

Computer-Generated Pencil Drawing

Mario Costa Sousa and John W. Buchanan [†]

Department of Computing Science
University of Alberta, Edmonton, Alberta, Canada
{mario.juancho}@cs.ualberta.ca

[†] Also Research Scientist at Electronic Arts Inc.
Burnaby, British Columbia, Canada
juancho@ea.com



Figure 1: A 3D pencil model rendered with our model.

Abstract

In this paper we give an overview of our research on non-photorealistic rendering methods for computer-generated pencil drawing. Our approach to the problem of simulating pencil drawings was to break it down into the subproblems of (1) simulating first the drawing materials (graphite pencil and drawing paper, blenders and kneaded eraser), (2) developing drawing primitives (individual pencil strokes and mark-making to create tones and textures), (3) simulating the basic rendering techniques (outlining and shading of 3D models) used by artists and illustrators familiar with pencil rendering, and (4) implementing the control of drawing steps from preparatory sketches to finished rendering results. We demonstrate the capabilities of our approach with a variety of images generated from reference images and 3D models.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms; I.6.3 [Simulation and Modeling]: Applications—

Keywords: Non-Photorealistic Rendering, natural media simulation, pencil rendering, 3D rendering, illustration systems.

1 Introduction

The display of models using highly realistic illumination models has driven much of the research in computer graphics. Research in non-photorealistic rendering (NPR) seeks to provide alternative display methods for 3D models or reference images. Existing research works in NPR can be classified using four main criteria which are derived from the elements for visual composition:

1. **Subject:** What is the input source for drawing/painting? This can be free composition, a reference image (information from 2D digitized images or 3D rendered objects), or a direct rendering from 3D models.
2. **Materials:** What media is simulated? This can be drawing media (pencil, pen-and-ink, charcoal) or painting media (oil, watercolor, pastel). What is the rendering primitive? This

can be simulation models for board (the drawing/painting surface), brushes, and strokes.

3. **Process:** What rules are used to generate the image? This can be interactive (user interaction techniques, metaphors, devices,...) and/or automatic (rules, algorithms,...).
4. **Application:** What is the target public for the system?

Recent work has focused on the modeling of traditional artistic media and styles such as pen-and-ink illustration [36] and watercolor painting [3]. By providing rendering systems that use these alternative display models users can generate traditional renderings. These systems are not intended to replace artists or illustrators, but rather to provide a tool for users with no training in a particular medium, thus enabling them to produce traditional images.

In this paper we present results from our research in pencil illustration methods for NPR which is classified as follows:

1. *Subject:* 3D polygonal models.
2. *Materials:* Board (pencil drawing paper model), Brush/strokes/Media (graphite pencil, stumps, erasers).
3. *Process:* automatic interpreted rule-based scripts with optional user interaction through parameters adjustment at runtime.
4. *Application:* art, design, architecture, illustration, scientific visualization.

We chose pencil because it is a flexible medium, providing a great variety of styles in terms of line quality, hand gesture, and tone building. It is excellent for preparatory sketches and for finished rendering results. Pencil renderings are used by many people in different contexts such as scientific and technical illustration, architecture, art, and design.

The images from the results show that our simulation model produces similar results to the strokes and swatches generated with real pencils, blenders, and erasers. The images were generated

using rendering methods recommended by review of pencil literature [24, 9, 22, 14, 10, 35, 15, 21, 6, 12, 19, 2] and contact with artists and illustrators. All the results were generated on an OCTANETM Power Desktop¹ and printed at 200 *dpi* on a 600 *dpi* HP LaserJet 5Si MX printer. Real samples were scanned at 150 *dpi* and printed at 150 *dpi*.

1.1 Related work

Our work is related to research on 3D non-photorealistic rendering dealing with display methods which approximate technical or artistic hand-drawn illustration or painting styles [1, 13, 25, 5, 26, 36, 32, 23, 11, 4, 18, 16, 3, 7].

We were inspired in our work by recent approaches that tailored 3D NPR techniques to particular media models, specifically the work of Winkenbach and Salesin [36] in which results were produced from emulating the pen-and-ink illustration style, and the work of Curtis *et al.* [3] describing a detailed simulation model for watercolor with its painting style. Our research has focused on developing a simulation model for the graphite pencil medium on drawing paper and implementing the basic rules for achieving traditional illustration styles adapted to the 3D rendering pipeline.

Our model for graphite pencils includes parameters for pencil lead composition and paper texture [29]. In addition to this our model allows the use of blenders and erasers [30]. Previous work on pencil simulation has addressed some of these issues. Vermeulen and Tanner [34] introduced a simple pencil model as part of an interactive painting system that does not include a model to handle textured paper, blenders, or erasers. Takagi and Fujishiro [33] presented a model for paper micro structure and pigment distribution for colored pencils to be used in digital painting. In the commercial realm, some interactive painting systems such as Fractal Design Painter² offer a pencil model with some interaction with the paper. Our pencil models improve the approximation of graphite pencil on drawing paper and the basic pencil drawing primitives.

1.2 Methodology

Our approach was to break the problem of simulating pencil drawings down into the following levels:

1. **Low level: Drawing materials:** low-level simulation models for wood-encased graphite pencil and drawing paper, and for blenders and kneaded eraser.
2. **Medium level: Rendering methods:** (a) Drawing primitives: pencil stroke and mark-making (for tones and textures) built on top of the drawing materials; (b) Rendering algorithms built on top of the drawing primitives. Algorithms for outlining, shading 3D objects with a look that emulates real pencil drawings.
3. **High level: Drawing composition:** Partial control of the drawing composition through ordering and repeating of drawing steps.

This approach allows that the effectiveness of each technique can be examined independently and in combination.

The next sections give a general description with results for each of the three levels. A detailed description of the components from each level can be found in Sousa and Buchanan [29, 30, 28].

¹All rendering is done in software.

²Even though a number of systems offer “pencil” mode it is difficult to determine what physical model, if any, is being used to simulate the graphite pencil, blenders, erasers, and the corresponding drawing primitives.

2 Low level: Drawing materials

2.1 Pencil and paper model [29]

Our approach is based on an observational model of how real graphite pencils interact with drawing paper. The goal was to capture the essential physical properties and behaviors observed in order to produce quality pencil marks at interactive rates. Our intention was not to develop a highly physically accurate model, which would result in a computationally expensive simulation.

