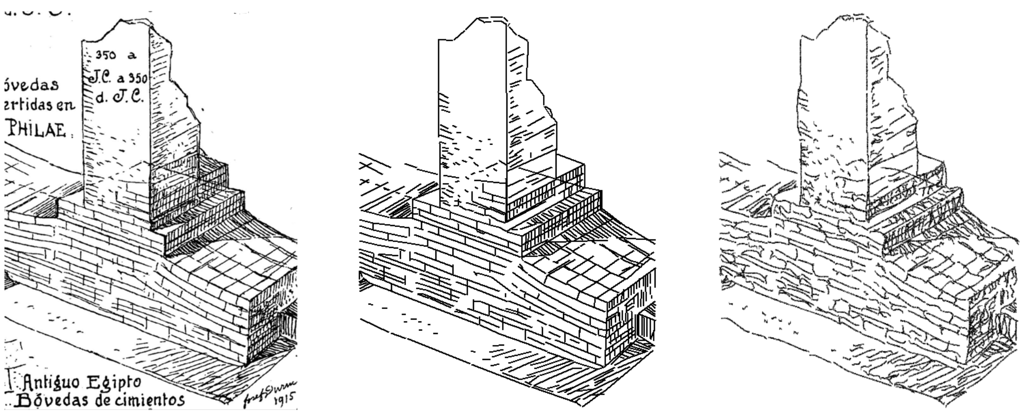
Mimicking Hand-Drawn Pencil Lines



(a) Hand Drawn Illustration. (b) Input Vector Lines. (c) Our Result.

Figure 1: *Our method takes as input lines specified only via their start and end points, such as the vector line drawing in (b), and produces a line drawing that mimics a real hand drawn graphite pencil drawing (a) by modeling human arm trajectory and a statistical model of graphite texture to produce unique, aesthetically pleasing and natural looking lines, without the need for databases of example strokes.*

Abstract

For many applications such as architecture, early design sketches containing accurate line drawings can often mislead the target audience. Approximate human-drawn sketches are typically accepted as a better way of demonstrating the fundamental design concepts. To this end we have designed an algorithm that creates lines that perceptually resemble human-drawn lines. Our algorithm works directly with input point data using simple Newton's first and second laws and a mathematical model of human arm trajectory movement. Further, the algorithm does not rely on a database of human lines, nor does it require any input other than the end points of the lines to generate a line of arbitrary length. The algorithm will generate any number of aesthetically pleasing and natural looking lines, where each one is unique. The algorithm was designed by conducting various user studies on human line sketches, and analyzing the lines to produce basic heuristics. We found that an observational analysis of human lines made a bigger impact on the algorithm than a statistical analysis. A further study has shown that the algorithm produces lines that are perceptually indistinguishable from that of a hand-drawn straight pencil line.

Keywords: Nonphotorealistic rendering, image processing, natural media simulation, pencil rendering, dynamic optimization yielding voluntary arm movement trajectory.

1 Introduction

NPR images convey information more effectively by omitting extraneous detail (abstraction), by focusing the viewer's attention on relevant features (emphasis), and by clarifying, simplifying, and disambiguating shape. In fact, a distinguishing feature of NPR is the concept of controlling and displaying detail in an image to enhance communication. The control of image detail is often combined with stylization to evoke the perception of complexity in an image without its explicit representation. NPR imagery, therefore, allows the:

- communication of uncertainty—precisely rendered computer graphics imply an exactness and perfection that may overstate the fidelity of a simulation or representation; and
- communication of abstract ideas—simple line drawings, like the force diagrams used in physics textbooks, can communicate abstract ideas in ways that a photograph cannot.

Although there are many current computer-generated drawing techniques that enable the creation of complex stylized images, such stylization techniques are typically limited to a library of previously drawn strokes. These techniques may not provide lines with the qualities of expressiveness and aesthetics matching hand-drawn illustrations. This paper presents a model based upon observation and statistical analysis of hand-drawn lines in conjunction with a model of human arm movement to create unique lines—given only a start and an end point and without the use of a large sample line database. Our method formulates and reproduces a curvature and texture that conforms to a real human drawn pencil line. In addition, our method does not require the setting of user-determined parameters (patterns of deformation, pressure, density, etc.).

