Mimicking Human-Drawn Pencil Lines

by

Zainab Faisal Al-Meraj
B.Sc. California State University, Long Beach 2002

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

Mimicking Human-Drawn Pencil Lines

by

Zainab Faisal Al-Meraj

B.Sc. California State University, Long Beach 2002

Supervisory Committee

Dr. Brian Wyvill (Department of Computer Science)

Supervisor

Amy Gooch (Department of Computer Science)

Departmental Member

Bruce Gooch (Department of Computer Science)

Departmental Member

Lynda Gammon (Department of Visual Arts)

External Member

Supervisory Committee

Dr. Brian Wyvill (Department of Computer Science)

Supervisor

Amy Gooch (Department of Computer Science)

Departmental Member

Bruce Gooch (Department of Computer Science)

Departmental Member

Lynda Gammon (Department of Visual Arts)

Outside Member

Abstract

In applications such as architecture, early design sketches containing accurate line drawings often mislead the target audience [Schumann et al. 1996]. Approximate human-drawn sketches are typically accepted as a better way of demonstrating fundamental design concepts. To this end I have designed an algorithm that creates lines that perceptually resemble human-drawn lines. My algorithm works directly with input point data and a physically–based mathematical model of human arm movement. Further, the algorithm does not rely on a database of human drawn 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 developed by conducting various user studies on human drawn pencil line sketches, and analyzing the lines to produce basic heuristics. I found that an observational analysis of human lines made a bigger impact on the algorithm than a statistical analysis. A further study shows that the algorithm produces lines that are perceptually indistinguishable from straight hand-drawn pencil lines.

Table of Contents

List of Figures

List of Tables

Acknowledgements

It is a pleasure to thank the many people who made this thesis possible, to them I am greatly indebted.

Firstly, and most importantly, praise and gratitude be to ALLAH for all I have been able to achieve throughout my life. With his grace I have reached my goals and experienced events I never imagined possible.

It is difficult to express my gratitude to my M.Sc. supervisor, Dr. Brian Wyvill. With his enthusiasm, his inspiration, and his great efforts to explain things clearly and simply, he helped to make Computer Graphics fun for me. Throughout my thesis-writing period, he provided encouragement, sound advice, good teaching, and lots of good ideas.

I would like to thank the many people who have taught me how to research and opened my eyes to NPR: my graduate mentors Tobias Isenberg, Amy Gooch and Bruce Gooch. For their kind assistance with writing letters and papers, giving wise advice and helping with various applications. I would also like to thank Dr. Richard Guy for his initial inspiration, and Petra Isenberg for her statistical wiz.

Thanks to Lynda Gammon for accepting to be on my committee at such short notice, her initial artistic critique helped develop much of my work.

I am indebted to my many student colleagues at UVIC Graphics Lab for providing a stimulating and fun environment in which to learn and grow academically. I am especially grateful to Sven Olsen, Jan Fisher, Masamichi Sugihara, Chris Serson, Jeremy Long and David Whittaker. Their technical and non-technical help made the difference I needed to overcome various obstacles.

I wish to thank my best friends Eman Q. Fayrouz, Fatma Khalaf, Abeer Al-Turkomani, Zainab Al-Jazzaf, and Zainab Al-Andaleeb and all my UVIC girl-friends for helping me get through the difficult times, and for all the emotional support, entertainment, and caring they provided.

I am grateful to the secretaries in the computer science departments of the University of Victoria, for helping the department run smoothly and for assisting me in many different ways.

I would like to express my deepest gratitude and sincere thanks and appreciation to my beloved parents for their infinite support and encouragement throughout my life. A special thanks and great appreciation to my *mum*, who is and always will be my source of love, inspiration and support, without her help in caring for my children, guidance and encouragement during the last two years this work would not have been possible.

I would like to acknowledge Autodesk for their research donation of Maya and 3DStudioMax. This material is based upon work partly supported by the National Science and Engineering Research Council of Canada, the University of Victoria, and Kuwait University. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the sponsors.

Dedication

To my parents, and my children *Batool* and *Haidar*.

Chapter 1

Introduction

NPR images can convey information more effectively by omitting extraneous detail (abstraction), by focusing the viewers attention on relevant features (emphasis), and by clarifying, simplifying, and disambiguating shape [Gooch and Gooch 2001], [Strothotte and Schlechtweg 2002]. 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.

In this research I examine NPR approaches in rendering techniques to create artistically pleasing computer-generated pencil drawings. The goal of using this NPR rendering method is to regenerate lines in portrait, sketched, and 3D model drawings that perceptually resemble human-drawn lines.

1.1 The Problem

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. In this thesis a model is presented; it is based on 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, without the use of a large sample line database. My algorithm formulates and reproduces a curvature and texture that conforms to a real human drawn pencil line. In addition, the algorithm does not require the setting of user-determined parameters (patterns of deformation, pressure, density, etc.). The only parameter users are required to manipulate through the system interface is one of eight commonly used pencil types and the input lines.

The ultimate 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 it could be used to replace traditional computer-drawn lines (e.g., the Bresenham Line Algorithm [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 with Sousa and Buchanan [2000]). Since the lines do not have set widths, colors, or particular textures, my 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.

No precise metric has been discovered to differentiate between hand-drawn and computer-generated line drawings. Some attempts have been made for specific techniques, stippling for example, in Maciejewski et al. [2008]. However humans can typically distinguish differences with ease by a glance, ([Isenberg et al. 2006]). 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. I conduct user studies to show that the proposed approach successfully captures and synthesizes aesthetically pleasing lines that mimic hand-drawn lines.

1.2 Capturing Aesthetic Qualities of Human-drawn Lines

The aim of this research is to capture the aesthetic quality of human-drawn lines. This is done using: methods taken from NPR approaches for generating human line drawings, the simulation of graphite pencil texture, texture synthesis, and literature on the trajectory of human arm movement while drawing.

The work is divided into two parts for generating a human-like drawn line:

1. synthesizing the path that corresponds to human arm movement; and
2. synthesizing the pencil texture corresponding to a specific pencil softness and applying it along the path.

I analyze both the path and the stroke characteristic (the style) as suggested by Schlechtweg et al. [1998], and Strothotte and Schlechtweg [2002].

An overview of the system is shown in Figure 1.1, it also includes an interactive display module that is used as an input/output screen. The first step (phase *A*) is a one time off-line statistical capturing routine. Each pencil type is used to create an absolute straight line that is scanned into the computer; the line is then analyzed for its content of grey levels. A histogram and a co-occurrence matrix (CCM) of the line is calculated and stored

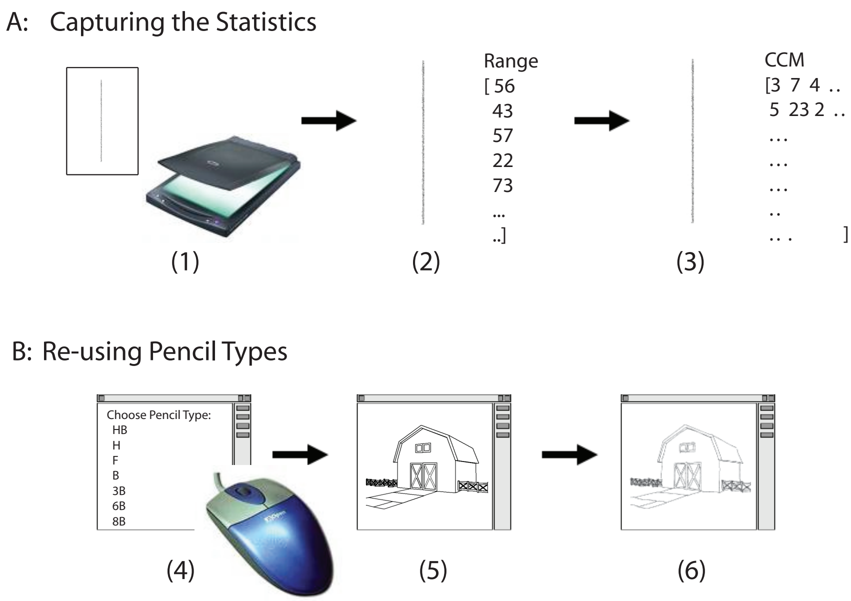


Figure 1.1: *An overview of the line drawing system. Part* A *(top), capturing of the pencil style from an original drawing by scanning it in (1), capturing the dynamic range of the pencil line (2); and (3) capturing the CCM of the line and storing it together with the histogram. The second part* B *(bottom), selecting a pencil type in (4); loading a 3D model into 2D space (5); and finally (6) the resulting illustration of applying the selected pencil type to the model.*

(Chapter 4).