Our model entails five main components:

1. **Pencil hardness:** Every pencil contains a writing core (or “lead”) which is made from a mixture of graphite, wax, and clay. The hardness of the lead depends on the amount of graphite and clay. The more graphite it contains, the softer and thicker it is. Pencil hardness is graded in nineteen degrees ranging from 9H (hardest) to 8B (softest).
2. **Pencil points:** Sharpening a pencil in different ways changes the shape of the contact surface between the pencil and the paper. A pencil point is defined by a polygonal shape and pressure distribution coefficients over the point’s surface. Pressure distribution coefficients are values between 0 and 1 representing the percentage of the pencil’s tip polygonal surface that, on average, makes contact with the paper. This value is used to locally scale the pressure being applied to the pencil.
3. **Drawing papers:** Paper textures for pencil work (categorized as smooth, semi-rough, and rough) have a slight roughness (“tooth” or grain) that enables lead material (graphite, clay, and wax particles) to adhere to the paper. We model the paper texture as a height field ($0 \leq h \leq 1$) as was reported by Curtis *et al.* [3]. These height fields can be either procedurally generated or digitized from a paper sample. Each paper location (x, y) accumulates lead material. The amount of material depends on the pencils that have crossed the location.
4. **Pencil and paper interaction:** Lead material is left on paper through friction between the lead and the paper. The amount of lead material depends on the pencil tip shape, the pressure applied to the pencil, and the pencil hardness. A pencil stroke changes these parameters to achieve different effects (Fig. 2). In addition to depositing lead, a pencil stroke may alter the texture of the paper by destroying its grains (see Fig. 3).

Interesting effects can be achieved by properly selecting the paper texture for pencil drawing. We implemented the pencil rubbing technique which is the process of reproducing a raised or impressed image or texture by placing a piece of paper over it and making a rubbing with the pencil [2, pp. 120-124]. Figure 4 illustrates results from our model by using a paper texture computed from a digitized sample of one coin and testing the effects of lead material over it.

2.2 Blender and eraser model [30]

A blender is any tool that can be used to soften edges or to make a smooth transition between tone values. We modeled two kinds of blenders: tortillons and stumps.

Erasers remove surface particles to lighten a drawing. We modeled the kneaded eraser which is one of the most effective erasers made for graphite pencil.

Like the pencil and paper model our approach is based on an observational model of how real blenders and erasers interact with lead material already deposited on drawing paper. The point shapes for blenders and kneaded erasers are defined as a polygonal outline similar to the modeling of pencil points. The model for interaction between blenders and erasers with lead and paper took into account

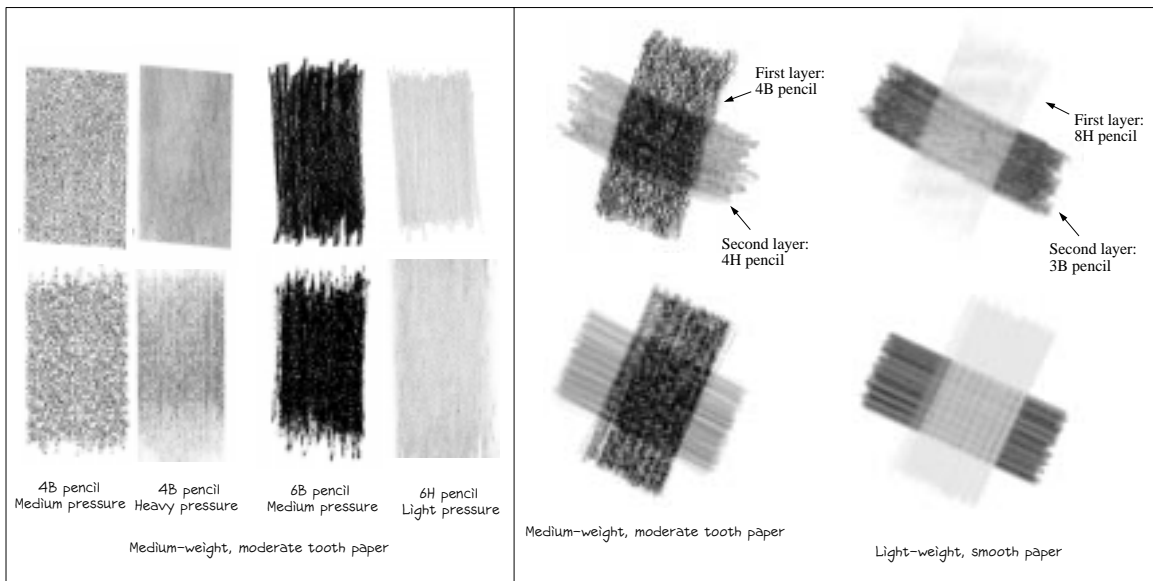


Figure 2: Our pencil and paper simulation model [29] applied over drawing paper (bottom row). Compare results with real pencil work (top row). The set of four swatches made with one single pencil (left box) was generated by adapting our model to an interactive illustration system. The set of blended swatches (right box) was generated by adapting our model to the mark-making primitive (subsec. 3.5).

parameters such as the particle composition of the lead over the paper, the texture of the paper, the position and shape of the blender and eraser, and the pressure applied to them.

Figure 5 illustrates the effects of blending and erasing pencil swatches over medium-weight, semi-rough paper's surfaces. Figure 6 illustrates results for tone rendering using a method called smudging [21, 15, 17]. Blenders and kneaded eraser are excellent for this rendering method, used for illustrating soft subject matter and shadows. Three rendering stages are necessary:

1. The tone values in the subject are rendered by using one pencil hardness (degree).
2. Certain portions of the drawing are smudged using blenders.
3. A kneaded eraser is then used to lighten the areas where there are highlights.