The goal of this research is to capture the essence of a single stroke, drawn by humans as straight pencil lines of arbitrary length, and encode it into an algorithm. In turn, an application may use this algorithm to produce a line that resembles a human-drawn line, and use it to replace traditional computer-drawn lines (e. g., [Bresenham 1965]). Ideally, such an algorithm would reproduce the details carried by a human-drawn pencil line, without the need of storage libraries or user input to specify line attributes (as is the case, e. g., with [Sousa and Buchanan 2000]). Since the lines do not have set widths, colors, or particular textures, our proposed method will approximately reproduce the pencil details within the stoke it follows. In addition, such a line should not have repeated sections so that each line is unique.

Our work is divided into two parts for generating a human-like drawn line: (a) synthesizing the path that corresponds to human arm movement and (b) synthesizing the pencil texture applied along the path inspired by the textures produced by real pencils by Schlechtweg et al. [1998] and Strothotte and Schlechtweg [2002]). Based on this approach our algorithm:

- produces high quality simulations of hand-drawn lines;
- easily incorporates into existing applications;
- produces lines of arbitrary length;
- does not require a large library of sample lines; and
- creates unique lines for every set of input point pairs.

Our contribution, therefore, is a high quality pencil media line reproduction agent for creating aesthetically pleasing lines that mimic human-drawn lines. For this purpose, we use methods of image synthesis and a model of human arm movement for its replication. Our method avoids computationally expensive techniques and large storage space while continuously producing new, unique lines.

2 Previous Work

Our work draws from research on interactive non-photorealistic rendering (NPR) methods that approximate artistic hand-drawn images or paintings. In particular, we draw from NPR approaches for generating human line drawings and the simulation of graphite pencil texture, texture synthesis, and literature on the trajectory of human arm movement while drawing.

2.1 Human Line Drawings

Characteristics of lines in sketched and hand-drawn images have been studied closely in the field of non-photorealistic rendering (NPR). Many algorithms captured style characteristics and applied multiple parameters (such as length, width, pressure, etc.). Previous methods [Schlechtweg et al. 1998; Strothotte and Schlechtweg 2002; Hsu and Lee 1994] used style parameters and distorted a textured predefined piecewise polynomial curve or polygon path to create a stylized line. Other approaches reproduced and synthesized similar styles from examples lines [Jodoin et al. 2002; Kalnins et al. 2002; Freeman et al. 2003; Kalnins et al. 2004; Brunn 2006]. Each of these approximations used large libraries for storing example styles for comparison. Our simple method is limited to one piece of acquired texture depending on the desired pencil softness, and uses it as a basis for the replication of the lines.

We were inspired in our work by approaches that simulated pencils as a medium, specifically the work by Sousa and Buchanan [1999; 2000], who contributed a low-level simulation of graphite pencils on a paper medium. Their work focused on generating fill strokes, used for hatching purposes to reproduce artistic drawings. Our method differs because our lines are not restricted to short strokes and can vary greatly in length with no repeating segments. We also base our work on interactive pen-and-ink illustration by Salisbury et al. [1994], who described a level-of-detail system that interactively produces multiples of stokes to avoid tediously placing them manually. Our algorithm for drawing lines could easily be incorporated into the approaches of Sousa et al. and Salisbury et al., adding the benefit of a model of human arm movement and a simple perceptual simulation model for graphite pencils without the requirement of a library of lines to copy, paste and reshape.

2.2 Texture Synthesis

We were also influenced by an idea for texture synthesis based upon a sequential synthesis method [Gagalowicz and Ma 1985], that synthesized a new texture by preserving the second order statistics of the natural texture into the newly synthesized texture. Gagalowicz and Ma [1985] also provided experimental results demonstrating that the visual system is only sensitive to second-order spatial averages of a given texture field. More recent texture synthesis research renames second order statistics using the term co-occurrence models. These are defined as the proportion of co-occurrence of pairs of luminesces when their locations differ by a delta in the texture field plane.

