Computational Aesthetics in Graphics, Visualization, and Imaging (2008)
(Editors)

Mimicking Hand-Drawn Pencil Lines

Zainab Meraj[1] Brian Wyvill[1] Tobias Isenberg[2] Amy A. Gooch[1] Richard Guy[3]

[1]University of Victoria, BC, Canada [2]University of Groningen, Netherlands [3]University of Calgary, AB, Canada

Abstract

For many applications such as architecture, early design sketches containing accurate line drawings can often mislead the target audience [SSRL96]. 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 and a mathematical model of using a physically based model of human arm 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.

Categories and Subject Descriptors (according to ACM CCS): I.3.m [Computer Graphics]: Miscellaneous— non-photorealistic rendering, natural media simulation, pencil rendering, dynamic optimization yielding voluntary arm movement trajectory, image processing.

1. Introduction

Non-photorealistic Rendering (NPR) can 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 such as diagrams used in textbooks can communicate abstract ideas in ways that a photograph cannot.

While there is no precise metric yet discovered to differentiate between hand-drawn and computer-generated line drawings (some attempts have been made for specific techniques [MIA*08]), humans can typically distinguish with ease the difference by a shear glance. For pencil lines this may be due to changes in grey levels, a variation not proportional to the path, or a glitch in the path orientation. Such variations make it difficult to produce aesthetically pleasing, natural looking results that mimic human-drawn lines.

Our method is 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. In addition, our method does not require the setting of user-determined parameters (patterns of deformation, pressure, density, etc.). The only parameter users are required to select through the system interface is one of eight commonly used pencil types. Our method then formulates and reproduces a curvature and texture that conforms and mimics real human-drawn pencil lines.

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., [Bre65]). 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 [SB00]). Since the lines do not have set widths, colors, or particular textures, our proposed method will approximately reproduce the pencil details within the stroke it follows. In addition, such a line should not have repeated sections so that each line is unique.

We divide our algorithm for generating a human-like pencil lines into two parts: (a) synthesizing the path that corresponds to human arm movement and (b) synthesizing the pencil texture applied along the path [SSSS98, SS02], inspired by the textures produced by real pencils. Based on this general 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 library of sample lines (only a grey level dynamic range array and a co-occurrence matrix); 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 (NPR). Many algorithms captured style characteristics and applied multiple parameters (such as length, width, pressure, etc.). Previous methods [HL94, SSSS98, SS02] 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 [JEGPO02, KMM*02, FTP03, FS94, SD04, Bru06].

Our method differs from example-based methods in that we do not require example lines to generate unique paths and textures. For each pencil type there exists two pieces of information in the algorithm, a dynamic range, and a co-occurrence matrix. We simply only require vector endpoints to produce lines. Our aim is not to generate varying style from a single given style as seen in example-based methods, but to develop an algorithm that generates lines that vary in path orientation and texture synthesis mimicking observed real human movement and graphite pencil deposits noticed from straight line drawings on paper.

Our work is inspired by methods that simulated pencils as a medium, specifically the work by Sousa and Buchanan [SB99, SB00], who contributed a low-level simulation of graphite pencils on a paper medium. They designed a system to simulate graphite pencil on paper using microscopic observations [SB99]. Their work focused on generating fill strokes, using it 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. [SABS94], who described a level-of-detail system that interactively produces multiples of strokes to avoid tediously placing them manually. Our algorithm for drawing lines could easily be incorporated into the above approaches, 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 a texture synthesis method based upon sequential synthesis [GM85], that synthesized a new texture by preserving the second order statistics of the natural texture into the newly synthesized texture. Gagalowicz and Ma also provided experimental results demonstrating that the visual system is only sensitive to second-order spatial averages of a given texture field, which is one of the reasons we adopted such methodology. More recent texture synthesis research renames second order statistics (spatial averages) [GM85] using the term co-occurrence (CC) models or grey level co-occurrence (GLC) models [CRT01] in our case. These are defined as the proportion of second order probability statistics of pairs of grey levels when their locations differ by a delta in the texture field plane.