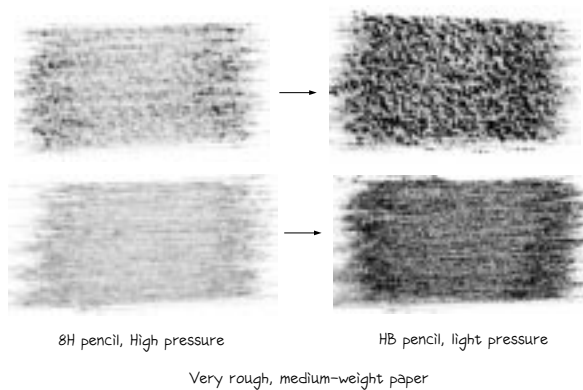
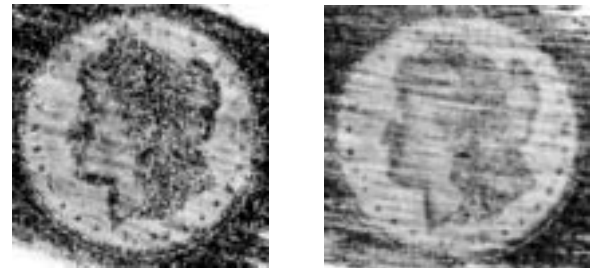


Figure 3: Our simulation model without paper damage (top row) and with paper damage (bottom row).



4B pencil
Light-medium pressure

4H pencil
light pressure

Fine quality transparent tracing paper on top of coin

Figure 4: Pencil rubbing technique [2, pp. 120-124] using our pencil and paper model.

3 Medium-level: Rendering methods [28]

3.1 3D models

Our pencil engine is built on the 3D modeling and rendering system presented in Glaeser [8]. The 3D models were generated using the modeling language from the same reference. Our system currently works just for polygonal models. The inputs are the visible edges, faces, and shadows. The lightness values for edges, faces, and shadows are evaluated using the Phong illumination model with flat shading, either as a pre-computation step, yielding a reference gray-scale image, or directly as the pencil strokes are generated. Most of the processing described in this paper assume that we have 3D information as well as the visible polygons and edges projected in the normalized coordinate space.

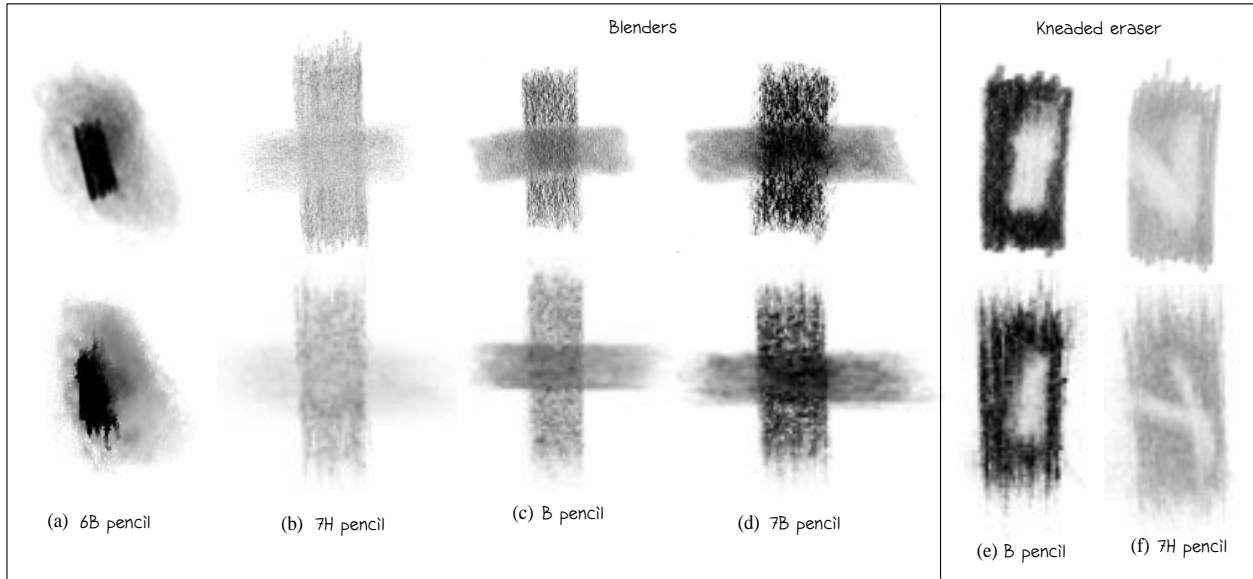


Figure 5: The bottom row shows results from our blender and kneaded eraser model [30] applied over the pencil and paper model [29]. Compare results with real pencil work (top row). For blenders: (a) a 6B pencil was rubbed firmly and then a tortillon was rubbed over it with circular gestures and medium to low pressure (20 secs); (b), (c), and (d): pencil strokes were rubbed vertically and then stumps were rubbed horizontally (15-25 secs).

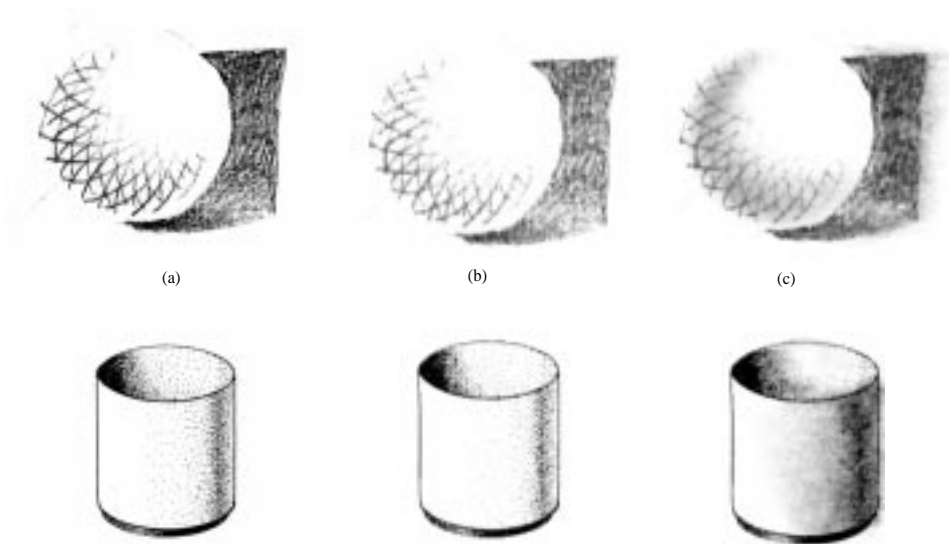


Figure 6: (a) Real pencil drawing of a sphere (resolution of 283x218 pixels) used for the paper's texture and rendered using a very soft pencil and cross-hatching to convey tone values (top row); Real pen-and-ink drawing of a cup (resolution of 240x282 pixels) also used as the paper's texture (bottom row). Next stages using our simulation models [29, 30]: (b) Automatic evaluation of the pencil and paper interaction model at each pixel using 2B pencil (1.24 secs for the sphere and 1.36 secs for the cup). (c) Adapting our blender and eraser model to an interactive illustration system and smudging the cross-hatched lines on the sphere (30 secs) and the ink dots on the cup (25 secs) creating a better effect on the tone. Shadow is also smudged around the sphere to make it softer. Notice the excess of graphite which spreads as we smudge the drawing. Kneaded eraser enhances highlight and clear some portions of the shadows (8 secs for the sphere and 10 secs for the cup)