Copeland et al. [2001] used a texture similarity metric based on the texture field of a model texture, which has a high correlation with human perception of textures. The multi-resolution version of their algorithm "Spin Flip" (also explained in [Copeland et al. 2001]) proved the best performance and resulted with pleasing outputs. Zalesny and Gool [2001] introduced a similar texture synthesis method that based its simulation on image intensity statistics. They collect the first order statistics (an intensity histogram), then extract the co-occurrence matrix (CCM) using cliques. A clique is a pair of two points (a head and a tail), and a clique type is a multiple of cliques at fixed relative positions see [Zalesny and Gool 2001]. The CCM only stores the distribution of intensity differences between the heads and tails pixels for a given orientation. The conventional way of calculating the CCM is by summing all the joint probabilities of intensities for a given pixel neighborhood into their relative position in the CCM. Our work follows their path but differs in the overall clique type selection criteria, we chose the conventional method to acquire the co-occurrence matrix.

2.3 Dynamic Optimization of Human Arm Movement

In order to produce a good simulation of a human line drawing we have also examined studies of the coordination of voluntary human arm movements. Human motor production has been analyzed, modeled and documented for well over a century [Woodworth 1899; Bernstein 1967; Adams 1971; Schmidt 1975]. Over the past few decades, theories of the functions of the Central Nervous System (CNS) with respect to human arm movement lead to the hypothesis of various mathematical models [Flash and Hogan 1985; Uno et al. 1989; Bizzi et al. 1991; Mazzoni et al. 1991; Kawato and Gomi 1992; Contreras-Vidal et al. 1997].

According to these CNS theories, arm movements are produced in either one of two ways:

- Natural movements maintain a constant ratio of angular velocities of joints to bring reduction in control complexity and constrain the degrees of freedom.
- Hand trajectories in extra-corporal space and joint rotations are tailored to produce the desired or intended hand movements.

Palmondon [1995] describes a synergy of agonist and antagonist neuromuscular systems involved in the production of arm movements. He developed his theory by modeling the impulse responses to neuromuscular activities. His system produces a close proximity bell-shaped velocity profile to represent an entire point-to-point movement.

The *minimum jerk model* introduced by Flash and Hogan [1985] formulated an objective function to solve a dynamic optimization problem for measuring the performance of any possible movement as the square of the jerk (rate of change of acceleration) of the hand position integrated over an entire movement from start to end positions. Flash and Hogan [1985] showed that the unique trajectory of planar, multi-joint arm movements that yields the best performance was in agreement with experimental data. Their analysis was based solely on the kinematics of movement and independent from the dynamics of the musculoskeletal system. We adopt the Flash and Hogan model because the model represents human arm trajectory in a planar field, similar to the movement of a human hand guided pencil across a piece of paper.

3 Overview

There are two parts to this work, the first is to construct an algorithm to generate a realistic path. The second is to synthesize a suitable texture to mimic a human line using a specific pencil. Our approach is to analyze both the path and the stroke characteristic (the style) as suggested in [Schlechtweg et al. 1998; Strothotte and Schlechtweg 2002].

3.1 The Path

We construct a path that conforms to a human arm trajectory using the method described in Flash and Hogan [1985]. We use this method as it provides "the smoothest motion to bring the hand from an initial position to the final position in a given time" [Flash and Hogan 1985]. Our goal is to simulate a hand drawn line only given two points. The Flash and Hogan model produces trajectories that (1) are invariant under translation and rotation and (2) whose shape does not change with amplitude or duration of the movement. Both of these properties suit our purpose very well.

Our lines are defined by Equation 1, a fourth degree polynomial in which $x(0), y(0)$ are the initial points and $x(f), y(f)$ are the end points.

$$
\begin{aligned}
x(t) &= x0 + (x0 - xf)(15t^4 - 6t^5 - 10t^3) \\
y(t) &= y0 + (y0 - yf)(15t^4 - 6t^5 - 10t^3) \qquad (1)
\end{aligned}
$$