The second step in the line generation process (phase *B*) can be summarized as follows; A pencil type is first acquired from the user, with the lines of the image to be drawn, then:

(1) The acquired set of statistical values for the pencil type selected are called (from the data stored in part one). (2) Point positions are used to calculate the trajectory for the movement. These points can be placed on the screen using mouse or tablet, or read in from

text files formatted to include start and end points of all segments (postscript files). (3) Pencil textures are then synthesized and placed along the line and modified to represent a real pencil texture more closely. (4) Then finally the textured line is output to the screen.

My approach uses the concept of path and style defined by Strothotte and Schlechtweg [2002], a human arm trajectory model by Flash and Hogan [1985], and a co-occurrence texture synthesis technique similar to that of Gagalowicz and Ma [1985] The system creates lines were no two generated lines are the same, path or texture.

I perform a distribution of texture across the resulting path aligned with its orientation, and use the pencil style co-occurrence matrix to make the line texture closely resemble a original pencil sample. The trajectory path requires creating a smooth spline along the line depending on the line length.

1.3 Contributions

The contribution of this work is to provide a high quality pencil media line reproduction agent for creating aesthetically pleasing lines that mimic human-drawn lines. For this purpose, I use basic methods of image synthesis and a model of human arm movement for replication of point-to-point arm trajectories.

Based on this approach the 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.

From this, convincing replications of pencil lines are achieved that have both the aesthetic quality and natural appearance of real human-drawn pencil lines.

This work has been published in *Computational Aesthetics (CAe) 2008*, and a poster of the work has been presented at the *6th Symposium* on *Non-Photorealisic Animation and Rendering (NPAR) 2008*.

1.4 Thesis Overview

The rest of this thesis is arranged as follows: Chapter 2 provides the background with a review of the related literature in line drawings, human arm movement theories, and texture synthesis. Chapter 3 describes the major stages of the human pencil line mimicking system; capturing the statistics (path and texture) of real-drawn lines from scanned in illustrations drawn from studies designed and implemented to observe how people draw, and generating paths using a human arm movement mathematical model. Chapter 4 explains the texture synthesis procedure and how it is generated along the path. Results are presented in Chapter 5. Chapter 6 the study done to verify the algorithms's reliability. Lastly, Chapter 7 finalizes my work by presenting the conclusions and directions for future research.

Chapter 2

Background and Previous Work

In this chapter I provide a basic background of techniques and methods used in this research. The papers reviewed here are classified into three categories: (1) research in human line drawing algorithms that use different style capturing techniques to achieve a replication of a similar stroke containing all of its feature characteristics; (2) mathematical models replicating unconstrained point-to-point human arm movement trajectories; (3) and finally texture synthesis methods used in research to reconstruct synthesized images from data take from natural scenes. Each section presents a brief description of how each method works and the reason why it was either just reviewed or adopted by this research.

2.1 Line Drawing

Here I provide background in methods and techniques commonly used to acquire near exact replications of line drawings. The papers reviewed here focus on methods of capturing style from recent advances in NPR research, for example, rendering-by-example, and graphite pencil rendering techniques for curves and line styles. I categorize these research fields as follows into three groups: (A) Capturing style characteristics Section 2.1.1, then (B) rendering-by-example methods Section 2.1.2, and (C) graphite pencil rendering methods Section 2.1.3.

2.1.1 Capturing Style Characteristics

Characteristics of lines in sketched and hand-drawn images have been studied closely in the field of non-photorealistic rendering (NPR) for just over a decade. Many algorithms that capture style characteristics have been developed based on analyzing basic drawing primitives. These algorithms tackled the generation of both path and texture separately and then provided techniques to conjoin the two. The algorithms discussed in the sections below studied and applied multiple parameters (such as length, width, pressure, etc.) to achieve stylized approximations of images and drawings that conform to user specified constraints.

2.1.1.1 Surfaces to Lines: Rendering Rich Line Drawings

Schlechtweg et al.'s [1998] "Rich Line Rendering" system provided a solution to map attributes of surfaces of an object onto lines which best represent it, through the use of a specialized rendering pipeline. Their aim was to allow a viewer to easily examine and understand an image without the distraction of unwanted information. The aim of their work was to solve the control of information overload in computer-generated presentations.

Schlechtweg et al.'s method was divided into two parts, a projection phase and an interpretation phase. The projection phase: projected the geometry of a model object from three dimensions into two dimensions producing an enriched 2D model. They achieved this by removing hidden lines and surfaces and computing lighting (by specifying an illumination model) and textural properties. Schlechtweg et al. then performed an interpretation of the enriched model by mapping the encoded surface attributes onto the style attributes as specified by the user.

The proposed *linestyle* concept was built on observing human illustrators at work. The line

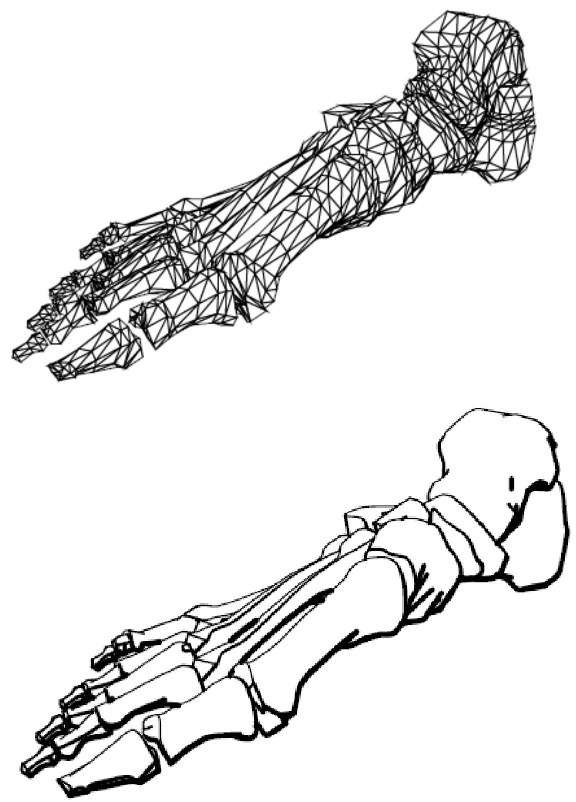


Figure 2.1: *Example result (below) achieved from the interpretation of light intensities as an attribute for line widths off an input (top) wire frame model. From Schlechtweg et al. [1998] Rich Line Rendering Pipeline. Copyright Schlechtweg et al. Used with permission.*

consists of two parts, (1) the path and thickness, (2) the deviations from the general path (caused by e. g. pen angle).