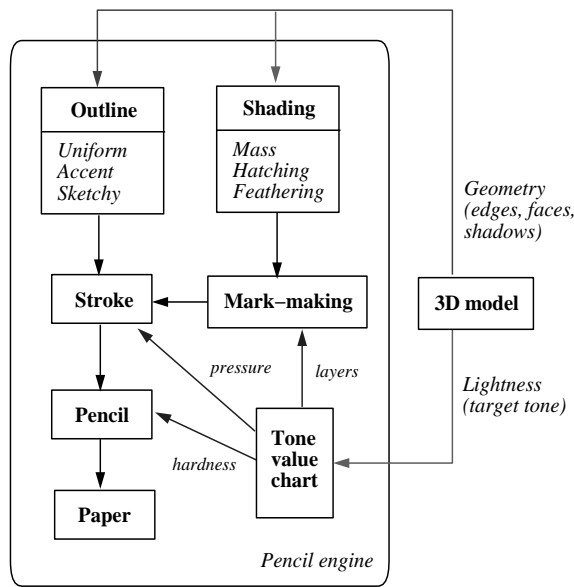


Figure 7: Architecture of our pencil rendering system.

3.2 Pencil engine

Our pencil engine (see Fig. 7) is organized in three main subsystems: (1) materials (pencil, paper), (2) primitives (stroke, mark-making), and (3) rendering methods (outline, shading, tone value chart). The parameters for materials and primitives are defined as scripts written in a C-based interpreted language. These scripts procedurally generate pencil marks and are automatically configured, depending on the rules for a particular pencil rendering method and on the stages of the drawing composition. The user also has the option of modifying those parameters during run-time while receiving feedback in real-time, thus guiding the rendering process. A tone value chart (subsec. 3.6) controls the number of pencil passes (layers) applied to the mark-making primitive, the pressure applied to each stroke, and the lead hardness of a particular pencil.

3.3 Stroke primitive

When using pencils, different types of strokes are produced depending on the pencil's hardness, its point, and how it is applied to the paper. Also there are many ways of handling the pencil and various effects over the stroke can be achieved [10, pp. 24-25], [2, pp. 39-42].

We define a pencil stroke S consisting of a number of line segments, a path, and a character function. The path $P(t) : [0, 1] \rightarrow R^2$ results from using a curve to approximate the line segments (Fig. 8, top row). Different approximation functions can be applied. We use Bezier curves and B-Splines.

The character function $C(t)$ varies stroke parameters at particular scalar distances t along the path. The character function includes parameters that relate to the factors that influence a real pencil stroke. Figure 8 shows a series of closeups of individual pencil strokes generated with our model. The strokes are rendered by scan-converting copies of the pencil tip polygon modified by the character function $C(t)$ placed at each pixel location along the path defined by the base curve with the waviness function added. Waviness functions simulate the hand movements by randomly modulating the curve defining the path. Previous researchers have reported using this approach [26, 20, 36, 32, 16]. We apply periodic waviness functions with random noise and turbulence to each pair of coordinates (x, y) at scalar distances t along the stroke's path.

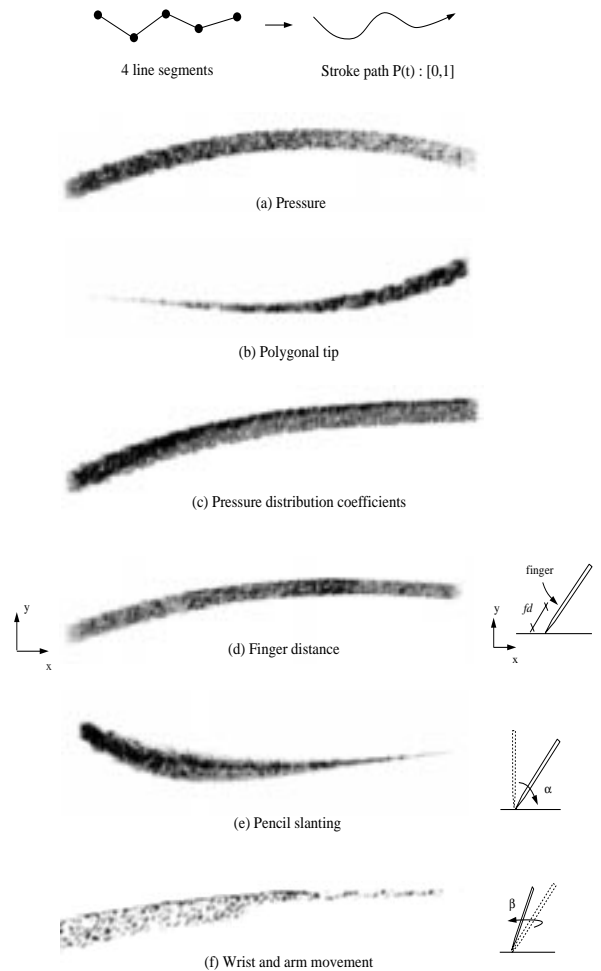


Figure 8: Example of a path for a pencil stroke (top row) and variation of six parameters from the character function $C(t)$ defining the pencil stroke primitive, rubbed with soft leads over a rough, medium-weight paper.