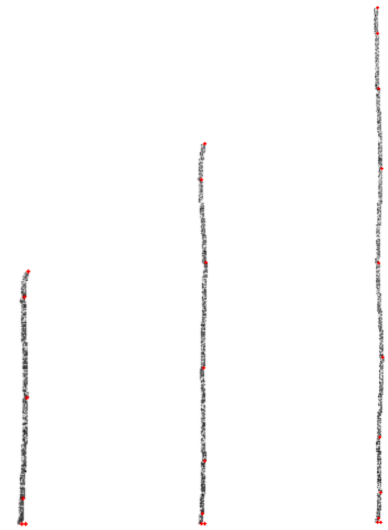
The value of t varies from 0 to t_{final}. We found empirically that $t_{final} = 2$ provides satisfactory results. The points found on the trajectory are then used as the control points for a Catmull-Rom interpolating spline [Shirley et al. 2005]. The number of points (defined by the time step, Δt) depend on the length of the line. Experimental evidence [Flash and Hogan 1985] shows that short hand drawn lines are perceptually closer to straight lines and longer lines have more variation. We conducted experiments to find reasonable values for Δt and used the results to construct Table 1. Figure 2 shows the positions of the control points, for three lines of varying length using each of the proscribed values for Δt.

Flash and Hogan conducted studies to generate trajectories, with the participants seated upright holding a two-link manipulandum. The participant's shoulders were restrained throughout the experiments, which in turn constrained the trajectory drawn. We conducted a pilot study in which the participants were seated upright in front of a horizontal table with no constraints. Each participant drew a number of pencil lines between pairs of points. The data we collected showed considerably more variation from the center line than can be accounted for with the variation of trajectories calculated by Equation 1.

To introduce a similar variation we incorporated a deviational force as an extension to the Flash and Hogan model. A deviational force $\vec{D}$ approximately normal to the Catmull-Rom spline is applied at each control point. A restoring force $\vec{R}$ increases in value the further the control point is displaced using a mass spring model. The velocity at each control point is calculated by differentiating Equation 1 with respect to t.

$$
\begin{aligned}
\vec{F} &= \vec{D} - k_1\vec{R} \qquad (2) \\
\vec{R} &= \vec{P_t} - \vec{P_{t+1}} \qquad (3)
\end{aligned}
$$

where P_t is the position of the control point at time t. The spring constant was set at $k_1 = 0.1$. The initial value for the deviation force



(a) Short line. (b) Medium Line. (c) Long line.

Figure 2: *Generated lines and control point positions for varying lengths. For (a) $\Delta t = 0.5$, for (b) $\Delta t = 0.3$ and for (c) $\Delta t = 0.2$. The red dots represent the control point.*

was chosen with some random variation in the range $[-3, +3]$ and re-assigned at each time step. The reason for this was to provide sufficient variation to closely resemble human drawn line paths. Experiments were conducted to validate these choices (see Section 5).

approx. Line Length in pixels	Time step Δt
$[0:200]$	0.5
$[200:400]$	0.3
> 400	0.2

Table 1: *Empirical Values for the time step Δt.*

3.2 The Texture

Example lines drawn by human participants using a wide range of pencils of the following types: *2H, H, HB, F, B, 3B, 6B, 8B* (examples are shown in Table 2) were scanned and analysed for their statistical information. The pencil lines were acquired on plain, 100% recycled, white paper. The following steps were taken to correctly capture the textural properties of the model texture:

- First, the range of grey levels was determined for pixels along the approximate centre of the pencil line and the histogram for the range $[min_{mid}, max_{mid}]$ was recorded.
- Next, the histogram of the grey levels for the line image was determined. The line segment is surrounded by white pixels, and since very few white pixels appear inside the line image, white is not considered in the histogram. This histogram is recorded in the intensity range $[min_{range}, max_{range}]$.
- The co-occurrence matrix for the line image is then computed.

To determine the co-occurrence matrix (CCM), each scanned image was rotated so that the line appeared to be approximately vertical. For each pixel, at (p,q), value $i \in [0:255]$ we found $(p,q+1)$, value $j \in [0:255]$ (the immediate neighbor in the positive y-direction) and used these values (i, j) as indices to increment the appropriate cell of the co-occurrence matrix as in Equation 4:

$$C(i,j) = \sum_{p=1}^{n} \sum_{p=1}^{m} \begin{cases} 1 & if\ \ I(p,q) = i\ \ and\ \ I(p+\Delta x, q+\Delta y) = j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

We use the dynamic range and the co-occurrence of a target pencil texture to synthesize the new texture. The predefined pencil width parameter is used to place the histogram values onto the path. A grey value distribution technique is formulated according to statistical observations. For most pencil lines, the distribution of grey pixel values starts at a dark value in the middle of the line and falls off according to the bell-shaped curve in the direction normal to the line, see Figure 3.