From this definition, Schlechtweg et al. modeled a technique to use style parameters and polygonal paths to create stylized lines in images. This resulted in a final sophisticated pen-like illustration as shown in Figure 2.1

My algorithm adopts the same concept of path and style as the basis for producing realistic pencil-drawn lines mimicking human-drawn lines. One draw back to the "Rich Line

Rendering" system was the scan-line based algorithm used to acquire the line segments. This was adopted from Sechrest and Greenberg [1981], and used in the projection phase. Their method could only handle one model at a time; intersecting models were classed as one when scanned causing a loss in the conveyed information. As described in Chapter 3, my rendering system only takes in start and endpoints of each single line segment, by doing this entangling issues of multiple objects in the same *2D* scene are resolved; hidden line removal is pre-computed in the input files. I adopt a somewhat similar texture filling algorithm as proposed by Schlechtweg et al. [1998], the algorithm will be explained in Chapter 4.

2.1.1.2 Drawing and Animation Using Skeletal Strokes

Hsu and Lee [1994] developed a rich framework for general image deformation, a drawing and animating system called *Skeletal Draw* that uses *skeletal strokes*. *Skeletal strokes* are basic drawing primitives that act as brush strokes, they do not use physical models (i. e. bristles of brushes or properties of paper) or repeated patterns as basic drawing units. To draw, an image and its deformation are abstracted into a skeletal stroke. These strokes were designed to overcome the constant thickness (computer strokes) seen in many of the painting programs, and create more expressive artistic strokes without the use of tracing or photo-editing methods.

The framework was created for general image deformation and only has one control parameter required which is the brush path itself. The structure found in Hsu and Lee was more general than those based on physical models (e. g. [Winkenbach and Salesin 1994; Curtis et al. 1997; Sousa and Buchanan 2000]). The only aim was to achieve a final acceptable appearance of the stroke, rather than the physical action of dragging a brush across the paper.

Figure 2.2: *Example drawing from Hsu et al.'s [Hsu and Lee 1994] Skeletal Strokes, mimicking Chinese brush painting. Copyright Hsu and Lee. Used with permission.*

The system also follows the concept of a base path which they call *reference backbone*, and reference thickness where the style image is placed on a path in such a way that it follows a *deformed* coordinate system defined by the path, width, and parameters like shear, angle and twist. Parts of the deformed images applied on the path are independently controlled. Figure 2.2 shows a result drawing using different types of skeletal strokes.

Hsu and Lee were able to avoid problems arising from concatenating images on a single path by only using one image. This image was mapped to a deformed coordinate system (the path) using a non localized deformation model to avoid effects such as the wrinkling of overlapping parts of the image.

My approach uses an on the fly synthetic texture rather than conventional methods. By using vector positions of the path between start and end points of line segments the algorithm directly places texture values onto the 2D image plane and synthesize accordingly. The deformation model could be adopted and would prove to be very useful if a conventional texture synthesis method is used.

2.1.2 Rendering-by-Example

Here I review multiple rendering-by-example methods used in NPR and graphics applications to acquire near-exact replication of stylized lines. The statistical matching techniques and multi-resolution algorithms applied in rendering-by-example methods to acquire replications of styles use large sample libraries and require scanning in illustrations of new styles when needed.

My research differed from rendering-by-example methods in that it does not use large style libraries. The aim of this research lies in its synthesis procedure which focuses on statistical collection of the graphite pencil texture and line style based on natural observation without the need of large sample databases. My algorithm is limited to two pieces of acquired texture information depending on the desired pencil softness, and uses it as a basis for the replication (re-generation) of the lines.

2.1.2.1 Statistical and matching methods

The WYSIWYG-NPR system by Kalnins et al. [2002] inputed multiple example strokes (three or four) and synthesized new but similar styles by introducing random variations, then repeated those styles around the silhouette of an object, as seen in Figure 2.3. Despite the fact that Kalnins et al. used a few example strokes to create a sample style, they captured the *base path* and *offsets* (small scale wiggles relative to the base path) using Markov random fields.

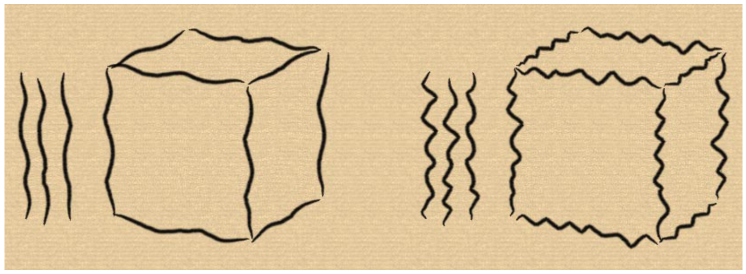
Applying stroke-based rendering on simulated natural media was avoided in my algorithm by using texture synthesis techniques.

Simhon and Dudek [2004] used a different approximation via hidden Markov models to statistically regenerate curve styles from example strokes onto illustrations. Jodoin et al. [2002] processed a set of hatched lines via statistical synthesis using Gibbs probability N-grams to generate similar strokes. Freeman et al. [2003] searched a multi-style library and acquires a matching line style using a k-nearest neighbour fitting algorithm and replaces the illustration style by the selected library style. Synthesis was achieved using locally weighted regression, a general learning method. Examples of the previously mentioned techniques are in Figure 2.3 and Figure 2.4.

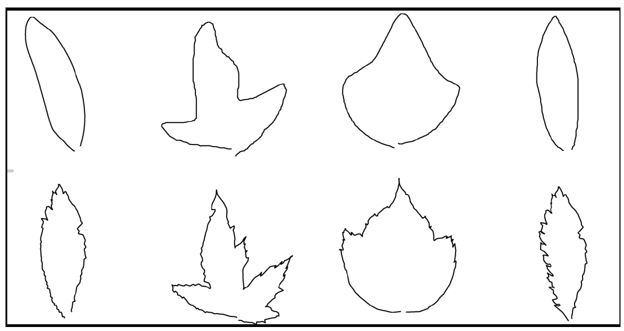
2.1.2.2 Multi-resolution methods

Other approaches to rendering-by-example are based on multi-resolution techniques. These capture the level of details within an input and rebuilds them directly onto an output with various modifications. For example, Hertzmann et al. [2002] used Gaussian pyramids to create complement curves that are analogous to a given pair of stokes. By providing two curves with a certain relation, he was able to extract the style in them and apply the same relationship onto other generated curves, see Figure 2.5.

Brunn [2006] applied a multi-resolution method adopted from Samavati and Bartels [1999] in order to capture styles for curve synthesis by separating the path from the style. His method inspired my work and further review into paths versus styles along with Schlechtweg et al.'s [1998] definitions. As shown in Figure 2.6. One more example by Finkelstein and Salesin [1994], used wavelets for smoothing purposes which required no extra storage.



(a) A cube rendered in two different styles. The Crease strokes were synthesized from the stroke examples shown.

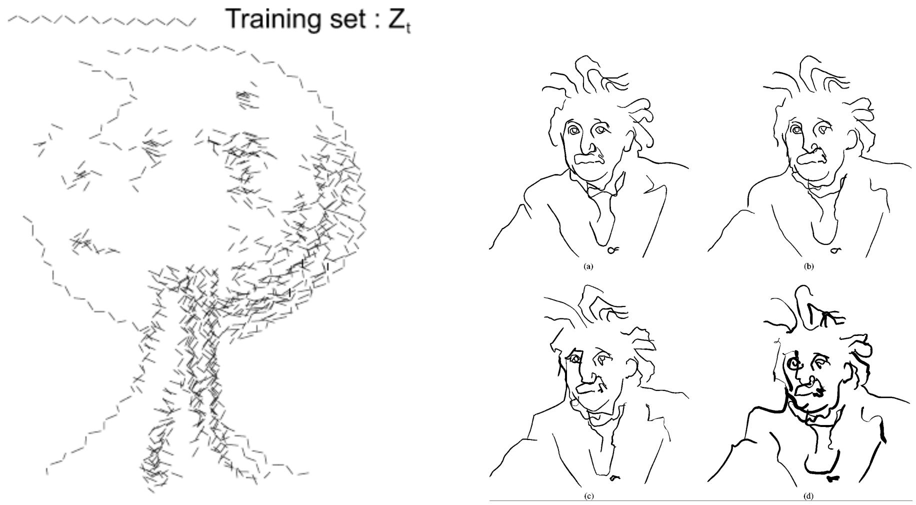


(b) Examples of synthesis using a leaf training set. The curves on the top are the input sample curves while curves on the bottom are the generated ones.