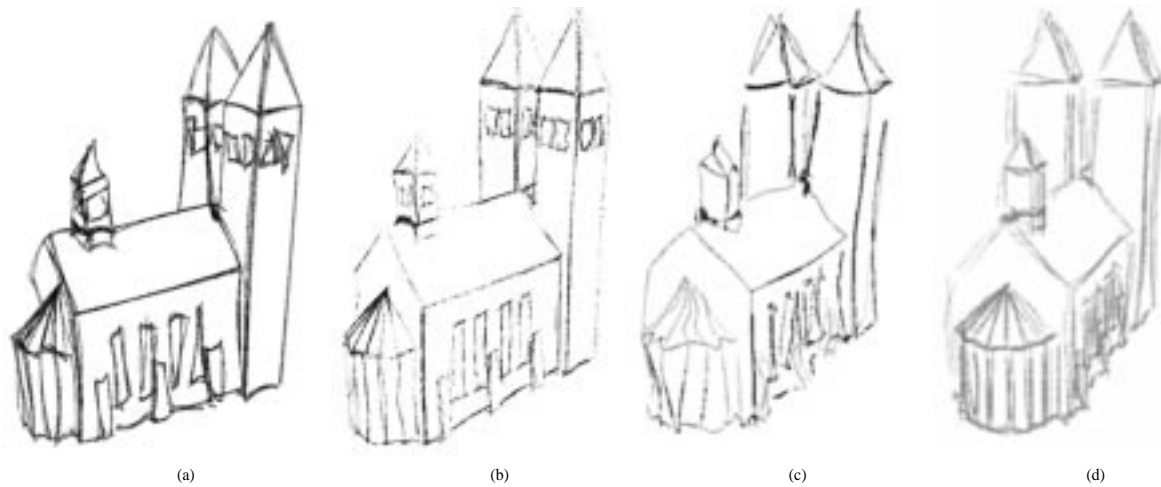


Figure 9: Outline results over semi-rough paper of 3D model of a church (298 edges, 100 faces) from our system: (a) uniform with 2B pencil (10 sec.), (b) accent with 3B pencil (7 sec.), (c) sketchy with H and B pencils (10 sec.) , and (d) less sketchy with 2H and HB pencils (9 sec.).

3.4 Outlining

In our system, outline pencil strokes are drawn for each visible edge from every visible face and shadow of the model. We have implemented three classes of traditional pencil-based outlines [10, 21]:

1. **Uniform or flat:** This method uses lines with a fixed degree of thickness and pressure for the whole drawing (Fig. 9(a)). It is good for illustration but it lacks sensitivity.
2. **Accented:** The pressure applied to the pencil is adjusted to lighten and darken the line giving more character and expressiveness to the outline.
3. **Sketchy:** The lines are drawn with quick and spontaneous strokes until the correct shape begins to emerge (Fig. 9(c), (d)). It emphasizes the vitality of the drawing marks themselves, making the drawing more subjective, because the focus is balanced between representation (what is drawn) and characterization (how it is drawn) [6].

3.5 Mark-making primitive

The mark-making primitive models a collection of strokes parallel to each other in a specific direction. It can be done in a formal, structured way or in a loose way, according to the drawing style and approach. The main purpose of this primitive is to create areas of tone and texture [12].

In our model, the mark-making M consists of a path $P(t) : [0, 1] \rightarrow R^2$ with one or more line segments (stroke primitive) on it. Figure 10 illustrates the results from our model of three basic kinds of mark-making techniques [12]: Hatching, zigzag (or back-and-forth), and feathering.

3.6 Light and shade

Drawing media differ in the techniques used to achieve shading that matches the target tone of the subject. In pen-and-ink the approach is to alternate the lines with the white of the paper itself. Each kind of line, depending on its proximity and thickness, can produce planes having different values and textures. This approach was implemented by Winkenbach and Salesin using prioritized stroke textures for the pen-and-ink renderer [36]. Graphite pencils on the other hand can produce gradations of values between black and

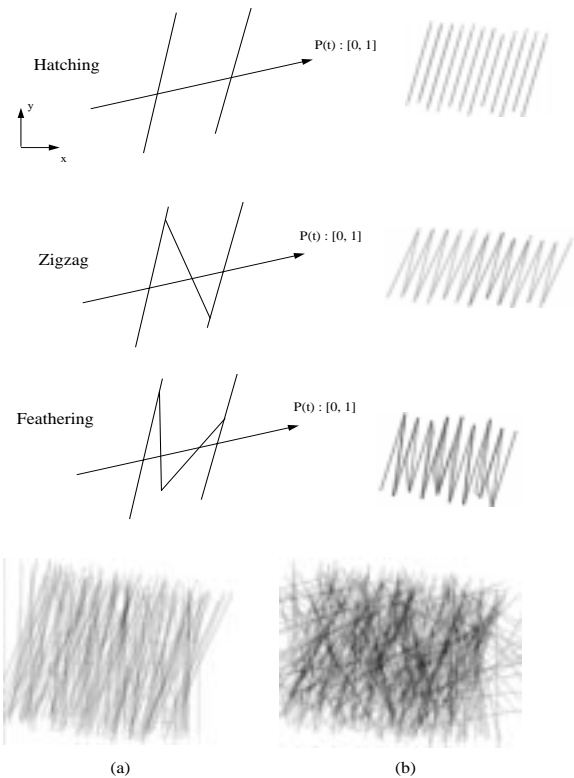


Figure 10: The mark-making primitive is used to build up tones and textures. This figure illustrates three variations of the mark-making primitive with results from our model. The two images (a) and (b) at the lower part of the figure start with one layer zigzagging and feathering in one direction over the path $P(t)$ with a medium soft pencil. Another layer of the primitive was laid at different angles variations.