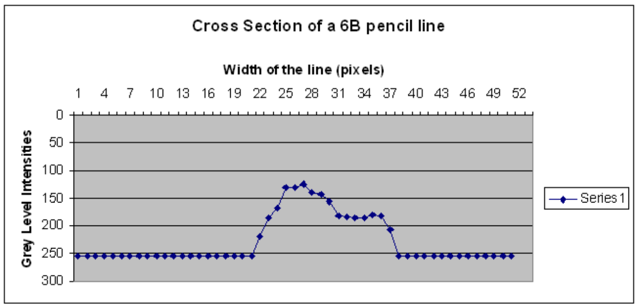


Figure 3: *The approximate bell shaped curve showing intensity values of cross sections of a number of lines drawn with a 6B pencil.*

According to this behavior we distribute the grey levels of the centre pixels of the generated spline, replicating the histogram in the range $[min_{mid}, max_{mid}]$. An approximate direction normal to the line is determined and the pixels along this direction either side of the centre pixel are set to values randomly chosen in the range $[min_{range}, max_{mid}]$ and pixels further away are given a lighter intensity in the range $[max_{mid}, max_{range}]$, see Figure 4. This serves as a basis for applying the co-occurrence matrix.

Figure 4: *Initial synthesized line texture. Placing the grey dynamic range values across the width and length of the path uniformly.*

The grey values of pixels are then read along the length of the line and each pixel is compared to a neighborhood of 3 pixels. Depending on whether or not the combination of pixel intensities exist in the CCM, each pixel's grey value is then either not changed or replaced with an existing neighboring value that best represents the relative pixel (index) positions in the co-occurrence matrix. This co-occurrence process can be repeated over the entire line multiple times until a threshold is reached of how well the synthesized co-occurrence matrix represents the model CCM (see Figure 5).

Once complete, a one pass Gaussian filter is applied to the generated lines to reduce tone differences and filter out aliasing effects. Further investigation of this method could better address line aliasing, such as using a multi-pass Gaussian filter, to enhance the quality of the presented lines. However, even without such further adaptation, our texture synthesis model enabled us to synthesize perceptually convincing textures of a pencil line as shown in Figure 6, Section 4 Section 5.

Figure 5: *The values of the pixels are analyzed and pairs of pixels are compared to their indices in the co-occurrence matrix and replaced with an appropriate value if their pair does not exist.*

Figure 6: *A 3×3 One-pass Gaussian filter is then applied resulting in the final pencil line texture.*

4 Results

The human line drawing system is implemented in C++ and runs at effectively on a 2.00GHz Intel dual core processor workstation, without any additional hardware assistance. The system has proven to be successful at assisting users in easily producing a variety of images. All line drawing figures in this paper indicated as "generated by "HLA"" were drawn using our Human Line Algorithm system. Table 2 shows a comparison of hand-drawn lines with lines that were synthesized by our system. Users can draw lines at interactive rates.

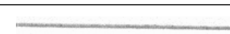
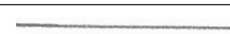
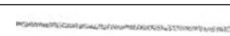
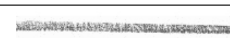
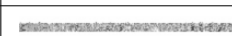
Line Type	Real Human Line	"HLA" Generated lines
H		
HB		
B		
3B		
6B		
8B		

Table 2: *Line samples: comparison of hand-drawn lines with synthesized lines deviation force set to zero.*

5 Verification

We designed a study with ethics approval using eighteen images; nine of which were scanned human made drawings, and the other nine were exact replications of the scanned images, made using our line algorithm. The aim of this study was to check whether our generated lines would pass for real human drawn lines. The study

