, Emil Praun, Adam Finkelstein, and Hugues Hoppe. Real-time fur over arbitrary surfaces. In *ACM Symposium on Interactive 3D Graphics*, March 2001.
- [15] Peter Litwinowicz. Processing Images and Video for an Impressionist Effect. In *SIGGRAPH 97 Conference Proceedings*, pages 407–414, August 1997.
- [16] Lee Markosian, Michael A. Kowalski, Samuel J. Trychin, Lubomir D. Bourdev, Daniel Goldstein, and John F. Hughes. Real-Time Nonphotorealistic Rendering. In *SIGGRAPH 97 Conference Proceedings*, pages 415–420, August 1997.
- [17] Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Lorin S. Holden, J.D. Northrup, and John F. Hughes. Art-based rendering with continuous levels of detail. In *Non-Photorealistic Animation and Rendering 2000 (NPAR '00)*, Annecy, France, June 5-7,2000.
- [18] Barbara J. Meier. Painterly Rendering for Animation. In *SIGGRAPH 96 Conference Proceedings*, pages 477–484, August 1996.
- [19] Alex Mohr and Michael Gleicher. Non-invasive and interactive, stylized rendering. In *ACM Symposium on Interactive 3D Graphics*, March 2001.
- [20] J. D. Northrup and L. Markosian. Artistic Silhouettes. In *Non-Photorealistic Animation and Rendering*, pages 31–38, 2000.
- [21] V. Ostromoukhov. Digital facial engraving. In *SIGGRAPH 1999 Conference Proceedings*, pages 417–424, August 1999.
- [22] Emil Praun, Adam Finkelstein, and Hugues Hoppe. Lapped textures. In *ACM Siggraph 2000 Proceedings*, July 2000.
- [23] Emil Praun, Hugues Hoppe, Matthew Webb, and Adam Finkelstein. Real-time hatching. In *ACM Siggraph 2001 Conference Proceedings*, pages 581–586, Aug 2001.
- [24] R. Raskar and M. Cohen. Image precision silhouette edges. In *ACM Symposium on Interactive 3D Graphics*, pages 135–140, 1999.
- [25] C. Rössl and L. Kobbelt. Line art rendering. In *Proceedings of Pacific Graphics*, 2000.
- [26] Takafumi Saito and Tokiichiro Takahashi. Comprehensible Rendering of 3-D Shapes. In *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 197–206, August 1990.
- [27] M. P. Salisbury, S. E. Anderson, R. Barzel, and D. H. Salesin. Interactive pen and ink illustrations. In *ACM Siggraph 1994 Conference Proceedings*, pages 101–108, Aug 1994.
- [28] M. P. Salisbury, M. T. Wong, J. E. Hughes, and D. H. Salesin. Orientable textures for image-based pen-and-ink illustration. In *ACM Siggraph 1997 Conference Proceedings*, pages 401–406, Aug 1997.
- [29] Mike Salisbury, Corin Anderson, Dani Lischinski, and David H. Salesin. Scale-Dependent Reproduction of Pen-and-Ink Illustrations. In *SIGGRAPH 96 Conference Proceedings*, pages 461–468, August 1996.
- [30] A. Sourin. Functionally based virtual computer art. In *Interactive 3D Conference Proceedings*, March 2001.
- [31] Mario Costa Sousa and John W. Buchanan. Observational model of blenders and erasers in computer-generated pencil rendering. *Graphics Interface '99*, pages 157–166, June 1999.
- [32] Mario Costa Sousa and John W. Buchanan. Observational models of graphite pencil materials. *Computer Graphics Forum*, 19(1):27–49, March 2000.
- [33] Mario Costa Sousa and John W. Buchanan. Computer-generated graphite pencil rendering of 3d polygonal models. *Computer Graphics Forum*, 18(3):195–208, September 1999.
- [34] Georges Winkenbach and David H. Salesin. Computer-Generated Pen-And-Ink Illustration. In *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, pages 91–100, July 1994.
- [35] Georges Winkenbach and David H. Salesin. Rendering Parametric Surfaces in Pen and Ink. In *SIGGRAPH 96 Conference Proceedings*, pages 469–476, August 1996.