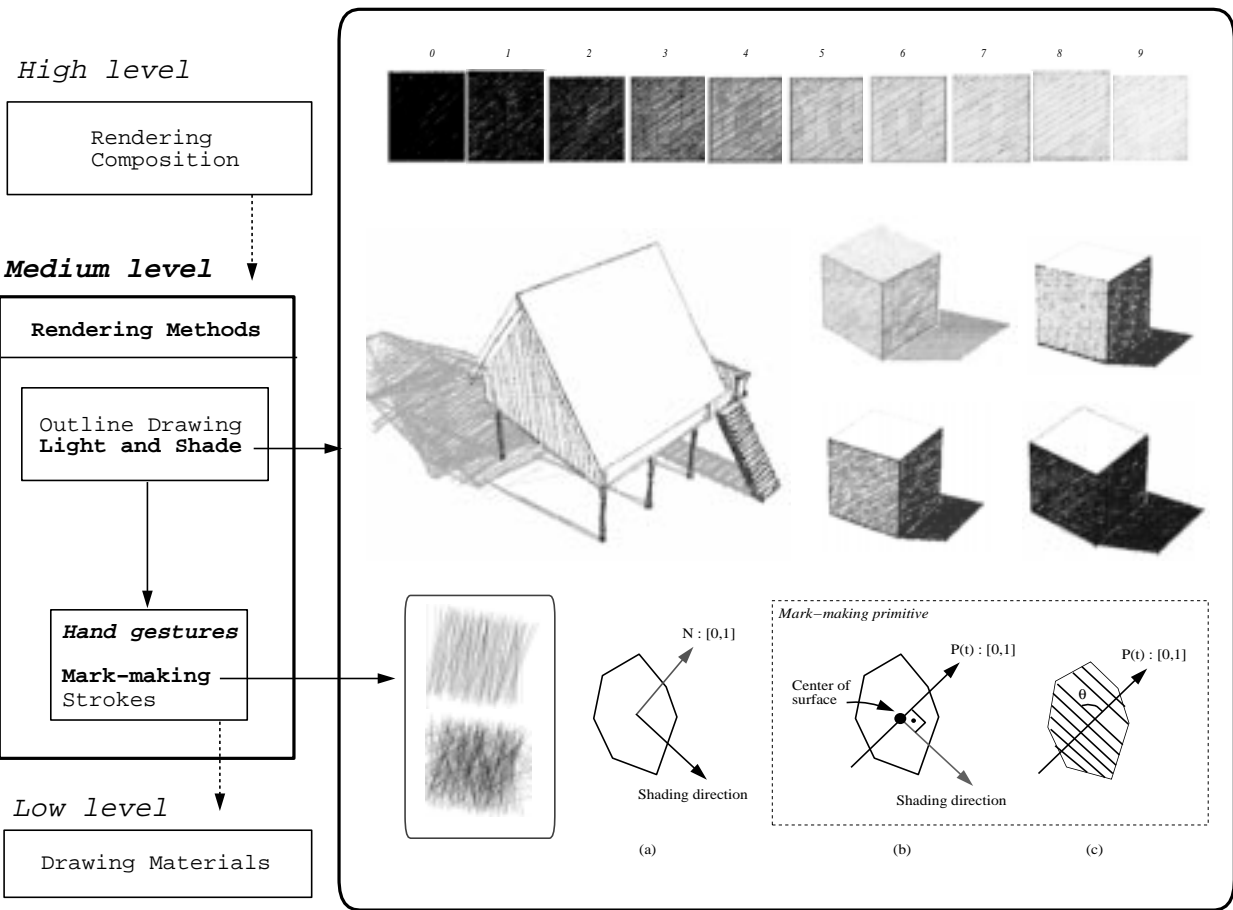


Figure 11: Examples of pencil rendering of 3D objects in tonal contrast using our system. The mark-making primitive (bottom of figure) is placed on each visible polygonal face from the model. The tone value chart (top of figure) determines the number of layers of the mark-making primitive, the pressure applied to the pencil, and the pencil hardness in order to match the target tone for the polygonal face (Fig. 7).

white. These gradations are usually organized into a tone value chart with three basic tones (light, mid and dark), or ten values, the lightest value being the white of the paper [10, 15, 21].

In our system every visible face and shadow from the 3D model are flat shaded resulting in a target tone. The system then finds the necessary parameters in the pre-computed look-up tone value chart (Fig. 11). These parameters are the number of times the mark-making primitive will be placed on the surface being shaded, the pressure applied to the stroke, and the pencil hardness (Fig. 7).

We implemented three basic pencil shading techniques:

1. **Mass shading:** In this method, the artist renders, by “mass” shading every visible tone in the subject as literally as possible. In “mass” shading the component pencil lines are so merged that their individual identity is wholly or largely lost [10].
2. **Hatching/Crosshatching:** the principle of hatching is drawing lines with one definite and continuous movement, parallel to each other, and very near together to produce an even tone. The hatching mark-making primitive (see Fig. 10) is used with the collection of strokes in the shading direction and equal distance between every pair of strokes.

Cross-hatching is the rendering of tone values by superimposing one series of parallel lines diagonally across another series of parallel lines [24]. It can be achieved by placing additional layers of the hatching mark-making primitive at different shading directions on top of the current pencil marks.
3. **Feathering or scumble:** With this technique the strokes are plainly visible because the pencil is used with a greater degree of freedom, blending tones optically so that while individual strokes are retained, they are also overlaid to create smoother tones [21, 10, 35]. The feathering mark-making primitive (see Fig. 10) is used.

4 High-level: Drawing composition

The control of the drawing composition is an important aspect of both traditional illustration practices and non-photorealistic rendering methods. Composing an illustration means putting together things and arranging them in order, to make one unit out of them all. Composition issues include proportion of the picture space according to the subject, focal points in the drawing, tone value studies, atmospheric effects, and so on [10, 22]. Some of these issues have been investigated in NPR research. Strothotte *et al.* [32] control the placement of lines depending on the areas of the image needing more attention. Streit and Buchanan [31] present techniques for creating non-photorealistic half toned images by controlling importance functions and the type and number of drawing primitives. Seligmann and Duncan [27] describe an automated intent-based approach to illustration which fulfills high-level description of the communicative intent and stylistic choice.

4.1 Drawing steps control

With our system it is possible to control the composition of a drawing work from the initial sketch to the finished rendering, a process achieved in a variety of drawing steps [10, 22, 21]. The rendering proceeds in layered steps emulating the process that artists take in order to make sure that the composition is correct at specific steps. Each drawing step is implemented by configuring the parameters of the pencil and the rendering methods described. Each step can be repeated a number of times before moving to next step. Figure 12 shows an example of how an illustration is improved by rendering in progressive steps in this way. The parameters for each step are

configured according to the guidelines found on pencil drawing literature [10, 22, 15].