Choosing co-occurrence matrices to synthesize texture have been shown to be extremely efficient in human texture discrimination [JB87]; the amount of data necessary to achieve the synthesis is very small and the texture can be generated easily to fit all kinds of surface shapes and sizes. Using this method allows us to control the second-order spatial averages of a given texture, since the synthesis is achieved directly from them without the computation of higher order statistics [GM85]. Also, texture similarity techniques using

GLC probability distribution have shown to have high correlation with human perception of textures (the images appear to be very similar visually to the original natural ones).

Copeland et al. [CRT01], for example, used a texture similarity metric based on the texture field of a model texture. The multi-resolution version of their algorithm "Spin Flip" (also explained in [CRT01]) showed satisfactory performance and resulted with pleasing outputs.

Zalesny and Gool [ZG01] 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 [ZG01]). 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 [Woo99, Ber67, Ada71, Sch75]. 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 [FH85, UKS89, BMIG91, MAJ91, KG92, CVGB97].

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 are in extra-corporal space, joint rotations and additional non-physical parameters are tailored to produce the desired or intended hand movements.

Plamondon's model [Pla95] 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 [FH85] 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 [FH85] 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 as in the work done by Plamondon [Pla95]. 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 [SSSS98, SS02]. The results, verification method, and the applications where the algorithm can be applied are then presented followed by the conclusion and future work.

3.1. The Path

We construct a path that conforms to a human arm trajectory using the method described by Flash and Hogan [FH85]. 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" [FH85]. 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. All of these conclusions were based on observations of low frequency movements of human subjects. No high frequency movements were observed or implemented. Our work follows through with the same assumptions.

Their mathematical model satisfies our criteria in that it leads to a fast and interactive line path algorithm by providing a realistic overall velocity and acceleration profile and trajectory. We conduct user studies to explore whether orientation, missing from the mathematical model, plays a role in the ability to reproduce human-like lines. And will show that the disparity of the line orientation was not noticed by the users and did not effect the overall classification criteria used by participants.

Our lines are defined by Equation 1, a fifth order polynomial:

$$\begin{aligned} x(t) &= x_0 + (x_0 - x_f)(15\tau^4 - 6\tau^5 - 10\tau^3) \\ y(t) &= y_0 + (y_0 - y_f)(15\tau^4 - 6\tau^5 - 10\tau^3) \qquad (1) \\ \textit{where}, & \\ \tau &= \frac{t}{t_f}, t \in [0,2] \end{aligned}$$

(a) Short line.

(b) Medium Line.

(c) Long line.

Figure 1: *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.*

(2)

in which x_0, y_0 are the initial hand position coordinates at time increment t, and x_f, y_f are the final hand positions at $t = t_f$. The value of t varies from $[0,2]$ the from start position t_0 to end position t_{final}. We found empirically that $t_{final} = 2$ provides satisfactory results.

The trajectory points provide control points for a Catmull-Rom interpolating spline [SAG*05]. The number of points (defined by the time step, Δt) depend on the length of the line. Experimental evidence [FH85] 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 provide the results in Table 1. Figure Figure 1 shows the positions of the control points for three lines of varying length using each of the prescribed values for Δt.

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 different orientation pairs of points.

To introduce a variation across the line, we include a deviation parameter based on observations of how humans draw lines. The data we collected showed considerably more variation from the centre line that wasn't accounted for with the variation of trajectories assumed by Equation 1.

To represent this variation, we introduce a deviational parameter we call *squiggle* as an extension to the Flash and Hogan model. The reason for this was to provide sufficient variation to closely resemble human drawn line paths based on observations. Experiments were conducted to validate this choice (see Sec 5).

We define *squiggle* as a variable, D, for controlling the deviation normal to the Catmull-Rom spline at each of the lines control points also defined similarly by Strothotte et al. [SS02]. By setting the *squiggle* value to a maximal amplitude random displacements are used to give the appearance of an irregular path.

The deviational value is applied approximately normal to the Catmull-Rom spline at each of its control points. To achieve a varying frequency across the path, the value D is computed. The variable D varies randomly in order to provide variation along the path; we empirically found a range [-5,+5] worked for our lines regardless of length which results in a displacement value from each independent control point. The displacement is then added by the algorithm to the appropriate position parameter when calculating the control points positions in Equation 3.

$$D = RandomNumber[-5:+5] \tag{3}$$

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

We obtain texture for simulating hand drawn lines by scanning and analyzing the statistical properties of 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) on plain, 100% recycled, white paper. The following steps are taken to correctly capture the textural properties of the model texture:

- First, we determine the range of grey levels for pixels along the approximate centre of the pencil line (non-repeating catmull-rom control points) and record the histogram for the range $[min_{mid}, max_{mid}]$.
- Next, we determine the histogram of the grey levels for the line image, assuming that white pixels surround each line segment; since very few white pixels appear inside the line image, we do not consider white in the histogram. The histogram records intensities in the range $[min_{range}, max_{range}]$.
- Finally, we compute the co-occurrence matrix for the line image.