Figure 2.3: *Examples of statistical and matching methods. Copyright Kalnins et al. [2002], Simhon and Dudek [2004]. Used with permission.*

2.1.3 Graphite Pencil Rendering

My research was inspired by approaches that simulated pencils as a medium, specifically work by [Sousa and Buchanan 1999a; Sousa and Buchanan 1999b; Sousa and Buchanan 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. They used a wide range of different pencil softness and implemented a simulation of drawing paper texture, pencil, kneaded eraser and blender, drawing primitives



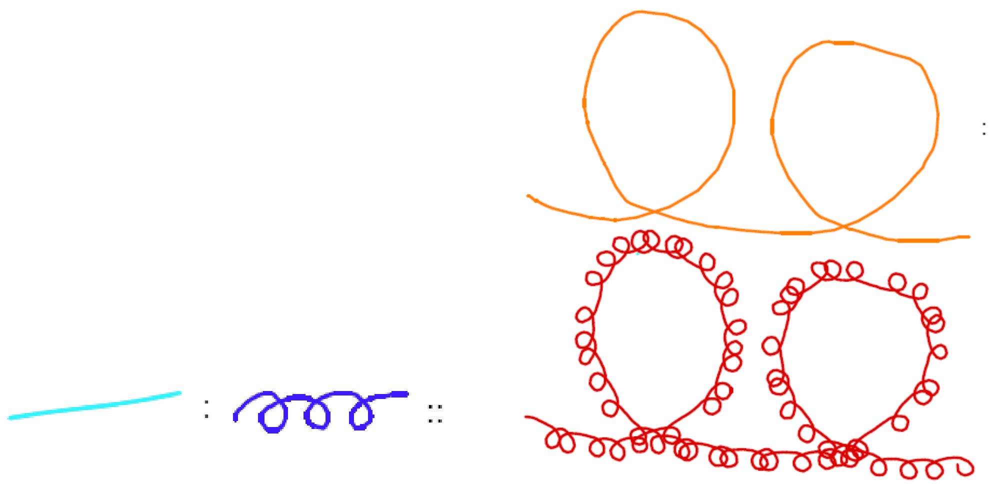
(a) (b)

Figure 2.4: *More examples of statistical and matching methods. (a) The image was generated interactively with a brush that generates stroke sequences similar to the training set Zt. From Jodoin et al. [2002]; (b) Presents is a style translation using the K-NN algorithm of Freeman et al. [2003] : (a) is the original line drawing; (b) 6-NN algorithm used to fit to image data in style 1. While not exactly the same as the original; (c) 6-NN algorithm image translation to style 2; (d) image data translated to style 3, where the line quality of the training data in that style is still maintained. Copyright Jodoin et al, and Freeman et al. Used with permission.*

(pencil strokes and mark-making), and a rendering technique for outlining and shading. Their system was an extension to their initial inspiration, concepts drawn from Vermeulen and Tanner [1989] paper on *PencilSketch*.

Figure 2.7 shows some of the art work produced by Sousa and Buchanan's method of simulating graphite pencil.

My research methods differ from the latter methods because the lines are not restricted to



(a) Curves A and A0, initial inputs

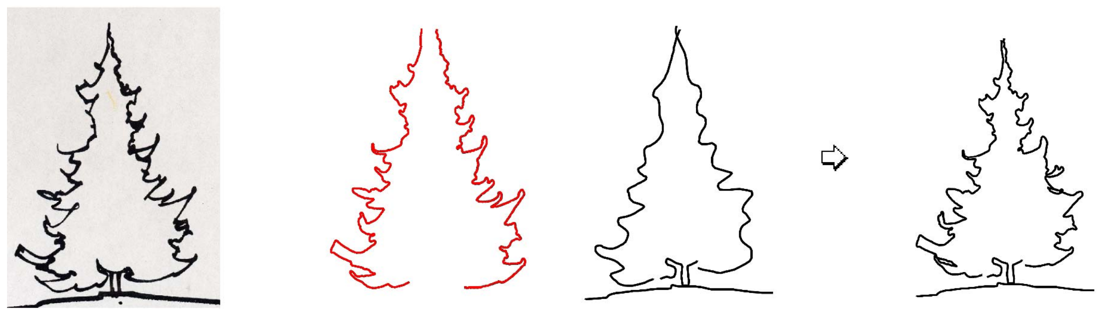
(b) Curves B and B0; curve B is used to apply the style transformation.

Figure 2.5: *Multi-resolution method by Hertzmann et al. [2002]. The figure shows curve A ((a) left) and an example filtered curve A0 ((a) right), an analogy is used to learn the transformation from A to A0, and then apply it to a new curve B ((b) top) to generate an output curve B0 ((b) bottom). The idea is to satisfy the relation A : A0 :: B : B0. Copyright Hertzmann et al. Used with permission.*

short strokes but can be of arbitrary length without any repeating segments. The statistical method currently lacks the preciseness in representing pressure along the textured path which Sousa and Buchanan [1999a] accommodated (which can be incorporated in the future), but offer an easier, and alternative approach to generating lines that pass for real lines which can be used in vector graphics programs.

Finally, my algorithm is limited to two pieces of acquired texture information depending on the desired pencil softness, which are used as a basis for the replication of the line textures.

The algorithm for drawing these lines could easily be incorporated into the approaches of Sousa and Buchanan [2000] and Salisbury et al. [1994], adding the benefit of a model



(a) left, original scanned illustration, and right, an extracted silhouette style in two parts.

(b) Shows a detailed base path following the original and result of style application

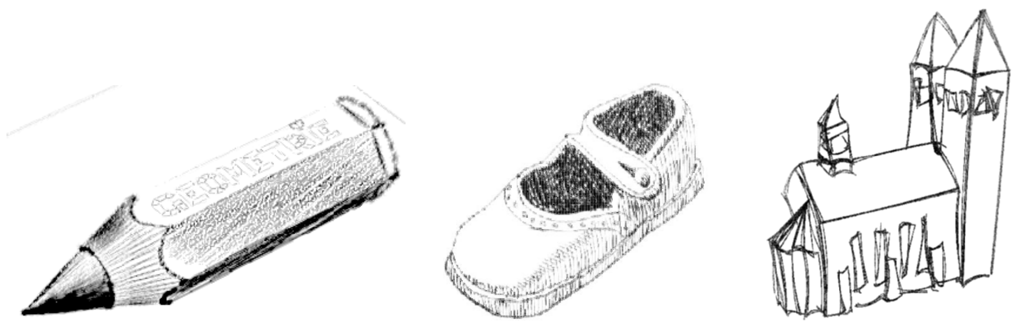
Figure 2.6: *Extraction and re-application of a coniferous tree silhouette style by Brunn [2006]. Copyright Brunn. Used with permission.*

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 lines.

2.2 Dynamic Optimization of Human Arm Movement

In order to produce a good simulation of the path of a human line drawing I have examined previous studies on the coordination of voluntary human arm movements. Human motor production have been analyzed, modeled and documented for well over the past century. In this section various mathematical models inspired to mimic human arm trajectory movements based on theories of the functionality of the human central nervous system are presented.