5 Conclusions and future work

In this paper we presented non-photorealistic rendering methods that simulate the basic rendering techniques used by artists and illustrators familiar with graphite pencil rendering. The methods are based on traditional pencil illustration techniques recommended by review of pencil literature and contacts with artists and illustrators. We implemented rendering techniques for automatic outlining and shading of 3D polygonal models. These techniques are built on top of an observational model of graphite pencil and drawing paper [29], and on the mark-making and stroke primitives [28]. We also describe the partial control of the drawing composition through ordering and repeating of drawing steps from preparatory sketches to finished rendering results. We have also presented an observational model of blenders and kneaded eraser [30] to be used with the graphite pencil and drawing paper model presented. The model for interaction between blenders and erasers with lead and paper took into account parameters such as the particle composition of the lead over the paper, the texture of the paper, the position and shape of the blender and eraser, and the pressure applied to them. We have illustrated the results of our blender and eraser model by duplicating pencil swatches and by generating images.

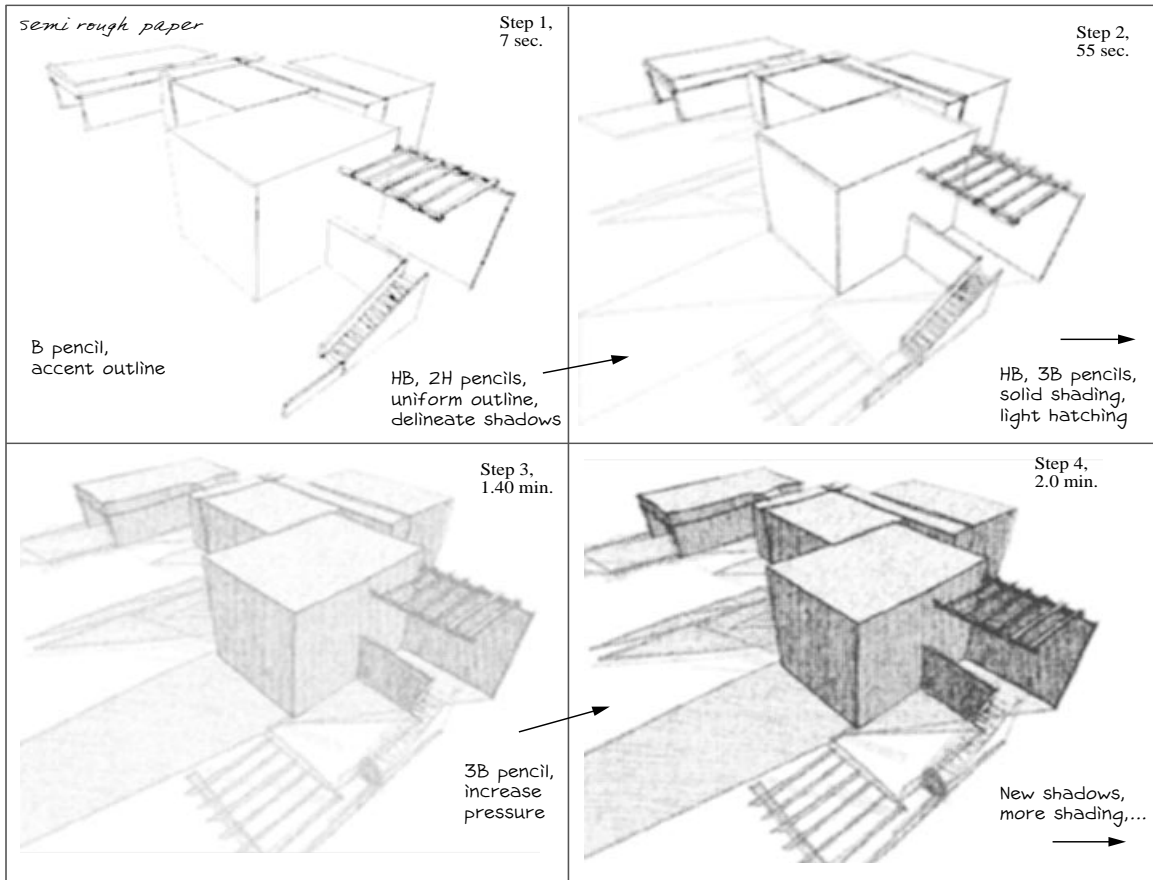
Several research issues remain open for future study in computer-generated pencil drawing. Methods for texture representation and other pencil outlining and shading techniques may also be explored and extended to render various classes of 3D models from different contexts (architecture, art, design). Drawing composition techniques can be further explored and modeled into a computer-generated pencil rendering system.

Acknowledgments

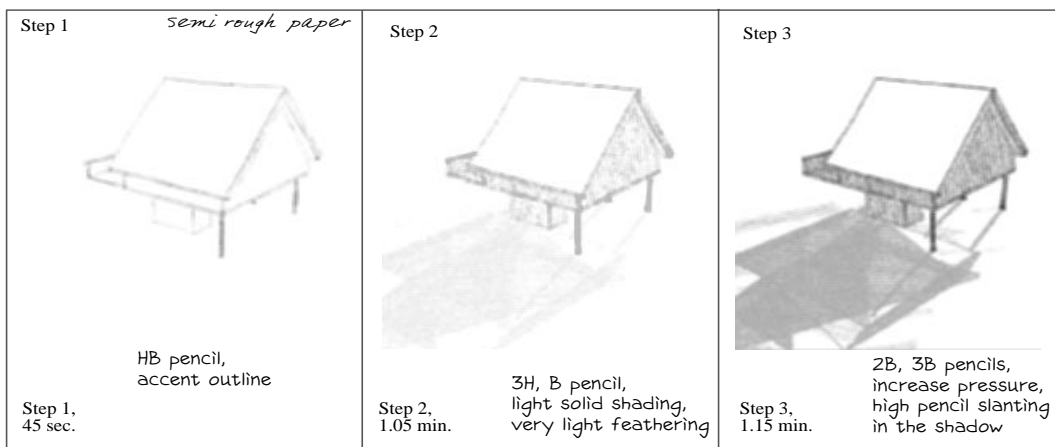
This research work was sponsored by the National Council of Scientific and Technological Development of Brazil (CNPq) and by the Natural Sciences and Engineering Council of Canada (NSERC). The authors wish to thank members of the University of Alberta graphics lab for their reviews and comments. Further thanks are due to Desmond Rochfort and Barbara Maywood of the Department of Art and Design, University of Alberta for their constructive criticism and positive comments.