To determine the co-occurrence matrix (CCM), each image scan is rotated so that the line is approximately vertical. The co-occurrence matrix is updated by examining each pixel (p,q) with value $i \in [0,255]$ and it's immediate neighbor in the positive y-direction, $(p,q+1)$ with value $j \in [0,255]$. The values (i,j) are indices to increment the appropriate cell of the co-occurrence matrix C defined over an *nxm* image I, parameterized by an offset (Δx, Δy) 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 new the texture. We formulate a grey value distribution technique according to statistical observations that indicate, 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 2.

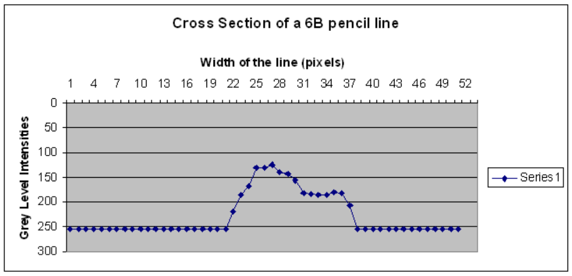


Figure 2: *The approximately bell-shaped curve showing intensity values of cross sections of a lines drawn with a 6B pencil.*

We first seed the line with initial values and then apply the co-occurrence matrix. According to the observed bell-shaped fall off of value, we distribute the grey levels of the centre pixels of the generated spline, replicating the histogram in the range $[min_{mid}, max_{mid}]$. We determine an approximate direction normal to the line and the pixels along this direction to 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 $[max_{mid}, max_{range}]$, as illustrated in the texture sample of Figure 3.

Figure 3: *Initial synthesized line texture. Placing the grey dynamic range values across the width and length of the path uniformly.*

Next, we apply the CCM by comparing a neighborhood of 3 pixels at a time. Depending on whether or not the combination of pixel intensities exist in the CCM, each pixel's grey value is then either left unchanged or replaced with an existing neighboring value based on the pixel index and the co-occurrence matrix. We repeat the co-occurrence process over the entire line multiple times until a the amount of pixel changes reach a minimum, indicating how well the synthesized co-occurrence matrix represents the model CCM (see Figure 4).

Once complete, a single-pass Gaussian filter is applied to the generated lines to reduce tone differences and filter out aliasing effects (Figure 5). Further investigation of this method

Figure 4: *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.*

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 5 and the following Sections Sec 4 and Sec 5.

Figure 5: *A 3 × 3 single-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 on a 2.00 GHz Intel dual core processor workstation without any additional hardware assistance. We can interactively draw lines while the system synthesizes the new path and texture. The system has shown 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 (HLA) system. Table 2 shows a comparison of hand-drawn lines with lines that were synthesized by our system.

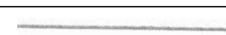
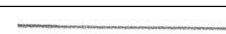
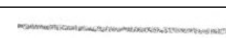
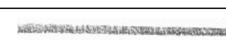
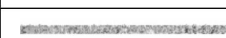
Line type	Real human line	"HLA"-generated lines
H		
HB		
B		
3B		
6B		
8B		

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

The following is a pseudo-code description of the HLA algorithm, to enable easy application of the method:

(1) Initialize the CCM and histogram to the user Selected Pencil type. (2) Read in Endpoints (3) If Endpoints == 2 then

- Find Control Points of the line between the two end points specified.
- Distribute random grey values across and along the line
- Use the CCM to replace random grey values along the line in 3x3 neighbourhood with existing co-located neighbour values in the CCM.
- Gaussian Blur the line. Else wait for a second endpoint entry.