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, for example, 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], and Plamondon



(a) (b) (c)

Figure 2.7: *Graphite pencil rendering system examples. (a) Is an automatic rendering of a shoe using 3B pencil rendered by Sousa and Buchanan [2000]; (b) Is an example of pencil rendering of 3D pencil object (602 edges, 206 faces) using mass shading, in 2B pencil used lightly over semi-rough paper, rendered by Sousa and Buchanan [1999b]; (c) Is a rendering of a 3D church model over semi-rough paper with a uniform with 2B pencil by Sousa and Buchanan [1999a]. Copyright Sousa et al. Used with permission.*

[1995]. One of the features of human multi-joint arm movement is that the path of the hand between two points is roughly straight, and its speed profile is approximately bell shaped.

The CNS has been proven to be responsible for the control of normal multi-joint movements by conducting many different strategies to control skilled activities, Woodworth [1899].

According to the various 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;
- Simple motor control is achieved by planning hand trajectories in extra-corporal space and joint rotations are tailored to produce the desired or intended hand movements.

Below I present an overview of the three most prominent human arm trajectory formation models developed to explain either of the two CNS theories, and I discuss why I chose the Flash and Hogan model to be implemented in the system.

Any of the models summarized below could be incorporated into the proposed system. This can be done by modeling the arm, the muscles, the joints and determining parameters to apply once the initial and final positions of the path are defined.

2.2.1 Uno's Model

Uno et al. [1989] presented a trajectory planning and control model for voluntary human arm movements. The aim of the Uno et al.'s model was to create the smoothest possible trajectory, and implied that the optimal motor commands (torques and forces) are directly obtained from the dynamics of the musculoskeletal system. Based on the fact that movement optimization (optimal trajectory) is related to movement dynamics, Uno et al. proposed a model that is a quadratic measure of performance for any possible movement (square of the rate of change of torque) integrated over the entire movement from initial to end positions.

$$C_\tau = 1/2 \int_0^{t_f} \sum_{i=1}^{m} (\frac{d\tau_i}{dt})^2 dt, \tag{2.1}$$

where τ_i is the torque generated by the *i-th* actuator (muscle) out of m actuators and t_f is the movement durations. This function depended entirely on the complex non-linear dynamics of the musculoskeletal system, which made it difficult to determine the unique trajectory that yields the best performance.

The trajectory derived from the minimum torque change model depended on the geometrical relationships between the initial and end positions of the movement trajectory, the arm

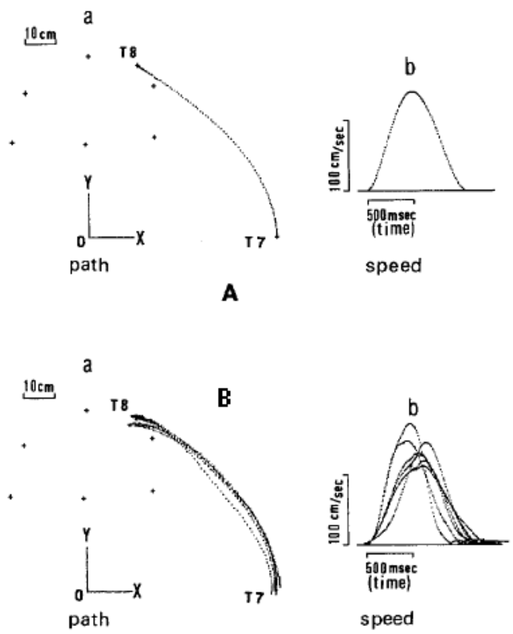


Figure 2.8: *Uno et al. [1989] results. Large free movements between two targets (T7− >T8); the starting posture is stretching an arm in the side direction and the end point is approximately in front of the body. (A) a hand trajectory predicted by the minimum torque-change model,* a *shows the path and* b *shows the corresponding speed profile. (B) show observed hand trajectories of seven subjects, (*a*) shows the paths and (*b*) shows the corresponding speed profiles.*

posture (the location of the shoulder relative to these points), and on external forces. Hence Uno et al.'s model predicted different formed paths depending on orientation of the stroke.

This model closely represents true human arm movement but required the estimation of the physical parameters to reach a dynamic model of the arm, then numerically estimated the joint torques. Further, to calculate a unique trajectory, extra numerical methods needed to be used. Because of this, including the time complexity to numerically process all the predictions that have to be reset at the start of every movement and the complexity of setting the musculoskeletal system parameters, I did not choose this model for the algorithm.

2.2.2 Plamondon's Model

Plamondon [1995] presented a kinematic theory based on applied studies of simple human movements; these can be described in the velocity domain as the response of a synergistic action of an agonist and an antagonist neuromuscular network, Plamondon [1995]. Plamondon's theory to produce arm movements by modeling the impulse responses to neuromuscular activities, resulted in a system that produces a close proximity bell-shaped velocity profile to represent an entire point-to-point movement.

Each network is composed of a large set of coupled neuromuscular subsystems that react to an input command $D1$ (for the agonist) and $D2$ (for the antagonist) with an impulse response that can be described by a lognormal function [Plamondon 1995].

Each lognormal impulse response $\bigwedge(t, t_0, \mu_i, \sigma_i^2)$ has three parameters: the starting time t_0, the parameter i which reflects its logtime delay, and μ and σ which reflects its log-response time Plamondon [1995]. The resulting curvilinear velocity V(t) of a single movement is described by calculating the delta-lognormal response; subtracting weighted impulse response of the antagonist network from the agonist one see Equation 2.2. The angular direction of the velocity vector follows immediately below.

$$V_\sigma t = (D_1\Lambda(t;t_0,\mu_1,\sigma_1^2) - (D_2\Lambda(t;t_0,\mu_2,\sigma_2^2)\theta_t = \theta_0 + C_0\int_{t_0}^{t}(V(\tau)d\tau \qquad (2.2)$$

A single movement, also called a stroke i, is represented in the space and velocity domains by a velocity vector starting at time $t_0(i)$ at point $P_0(i)$ with an initial direction $\theta_0(i)$, and moving along a circular path of length $D_1(i)$-$D_2(i)$ with a constant curvature $C_0(i)$.

According to the kinematic theory, the movement described by this model will reach its target with a movement time that is proportional to the ratio of the agonist and antagonist commands, $\frac{D_1(i)}{D_2(i)}$ [Plamondon 1995].

Plamondon's model can be incorporated into my model. It was not adopted for this research because it required considerable effort to organize based on its structure, and it needed detailed effort to select the correct parameter values for the agonist and antagonist muscles of the whole arm. This caused it to be slower than the Flash and Hogan model and increased the likely-hood that the generated paths were not unique. My research aimed to generate unique paths to be textured and presented as real hand drawn lines, adopting Plamondon's methodology would not provide the best results.

2.2.3 Flash and Hogan's Model

The Flash and Hogan model Flash and Hogan [1985] solved for trajectory formation, of moving from a start position to an end position. It did this by separating trajectory planning from trajectory execution. In a visually guided reaching movement, the position of the target object is provided by the visual system. The current position of the hand is measured by the visual and/or somatosensory system Zelaznik [1996]. The time it takes the hand to move from initial to final target position should be selected within a reasonable range. From this an infinite number of possible trajectories connect the initial and final positions for the given time. Trajectory is defined in this sense as path shape including a velocity profile.

Flash and Hogan [1985] introduce the minimum jerk model which assumes that the trajectory followed by a subject's arm minimizes the square of the movement jerk (rate of change of acceleration) integrated over the entire movement from start to end positions.

$$C_j = 1/2 \int_0^{t_f} ((d^3X/dt^3) + (d^3Y/dt^3))^2 dt \tag{2.3}$$

Equation 2.3 shows the criterion function used to optimize the arm movements. Where X, Y are the cartesian coordinates of the hand, and t_f was the movement duration. They 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. It was successful only when formulated in terms of hand positions in extra-corporal space, and failed when defined in terms of joint angles, because the jerk model predicts straight trajectories in the space where the objective function is defined.