References

- [1] A. Appel, F.J. Rohlf, and A.J. Stein. The Haloed Line Effect for Hidden Line Elimination. In *ACM Computer Graphics (Proc. of SIGGRAPH '79)*, volume 13, pages 151–157, August 1979.
- [2] S.W. Camhy. *Art of the Pencil: A Revolutionary Look at Drawing, Painting, and the Pencil*. Watson-Guption Publications, 1997. ISBN 0-8230-1373-1.
- [3] C.J. Curtis, S.E. Anderson, J.E. Seims, K.W. Fleischer, and D.H. Salesin. Computer-Generated Watercolor. In *Proc. SIGGRAPH '97*, pages 421–430, August 1997.
- [4] P. Decaudin. Cartoon-looking rendering of 3d-scenes. In *Research Report INRIA 2919*. INRIA, Syntim Research Group, June 1996.
- [5] D. Dooley and M.F. Cohen. Automatic Illustration of 3D Geometric Models: Surfaces. In *Proc. IEEE Visualization '90*, pages 307–314, October 1990.



(a) Modern house (450 edges, 224 faces)



(b) Beach house (510 edges, 238 faces)

Figure 12: The evolution of a pencil drawing in traditional steps [15] implemented in our system. For composition (a) steps 2 and 3 are repeated 2 times. For composition (b) step 3 is repeated 3 times.

- [6] D. Douglas and D. van Wyk. *The Drawing Process: Rendering*. Prentice Hall, Inc., 1993. ISBN 0-13-219833-9.
- [7] G. Elber. Line Art Illustrations of Parametric and Implicit Forms. *IEEE Trans. Visualization and Computer Graphics*, 4(1):71–81, September 1998.
- [8] G. Glaeser. *Fast Algorithms for 3D-Graphics*. Springer-Verlag New York, Inc., 1994. ISBN 0-387-94288-2.
- [9] A.L. Gupstill. *Sketching and Rendering in Pencil*. The Pencil Points Press, Inc., New York, 1933.
- [10] A.L. Gupstill. *Rendering in Pencil*. Watson-Gupstill Publications Inc., New York, 1977. ISBN 0-8230-4531-5.
- [11] P.M. Hall. Non-Photorealistic Shape Cues for Visualization. In *Winter School of Computer Graphics*, February 1995. Univ. of West Bohemia, Plzen, Czech Republic.
- [12] J. Horton. *An Introduction to Drawing*. Dorley Kindersley Limited, 1994. ISBN 0-13-123902-3.
- [13] T. Kamada and S. Kawai. An Enhanced Treatment of Hidden Lines. *ACM Transactions on Graphics*, 6(4):308–323, October 1987.
- [14] N. Lalibert and A. Mogelon. *Drawing with Pencils, History and Modern Techniques*. Van Nostrand Reinhold Company, New York, 1969.
- [15] D. Lewis. *Pencil Drawing Techniques*. Watson-Gupstill Publications Inc., New York, 1984. ISBN 0-8230-3991-9.
- [16] L. Markosian, M.A. Kowalski, S.J. Trychin, L.D. Bourdev, D. Goldstein, and J.F. Hughes. Real-Time Nonphotorealistic Rendering. In *Proc. SIGGRAPH '97*, pages 415–420, August 1997.
- [17] B. Maywood. Personal communication. *Dept. Art and Design, University of Alberta*, 1998.
- [18] B.J. Meier. Painterly rendering for animation. In *Proc. of SIGGRAPH 96*, pages 477–484, August 1996.
- [19] H. Misawa. *Introduction to Pencil Techniques (Easy Start Guide)*. Books Nippan, 1994. ISBN 4766107144.
- [20] R. Barzel M.P. Salisbury, S.E. Anderson and D.H. Salesin. Interactive Pen-and-Ink Illustration. In *Proc. SIGGRAPH '94*, pages 101–108, July 1994.
- [21] G. Price. *Pencil Drawing*. (from the Art is... Video Series). Crystal Productions, August 1993. SBN 1-56290-077-3.
- [22] C. Ruffino. *A Guide to Pencil Drawing*. Van Nostrand Reinhold Company, New York, 1969.
- [23] T. Saito. Real-Time Previewing for Volume Visualization. *Symp. Volume Visualization*, pages 99–106, October 1994.
- [24] J. Salwey. *The Art of Drawing in Lead Pencil*. B.T. Batsford, Ltd., London, 1925. Second edition.
- [25] T.T. Sasada. Drawing Natural Scenery by Computer Graphics. *Computer Aided Design*, 19(4):212–218, May 1987.
- [26] S. Schofield. *Non-Photorealistic Rendering: A Critical Examination and Proposed System*. PhD thesis, Middlesex Univ., England, October 1993.
- [27] D.D. Seligmann and S. Feiner. Automated generation of intent-based 3d illustration. In *Proc. SIGGRAPH '91*, pages 123–132, July 1991.
- [28] M.C. Sousa and J.W. Buchanan. Computer-Generated Graphite Pencil Rendering of 3D Polygonal Models. *Submitted for publication*.
- [29] M.C. Sousa and J.W. Buchanan. Observational Models of Graphite Pencil and Drawing Paper for Non-Photorealistic Rendering. *Submitted for publication*.
- [30] M.C. Sousa and J.W. Buchanan. Observational Model of Blenders and Erasers in Computer-Generated Pencil Rendering. In *Proc. Graphics Interface '99 (to appear)*, June 1999.
- [31] L. Streit and J.W. Buchanan. Importance driven halftoning. In *Proc. Eurographics '98*, pages 207–217, August 1998.
- [32] T. Strothotte, B. Preim, A. Raab, J. Schumann, and D.R. Forsey. How to Render Frames and Influence People. *Computer Graphics Forum*, 13(3):455–466, August 1994.
- [33] S. Takagi and I. Fujishiro. Microscopic Structural Modeling of Colored Pencil Drawings. In *SIGGRAPH '97 Visual Proc.*, page 187, August 1997.
- [34] A.H. Vermeulen and P.P. Tanner. Pencilsketch — A Pencil-Based Paint System. In *Proc. Graphics Interface '89*, pages 138–143, June 1989.
- [35] E. W. Watson. *Course in Pencil Sketching, Four Books in One*. Van Nostrand Reinhold Company, New York, 1978. ISBN 0-442-29230-9.
- [36] G. Winkenbach and D.H. Salesin. Computer-Generated Pen-and-Ink Illustration. In *Proc. SIGGRAPH '94*, pages 91–100, July 1994.