(4) Else wait for a second endpoint

5. Verification

We designed a study 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 evaluate whether our generated lines would pass for real human drawn lines. The study was performed on eleven subjects, all graduate and undergraduate students in the department. 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. The timing was chosen empirically, such that participants had enough time to compare drawings without having too much time to over analyzing their decision.

Out of the 42decisions selected images to be of hand drawn type (when they were actually generated) and 30.5to be of generated type (when they were actually real). This made the ratio of mistakes made choosing generated line to be of real type and real line to be of generated type about 2 : 1.

A paired sample t-test showed that our computer-generated lines were significantly more often thought to be hand-drawn than the other way around (paired $t(10) = 2.849, p < .05$). This results shows that the HLA generated line drawings were good enough to pass for hand-drawn ones.

6. Applications

We apply our technique to a number of different example domains. Our line generation incorporates easily into existing graphics programs that produce line end points, and are editable by artists or additional algorithms. Our technique only requires the selection of a pencil type and specification of the end points of each line. For example, Figure Figure 6 and Figure Figure 7 show two space filling curves, Gosper's Flowsnake [Gar76] and the dragon curve, rendered with human like lines using the *B* pencil setting. The synthesized

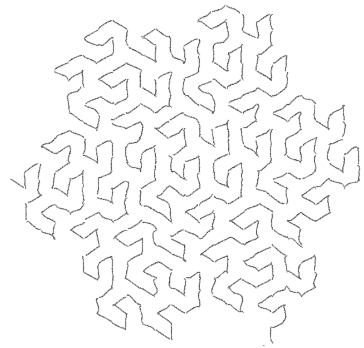


Figure 6: *Flowsnake space filling curve using a B pencil generated by HLA.*

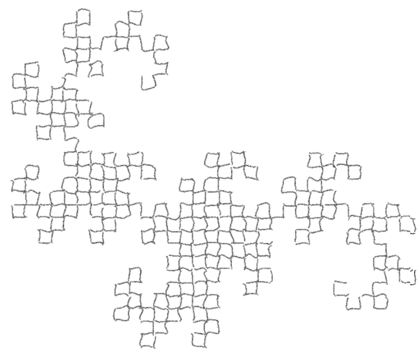


Figure 7: *Dragon space filling curve generated by HLA (B pencil).*

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.

In Figure 8 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.

In Figure 9 a hand drawn illustration is used to extract vector

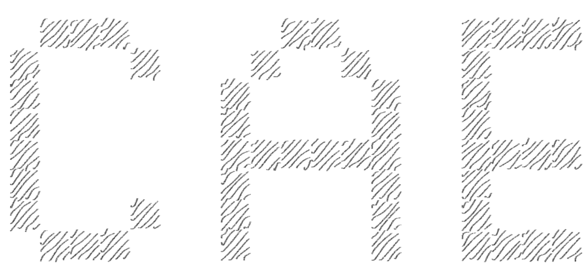
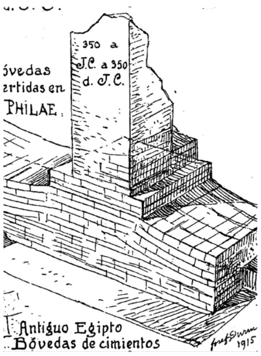
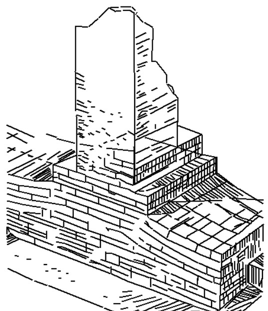


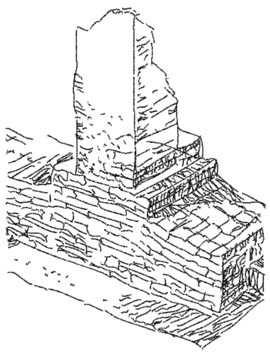
Figure 8: *Line hatching generated by HLA (6B pencil).*



(a) Hand Drawn Illustration.



(b) Input Vector Lines.



(c) Our Result.

Figure 9: *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.*

lines, then used as input to the "HLA" algorithm resulting in part (c) of the figure.

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 8B pencil.

In Figure 11 an architectural drawing has been rendered using our lines. This example shows the variability of the line paths and, when compared with the original straight line drawing, provides a convincing hand made image.

Finally, by capturing (e. g., through tablets) or tracing the lines drawn by artists we can apply the pencil style to those drawings as well.