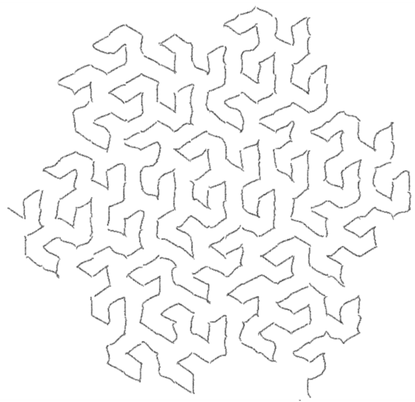


Figure 7: *Flowsnake space filling curve using a B pencil generated by HLA.*

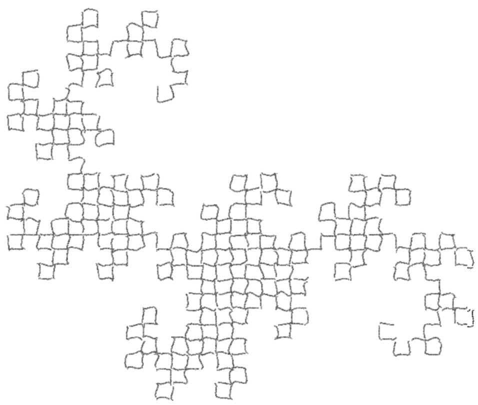


Figure 8: *Dragon space filling curve using a B pencil generated by HLA.*

was performed on eleven subjects, all graduate and undergraduate students in the department of Computer Science at the University of Victoria. Each participant spent three seconds viewing each image and then selecting true if they thought the drawing was hand-made, false if they thought the drawing was computer generated.

Computer-generated images were perceived to be hand-drawn fifty nine times whereas hand-drawn ones were perceived to be computer-generated only twenty six times. A paired sample t-test showed a significant difference between the amount of times each image was rated to be of the correct type (paired $t(10) = 2.849$, p ¡ .05). This results shows that the generated line drawings were good enough that they were often taken for hand-drawn ones.

6 Applications

We have applied our technique to a number of different example domains. Our line generation can be easily incorporated into an existing graphics program that produces line end points, for example by artists or algorithmically. A pencil type and the end points of the line are all that is required. For example Figure 7 and Figure 8 show two space filling curves, Gosper's Flowsnake [Gardner 1976] and the dragon curve generated with human like lines using the *B* pencil setting. The synthesized lines are applied to each of the short individual line segments of each curve. The drawings could be improved by detecting co-linear segments and processing them as a longer line to better emulate a humans artist.

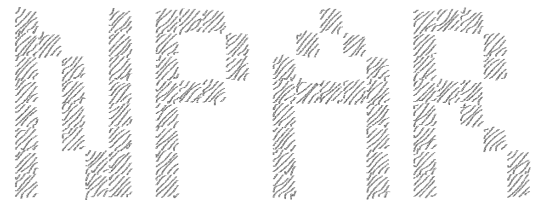


Figure 9: *Line hatching using a 6B pencil generated by HLA.*

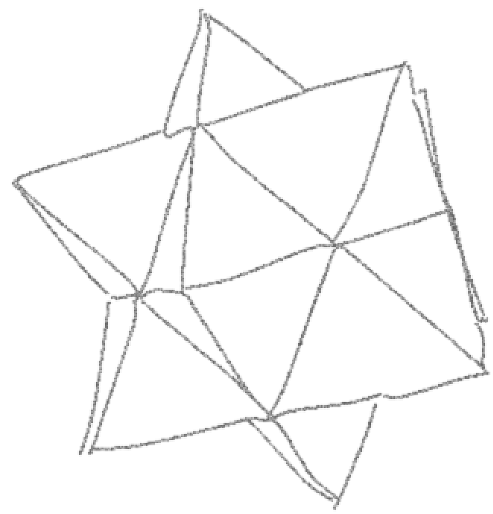


Figure 10: *A 36-sided object using H pencil generated by HLA.*

In Figure 9 hatching lines are generated and replaced with synthesized lines; in this way we can simulate the effect of filling a part of the plane with human-like lines as shown.

Figure 10 shows an example where the output from a CAD application has been replaced by our lines. A rendering of an object with 36-edges is shown using a simulated H pencil, the same image simulated with a 8B pencil is shown in Figure 11.