My algorithm adopts the Flash and Hogan model because the model represents human arm trajectory in a planar field, similar to the movement of a unconstrained human hand guiding a pencil across a piece of paper, see Figure 2.9. Also because of its simplicity and power in producing unique optimal solutions analytically, by only supplying it with a movement time and kinematic initial and final positions. The model not only reproduced the qualitative features, but also the quantitative details of trajectories. Hence this model served the purposes of regenerating realistic point-to-point trajectories. The only drawback is that this model obtained a single unique solution regardless of the environmental conditions or the task at hand.

I present a complete description of the model used in this research and include a proposed modification in Chapter 3 which includes some environmental conditions into the model based on the user studies and my own observations.

2.3 Texture Synthesis

A co-occurrence matrix can be defined as "the proportion of occurrence of pairs of luminesces when their locations differ by a delta (a displacement) in the texture field plane".

The papers I review here focus on texture synthesis techniques based on texture similarity

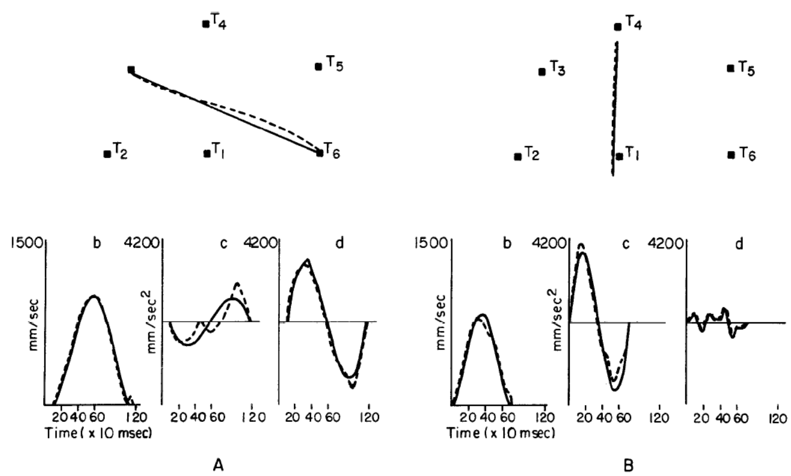


Figure 2.9: *Flash and Hogan overlapped predictions (solid lines) and measured (dashed lines) hand paths (a), speed (b), and acceleration components along the y-axis (c), and along the x-axis (d) for two unconstrained point to point movements. (A), a movement between targets 3 and 6. (B) a movement between targets 1 and 4.*

metrics. They are classified as one class depending on their major contribution, because they agreed on the basic texture acquiring methods. They all used the co-occurrence matrix as basis for reproducing a similar generated image but differed slightly in the way they implemented it. Here are three methods with explanation as to why they are successful and how they inspired this research.

2.3.1 Sequential Synthesis of Natural Textures By Gagalowicz

This research is influenced by an idea for texture synthesis based upon a sequential synthesis method introduced by Gagalowicz and Ma [1985], that used a priori-given second-order statistics (second-order spatial averages or autocorrelation parameters) of a natural texture field as input, and once processed, an output with the same second-order statistics is outputted. They assumed that the synthesized texture and the original were to be *identical* if they were perceived visually as identical.

A procedure was given to synthesize an artificial texture field in such a way that its second-order statistics were equal to the desired ones. Gagalowicz and Ma [1985] aimed to discover the nature and number of the elements characterizing a texture, determine the resolution of a texture, propose a model for textures based upon these characteristic elements, and finally use that model to synthesize textures. They accomplished this by first acquiring a first-order probability distribution for all pixels in the image as in Equation 2.4.

$$Px_m = L_1 \tag{2.4}$$

where L_1 is the grey value at pixel location x_m. Then they go on to find the second-order probability distributions of all location (x_i, y_i) defined as in Equation 2.5.

$$Px_{m_1} = L_1, x_{m_2} = L_2 \tag{2.5}$$

where $x_{m_1} = L_1$ is the probability that x is equal to L_1 at location m_1, and $x_{m_2} = L_2$ is the, probability that x is equal to L_2 at location m_2 for all possible pairs of points (m_1, m_2) and all grey level values L_1 and L_2

The final information gathering is done by finding the second-order co-occurrence probabilities $P_\Delta(L_1, L_2)$ which were determined using a translation component Δ in the discrete plane N^2. Were $\forall\ \Delta$ $(\Delta = \Delta_{x,\Delta_y})$, $P_\Delta(L_1, L_2)$ was the probability occurrence of a grey level L_1 and a grey level L_2 when their respective locations differ by Δ as shown in Equation 2.6, where I was the number of pairs of points (i,i+Δ) in the texture field plane.

$$P_\Delta L_1, L_2 = \frac{i}{I} \sum_{i=1}^{I} P(x_i = L_1, x_{i+\Delta} = L_2) \tag{2.6}$$

the second-order spatial average were then computed $P_\Delta(L_1,L_2),\forall L_1,L_2$ and used to fill the co-occurrence matrix $M\Delta$. Then using an iterative process over the image, Gagalowicz and Ma constructed a texture field Tx such that its feature vector B^{Tx} (vector of CCM) would be equal to a priori-given vector B. Synthesis is achieved by first creating a texture which is a realization of homogenous white noise, that histogram was equal to the desired one, and then compute directly from any co-occurrence matrix M_Δ the desired vector B from the spatial averages previously computed. The texture was then modified sequentially point-by-point to minimize the square error between B and B^{Tx}, and the problem was solved once it converged to zero.

This method allowed the synthesis of grey tone texture fields while 'controlling' their second-order statistics in rather large neighborhoods. The synthesized textures were very similar visually to the original natural textures used to compute the second-order statistics, but second-order spatial averages gave better results than autocorrelation parameters. This seemed to strongly support the conjecture that the visual system is only sensitive to the second-order spatial averages of a given texture field, and that these statistics should be used to model textures.

The strategy of Gagalowicz and Ma [1985] was what my research intended to implement; to acquire realistic looking textures that resemble natural graphite material of pencils. The synthesis algorithm follows Gagalowicz et al.'s steps but differs in the synthesis method of the complete texture image. Gagalowicz and Ma [1985] concatenated statistical pieces of the image on an image grid to result in a feature vector that was used to hold the co-occurrence values, while my algorithm performs an incremental stepping pattern from one side of the image to the other using neighbourhoods to apply the co-occurrence values and achieve texture similarity. Some accuracy is traded for efficiency, since the original method is rather slow.

2.3.2 Texture Synthesis using Grey Level Co-occurrence Models by Copeland

Copeland et al. [2001] presented multiple texture synthesis algorithms based on grey-level co-occurrence (GLC) model of a texture field. These algorithms use a texture similarity metric, which was shown to have high correlation with human perception of textures. The synthesis algorithms were evaluated using an extensive experimental analysis method. These experiments were designed to compare various iterative algorithms for synthesizing a random texture which were discussed extensively in their paper.

Copeland et al. [2001] judged the quality of the synthesized result in term of visual similarity to the target texture as well as numerical measure of the error between the GLC models of the synthesized texture and the target texture.

They acquired the statistics needed from a model texture by calculating the second-order grey level probability distribution (also called the Gray Level Co-occurrence (GLC) probability) of the image.

The resulting GLC probabilities for any single displacement in the image are normally tabulated in the form of *GxG* matrix, with i as the row index and j as the column index. They then use a texture model developed based on Gagalowicz and Ma [1985]'s texture model.