We have applied our work in this paper to these four area: Space filling curves, architectural drawings (CAD, Google Sketch up), line patterns generated from the Life Game, and reproductions of artist's drawings.

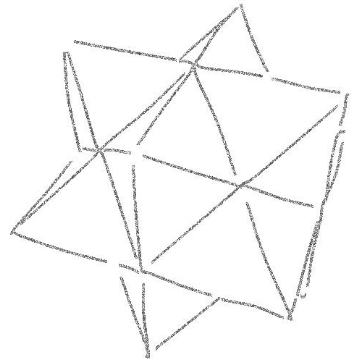


Figure 10: *A 36-sided object generated by HLA (8B pencil).*

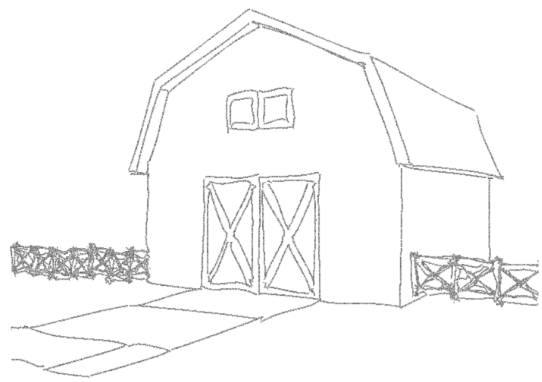


Figure 11: *A barn generated by HLA (H pencil).*

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 model. The algorithm avoids computationally expensive techniques and large storage space. More investigation is needed to mimic the appearance of pressure along the lines. Choosing parameters that will make the lines appear to have more character will defiantly increase the aesthetic property of the lines. Similar approaches to the work documented here may work for drawing curved lines Figure 12, but further investigation would be necessary to correctly mimic the resulting graphite textures on curved paths. Also more work can be done to mimic human hatching to a better extent through conducting studies and observing the relationship between arm and wrist movement when producing shorter strokes. CAe, Figure 8 is our first attempt to consider hatching as an application, more experiments and analysis are needed to present accurate human hatching techniques.

Figure 12: *Example curve generated by HLA, using Flash and Hogan curve optimization methods, inspired by [FS94].*

References

[Ada71] ADAMS J.: A Closed-Loop Theory of Motor Behavior. *Journal of Motor Behavior 3* (1971), 111–149.

[Ber67] BERNSTEIN N.: *The Co-ordination and Regulation of Movements*. Pergamon Press, London, 1967.

[BMIG91] BIZZI E., MUSSA-IVALDI F., GISZTER S.: Computations Underlying the Execution of Movement: A Biological Perspective. *Science 253*, 5017 (1991), 287–291.

[Bre65] BRESENHAM J.: Algorithm for Computer Control of a Digital Plotter. *IBM Systems Journal 4*, 1 (1965), 25–30.

[Bru06] BRUNN M.: *Curve Synthesis by Example*. Master's thesis, University of Calgary, 2006.

[CRT01] COPELAND A. C., RAVICHANDRAN G., TRIVEDI M. M.: Texture Synthesis Using Gray-level Co-occurrence Models: Algorithms, Experimental Analysis and Psychophysical Support. *Optical Engineering 40*, 11 (Nov. 2001), 2655–2673.

[CVGB97] CONTRERAS-VIDAL J., GROSSBERG S., BULLOCK D.: A Neuronal Model of Cerebellar Learning for Arm Movement Control: Cortico-Spino-Cerebellar Dynamics. *Learning and Memory 3* (1997), 475–502.

[FH85] FLASH T., HOGAN N.: The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model. *The Journal of Neuroscience 5*, 7 (1985), 1688–1703.

[FS94] FINKELSTEIN A., SALESIN D. H.: Multiresolution Curves. In *Proc. SIGGRAPH* (New York, 1994), ACM Press, pp. 261–268.

[FTP03] FREEMAN W. T., TENENBAUM J. B., PASZTOR E. C.: Learning Style Translation for the Lines of a Drawing. *ACM Transactions on Graphics 22*, 1 (Jan. 2003), 33–46.

[Gar76] GARDNER: Mathematical games: In which "monster" curves force redefinition of the word "curve". *SCIAM: Scientific American 235* (1976), 124–133.