In Figure 12 an architectural drawing has been rendered using our lines. This example shows, the variability of the line paths and compared with the original straight line drawing provides a convincing hand made image.

Finally, by capturing (e. g., though tablets) or tracing the lines drawn by artists we can apply the pencil style to those drawings as well, as was shown in the teaser Figure 1 on the first page of the paper.

7 Conclusion and Future Work

The main contribution of this work is to provide a system that will serve as a high quality pencil line reproduction agent, to create aesthetically pleasing human drawn pencil lines by using an image synthesis method and human arm movement replication. The method avoids computationally expensive techniques and large storage space. Similar approaches to the work documented here may work for drawing curved lines, but further investigation would be necessary to correctly mimic human arm movement for curved lines and the resulting graphite textures.

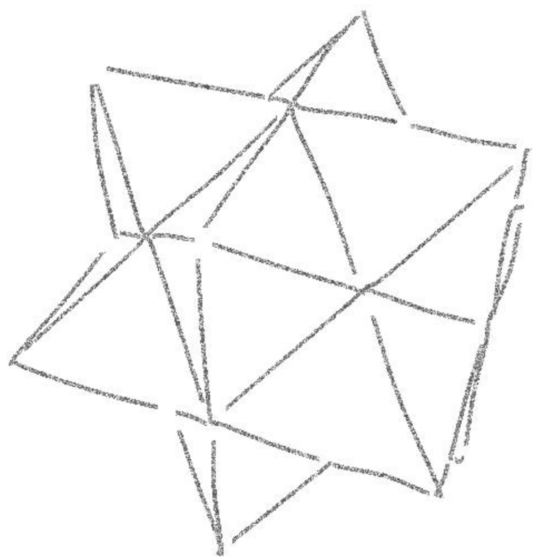


Figure 11: *A 36-sided object using 8B pencil generated by HLA.*

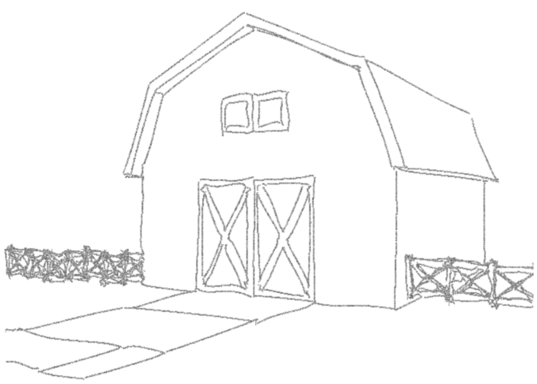


Figure 12: *A barn using H pencil generated by HLA.*

References

ADAMS, J. 1971. A Closed-Loop Theory of Motor Behavior. *Journal of Motor Behavior 3*, 111–149.

BERNSTEIN, N. 1967. *The Co-ordination and Regulation of Movements*. Pergamon Press, London.

BIZZI, E., MUSSA-IVALDI, F., AND GISZTER, S. 1991. Computations Underlying the Execution of Movement: A Biological Perspective. *Science 253*, 5017, 287–291.

BRESENHAM, J. 1965. Algorithm for Computer Control of a Digital Plotter. *IBM Systems Journal 4*, 1, 25–30.

BRUNN, M. 2006. *Curve Synthesis by Example*. Master's thesis, University of Calgary.

CONTRERAS-VIDAL, J., GROSSBERG, S., AND BULLOCK, D. 1997. A Neuronal Model of Cerebellar Learning for Arm Movement Control: Cortico-Spino-Cerebellar Dynamics. *Learning and Memory 3*, 475–502.

COPELAND, A. C., RAVICHANDRAN, G., AND TRIVEDI, M. M. 2001. Texture Synthesis Using Gray-level Co-occurrence Models: Algorithms, Experimental Analysis and Psychophysical Support. *Optical Engineering 40*, 11 (Nov.), 2655–2673.

FLASH, T., AND HOGAN, N. 1985. The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model. *The Journal of Neuroscience 5*, 7, 1688–1703.

FREEMAN, W. T., TENENBAUM, J. B., AND PASZTOR, E. C. 2003. Learning Style Translation for the Lines of a Drawing. *ACM Transactions on Graphics 22*, 1 (Jan.), 33–46.