To measure the correlation between the images Copeland et al. [2001] used a simple average co-occurrence error *ACE* which had high correlated with human judgments of the visual distances between different textures. *ACE* was defined as:

$$ACE = \frac{1}{T_{NGLC}} \sum_{\Delta \in D} \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} |P_t(i,j|\Delta) - P_b(i,j|\Delta)| \tag{2.7}$$

Copeland et al. [2001] conducted studies on three types of test cases for the texture synthesis experiments, on a disordered, strongly ordered, and weakly ordered material. The

Figure 2.10: *Cliques. Point pairs are called cliques, and pairs with the same relative position form a so called cliquetype Zalesny and Gool [2001]*

psychophysical experiments conducted on the synthesized images demonstrated that the results for the texture synthesis algorithms have high correlation with the texture similarities perceived by human observers.

My method follows on from their technique, mimicking the second order statistics extraction but differs in the implementation of the synthesis itself. Due to the time complexity of their synthesis procedure I developed a more compact version which allows for faster synthesis.

2.3.3 A Compact Model for Viewpoint Dependant Texture Synthesis by Zalesny

Zalesny and Gool [2001] introduced a texture synthesis method that generated similar texture from an example image. It was based on the emulation of simple but rather carefully chosen image intensity statistics.

Zalesny and Gool [2001] method started off by collecting the first order statistics (an intensity histogram), then they extracted the co-occurrence matrix (CCM) by acquiring pairs of relative pixel intensities directly from the model texture using their proposed capturing method called 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 Figure 2.10. The CCM only stored the distribution of intensity differences between the heads and tails pixels for a given orientation.

Clique types were first selected by determining the neighbourhood system, then used to collect the pairwise statistics up to a maximal length. This length was a head-length acting as the maximal displacement in any given direction; their maximum was about (45 pixels).

The conventional way of calculating the CCM was by summing all the joint probabilities of intensities for a given pixel neighborhood into their relative position in the CCM.

The synthesis steps are simple and direct but required large processing time.

Zalesny et al.'s method initially collected the first and second order statistics from the model texture, then generated an image with random noise in the range of the model texture and collected the pairwise statistics of all clique types (the noise). Next, they calculated the Euclidean distance between the distribution of the example texture and the synthesized texture. And finally, they synthesized a new texture using the updated neighbourhood system and parameter set, repeating all the steps again until the minimum difference was achieved.

My research did not adopt this type of synthesis procedure due to the time complexity. The clique criteria proves very efficient for larger images, but for thinner pencil line textures clique types are limited and line lengths continuously set boundaries on the overall second-order statistical collection scheme.

Chapter 3

Generating The Path

There are two parts to my work: the first generates a realistic path; and the second synthesizes a suitable texture to mimic a human line using a specific pencil for that path. In this chapter I will focus on the first part, generating a realistic path. Section 3.1 covers the observations I notice when conducting user studies for capturing the line drawing process; Section 3.2 covers the steps I follow to model a path realistically based on the Flash and Hogan mathematical model, including an extension based on variation observations in Section 3.2.1. The texture synthesis algorithm follows in Chapter 4.

The algorithm operates upon two points, representing an initial and a final position of a the segment. The points can be entered into the system using one of three methods: (1) by interactively entering them through the software interface; (2) reading in input files (such as postscript); and (3) by parsing a 3D model. Every point pair is used to specify the start and end position of the path to be generated by the Flash and Hogan model (Section 3.2). The mathematical model produces the trajectory (the path) control points that are then used as input for a Catmull-Rom spline.

3.1 Initial User Study: Line Drawing Observations

In this section I discuss a study conducted on how people draw using pencils. During the review of background information for this research, it became apparent that a carefully designed user study would allow the examination and comparison of new observations with results previously gathered by models like, ([Sousa and Buchanan 1999b], [Schlechtweg et al. 1998], and [Kalnins et al. 2002]).

I designed a study, provided in Appendix B, which is used to acquire line data. The main aim of the study was to capture the following:

- data of pencil texture and path deviations from various path orientations;
- how pencil texture appeared for different line orientations;
- whether pencil texture properties can be classified into groups depending on line orientation;
- how the human arm performs its movement when drawing different line orientations; and
- how much the line path is affected by the human arm movement.

With ethics approval to conduct this type of research, I accepted twelve volunteer students and faculty from the Computer Science Department at the University of Victoria. The approval form, verbal script used, and the participant consent form are available in Appendix A.

A video recorder is used to capture the complete arm movements and drawing process of the participants. This proves very useful when making the decision of whether or not to use a mathematical arm movement model to replicate human movement.

The study takes on average *three* minutes to complete. The participants are seated in front of a horizontal desk at a reasonable distance, and are asked to: (1) sit up right, (2) hold the pencil steadily, (3) draw lines at a comfortable pace, (4) not to move the papers position throughout the experiment, (5) not to draw over the lines once already done, (6) not to erase anything, and (7) to adhere to starting and ending points. A *Circle* defines a starting point and a *Square* defines an ending point. Each study paper is placed in a frame created on the table to ensure that the user kept the paper in one position as he/she drew. The types of lines that the user drew ranged from horizontal, vertical, and angled lines of three different lengths (short, medium, and long).

The study is divided into two parts, the first was for straight lines, and the second was for curved lines. The curved line analysis is left for further analysis in future work. The objective assigned to participants for both parts (straight and curved) is to:

1. draw straight/curved lines between the given dots.
2. not use more than one dot twice.
3. adhere to dot starting and ending points.

Sample line figures were placed on each sheet to give the participant an exact idea of what was expected. The ideas I collect from this experiment formed the basis of the Human Line Algorithm.

When observing straight lines drawn on paper in random orientations from a starting position to an ending position, I noticed the lack of invariance of the paths based on orientation to a certain extent. The amount of deviation found in paths for different line orientations were approximately equal and a large observational significance was not noticed. Some characteristics were observed and left for further investigation in future work.

Flash and Hogan evaluated their model by comparing their simulated trajectories to measured hand trajectories. Planar horizontal arm movements were recorded using an apparatus, Figure 3.1 that involved a horizontal table and a mechanical two link manipulandum that the user held with their hand. The shoulders were restrained in all the experiments. By using precision potentiometers the joint angles of the apparatus were measured. Data analysis of this information was done to compute the hand position, and joint angles. Lagrange polynomial differentiation was used to calculate joint angular velocities, hand velocities and curvature of the path.

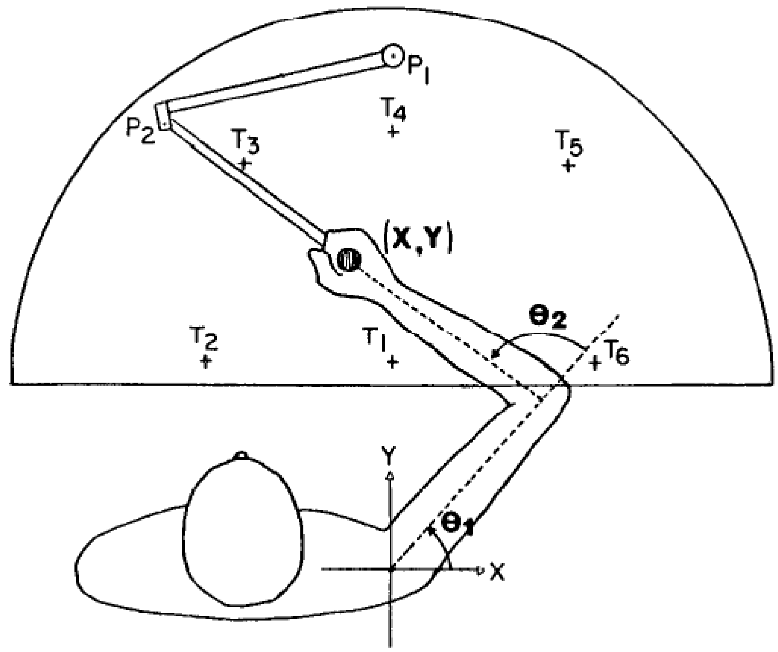


Figure 3.1: *Apparatus used by Flash and Hogan to measure arm trajectories in a horizontal plane. Movement of the hand was measured using potentiometers P1,P2.*

Flash and Hogan [1985] specifically noted the lack of invariance of joint trajectories to translation and rotation of the movements compared to the invariance of hand paths and speed profiles, due to the assumptions they made in the initial optimization stage. Very little discussion was made as to how the lines were actually perceived and how the observational conclusions were drawn.