[GM85] GAGALOWICZ A., MA S. D.: Sequential Synthesis of Natural Textures. *Computer Vision, Graphics, and Image Processing 30*, 3 (June 1985), 289–315.

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

[JB87] JULESZ B., BERGEN R.: Textons, the fundamental elements in preattentive vision and perception of textures. In *RCV87* (1987), pp. 243–256.

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

[KG92] KAWATO M., GOMI H.: The Cerebellum and VOR/OKR Learning Models. *Trends in Neurosciences 15*, 11 (Nov. 1992), 445–453.

[KMM*02] KALNINS R. D., MARKOSIAN L., MEIER B. J., KOWALSKI M. A., LEE J. C., DAVIDSON P. L., WEBB M., HUGHES J. F., FINKELSTEIN A.: WYSIWYG NPR: Drawing Strokes Directly on 3D Models. *ACM Transactions on Graphics 21*, 3 (July 2002), 755–762.

[MAJ91] MAZZONI P., ANDERSEN R., JORDAN M.: A More Biologically Plausible Learning Rule for Neural Networks. *Proceedings of the National Academy of Sciences 88*, 10 (May 1991), 4433–4437.

[MIA*08] MACIEJEWSKI R., ISENBERG T., ANDREWS W. M., EBERT D. S., SOUSA M. C., CHEN W.: Measuring Stipple Aesthetics in Hand-Drawn and Computer-Generated Images. *Computer Graphics & Applications 28*, 2 (Mar./Apr. 2008), 62–74.

[Pla95] PLAMONDON R.: A Kinematic Theory of Rapid Human Movements, Parts I and II. *Biological Cybernetics 72*, 4 (Mar. 1995), 295–307, 309–320.

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

[SAG*05] SHIRLEY P., ASHIKHMIN M., GLEICHER M., MARSCHNER S., REINHARD E., SUNG K., THOMPSON W., WILLEMSEN P.: *Fundamentals of Computer Graphics*, 2nd ed. A. K. Peters, Ltd., Natick, MA, 2005.

[SB99] SOUSA M. C., BUCHANAN J. W.: Computer-Generated Graphite Pencil Rendering of 3D Polygonal Models. *Computer Graphics Forum 18*, 3 (Sept. 1999), 195–207.

[SB00] SOUSA M. C., BUCHANAN J. W.: Observational Model of Graphite Pencil Materials. *Computer Graphics Forum 19*, 1 (2000), 27–49.

[Sch75] SCHMIDT R. A.: A Schema Theory of Discrete Motor Skill Learning. *Psychological Review 82*, 4 (July 1975).

[SD04] SIMHON S., DUDEK G.: Sketch Interpretation and Refinement Using Statistical Models. In *Rendering Techniques* (Aire-la-Ville, Switzerland, 2004), EUROGRAPHICS Association, pp. 23–32.

[SS02] STROTHOTTE T., SCHLECHTWEG S.: *Non-Photorealistic Computer Graphics. Modeling, Animation, and Rendering*. Morgan Kaufmann Publishers, San Francisco, 2002.

[SSRL96] SCHUMANN J., STROTHOTTE T., RAAB A., LASER S.: Assessing the Effect of Non-photorealistic Rendered Images in CAD. In *Proc. CHI* (New York, 1996), ACM Press, pp. 35–42.

[SSSS98] SCHLECHTWEG S., SCHÖNWÄLDER B., SCHUMANN L., STROTHOTTE T.: Surfaces to Lines: Rendering Rich Line Drawings. In *Proc. WSCG* (1998), vol. 2, pp. 354–361.

[UKS89] UNO Y., KAWATO M., SUZUKI R.: Formation and Control of Optimal Trajectory in Human Multijoint Arm Movement. *Biological Cybernetics 61*, 2 (June 1989), 89–101.

[Woo99] WOODWORTH R.: The Accuracy of Voluntary Movement. *Psychological Review: Monograph Supplements 3* (1899), 1–114.

[ZG01] ZALESNY A., GOOL L. V.: A Compact Model for Viewpoint Dependant Texture Synthsis. In *Proc. SMILE* (Berlin, 2001), vol. 2018 of *LNCS*, Springer-Verlag, pp. 124–143.