GAGALOWICZ, A., AND MA, S. D. 1985. Sequential Synthesis of Natural Textures. *Computer Vision, Graphics, and Image Processing 30*, 3 (June), 289–315.

GARDNER. 1976. Mathematical games: In which "monster" curves force redefinition of the word "curve". *SCIAM: Scientific American 235*, 124–133.

HSU, S. C., AND LEE, I. H. H. 1994. Drawing and Animation Using Skeletal Strokes. In *Proc. SIGGRAPH*, ACM Press, New York, 109–118.

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

KALNINS, R. D., MARKOSIAN, L., MEIER, B. J., KOWALSKI, M. A., LEE, J. C., DAVIDSON, P. L., WEBB, M., HUGHES, J. F., AND FINKELSTEIN, A. 2002. WYSIWYG NPR: Drawing Strokes Directly on 3D Models. *ACM Transactions on Graphics 21*, 3 (July), 755–762.

KALNINS, R. D., MARKOSIAN, L., MEIER, B. J., KOWALSKI, M. A., LEE, J. C., PHILIP L. DAVIDSON, MATTHEW WEBB, J. F. H., AND FINKELSTEIN, A. 2004. Wysiwyg npr: Drawing strokes directly on 3d models. In *In Proceedings of Eurographics Symposium on Rendering '04*, Eurographics, 23–32.

KAWATO, M., AND GOMI, H. 1992. The Cerebellum and VOR/OKR Learning Models. *Trends in Neurosciences 15*, 11 (Nov.), 445–453.

MAZZONI, P., ANDERSEN, R., AND JORDAN, M. 1991. A More Biologically Plausible Learning Rule for Neural Networks. *Proceedings of the National Academy of Sciences 88*, 10 (May), 4433–4437.

PALMONDON, R. 1995. A Kinematic Theory of Rapid Human Movements, Parts I and II. *Biological Cybernetics 72*, 4 (Mar.), 295–307, 309–320.

SALISBURY, M. P., ANDERSON, S. E., BARZEL, R., AND SALESIN, D. H. 1994. Interactive Pen-and-Ink Illustration. In *Proc. SIGGRAPH*, ACM Press, New York, 101–108.

SCHLECHTWEG, S., SCHÖNWÄLDER, B., SCHUMANN, L., AND STROTHOTTE, T. 1998. Surfaces to Lines: Rendering Rich Line Drawings. In *Proc. WSCG 1998*, vol. 2, 354–361.

SCHMIDT, R. A. 1975. A Schema Theory of Discrete Motor Skill Learning. *Psychological Review 82*, 4 (July).

SHIRLEY, P., ASHIKHMIN, M., GLEICHER, M., MARSCHNER, S., REINHARD, E., SUNG, K., THOMPSON, W., AND WILLEMSEN, P. 2005. *Fundamentals of Computer Graphics*, 2^{nd} ed. A. K. Peters, Ltd., Natick, MA.

SOUSA, M. C., AND BUCHANAN, J. W. 1999. Computer-Generated Graphite Pencil Rendering of 3D Polygonal Models. *Computer Graphics Forum 18*, 3 (Sept.), 195–207.

SOUSA, M. C., AND BUCHANAN, J. W. 2000. Observational Model of Graphite Pencil Materials. *Computer Graphics Forum 19*, 1, 27–49.

STROTHOTTE, T., AND SCHLECHTWEG, S. 2002. *Non-Photorealistic Computer Graphics. Modeling, Animation, and Rendering*. Morgan Kaufmann Publishers, San Francisco.

UNO, Y., KAWATO, M., AND SUZUKI, R. 1989. Formation and control of optimal hrajectory in human multijoint arm movement. *Biological Cybernetics 61*, 2 (june), 89–101.

WOODWORTH, R. 1899. The Accuracy of Voluntary Movement. *Psychological Review: Monograph Supplements 3*, 1–114.

ZALESNY, A., AND GOOL, L. V. 2001. A Compact Model for Viewpoint Dependant Texture Synthsis. In *Proc. SMILE*, Springer-Verlag, Berlin, vol. 2018 of *LNCS*, 124–143.