I verify this observation with a second user study described in Chapter 6. It is not based on neuro-muscular limitations or facts about how humans move their arms, but solely on observational conclusions.

After the study is complete, the participants are asked to fill out a questionnaire. The questions includes some inquiry into background, age, artistic ability, etc. But non of this information is used in the development of the algorithm.

3.2 Model Path Trajectory

The algorithm constructs a path that conforms to a human arm trajectory using the method described in Flash and Hogan [1985]. I use this method as it provides a simple model to give "the smoothest motion to bring the hand from an initial position to the final position in a given time" [Flash and Hogan 1985]. The goal is to simulate a hand drawn path only given two points.

The mathematical model matches the observed human planar two-joint arm movements from Flash and Hogan's experiments. The organization of the movements was modeled through the criterion function . From the optimization of the criterion function, explicit analytic expressions describing the position of the hand along a trajectory were derived, see Equation 2.3.

The Flash and Hogan model offered a closed form solution that produced trajectories that are: (1) invariant under translation and rotation and (2) whose shape does not change with amplitude or duration of the movement.

All of these characteristics are based on observations of low frequency movements of human subjects. No high frequency movements were observed or implemented in the Flash and Hogan model. My work adopts the same assumptions.

The Flash and Hogan [1985] mathematical model satisfies my criteria in that it leads to a fast and interactive line path algorithm by providing a realistic overall velocity and acceleration profile and trajectory. I conducted user studies to explore whether line orientation, missing from the mathematical model, plays a role in the ability to reproduce human-like lines. The results of the studies indicated that the disparity of the line orientation was not noticed by participants and did not effect the classification criteria of real versus computer-generated drawings used by participants Chapter 6.

The paths are defined by Equation 3.1, as a fifth degree polynomial:

$$
\begin{aligned}
x(t) &= x_0 + (x_0 - x_f)(15\tau^4 - 6\tau^5 - 10\tau^3) + D \\
y(t) &= y_0 + (y_0 - y_f)(15\tau^4 - 6\tau^5 - 10\tau^3) + D \qquad (3.1) \\
\textit{where}, & \\
\tau &= \frac{t}{t_f},\ \ t \in [0,2] \\
D &= n,\ \ n \in [-5,5],
\end{aligned}
$$

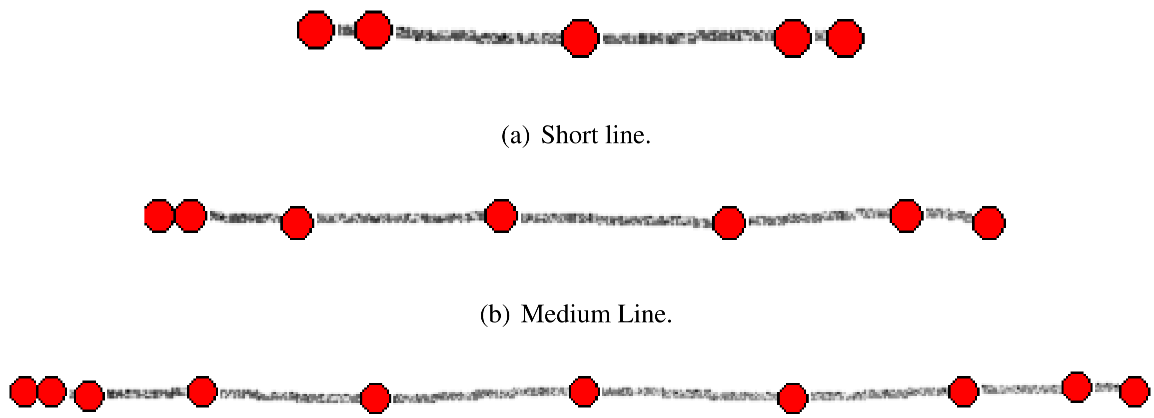
in which x_0, y_0 are the initial hand position coordinates at time increment t, and x_f, y_f are the final hand positions at $t = t_f$. The value of t varies from $[0,2]$ from the start position t_0 to end position t_{final}. And D is the deviation (*squiggle*) parameter discussed in Section 3.2.1.

The path to be generated depends on the distance between the two points, and a time step given to generate trajectory points along the path defining the trajectory of the movement.

I found empirically through experimental tests that $t_{final} = 2$ provides satisfactory results. The resulting trajectory points produced from the polynomial Equation 3.1 are well proportioned along the trajectory and offer enough spacing between every two instances of points that allows for maintaining a smooth path when introducing variation, as discussed later.

The trajectory control points found along the trajectory are then used as input for a Catmull-Rom interpolating spline, Shirley et al. [2005]. The number of control points (defined by the time step, Δt) depend on the length of the line.

Experimental evidence in Flash and Hogan [1985] showed that short hand drawn lines were perceptually closer to straight lines and longer lines have more variation. I conducted experiments to find reasonable values for Δt and provide the results in Table 3.1.



(a) Short line.

(b) Medium Line.

(c) Long line.

Figure 3.2: *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.*

Figure 3.2 shows the positions of the control points for three lines of varying length using each of the prescribed values for Δt.

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

Table 3.1: *Empirical Values for the time step* Δt.

3.2.1 Path Deviation

Flash and Hogan conducted studies to generate trajectories, with the participants seated upright holding a two-link manipulandum and recorded their movements accordingly. The participant's shoulders were restrained throughout the experiments, which in turn constrained the trajectory drawn.

I conducted a pilot study in which the participants were seated upright in front of a horizontal table with no constraints, from Section 3.1. Each participant drew a number of pencil lines between pairs of points. The data collected shows considerably more variation from the center line than can be accounted for with the variation of trajectories calculated by Equation 3.1.

To represent this variation, I introduce a deviational parameter called *squiggle* as an extension to the Flash and Hogan model. The squiggle as defined by Strothotte and Schlechtweg [2002] acts as a control variable; by setting its value to a maximal amplitude random displacements are used to give the appearance of a more irregular path. This additional term is necessary to provide sufficient variation in the line paths based on my observations of human drawn drawings. Experiments were conducted to validate this choice, Chapter 6 provides details on the experiments that were conducted. I define *squiggle* as a variable, D, for controlling the deviation normal to the Catmull-Rom spline at each of the lines control points also similar to methods used by Strothotte and Schlechtweg [2002]. The magnitude of the *squiggle* parameter controls the magnitude of random displacements applied to the Catmull-Rom spline controls points and therefore strongly influences the appearance of the path.

The deviational value is applied approximately normal to the Catmull-Rom spline at each of its control points. The variable D, see Equation 3.1, varies randomly in order to provide variation along the path; Empirically I found a range [-5,+5] worked for the lines regardless of length and produces independent displacement values for each control point. The appearance of the deviation factor can be noticed in Table 3.2.

Path Without Texture	Path With Texture
Path Without Squiggle	Path With Squiggle
Path Without Catmull-Rom Spline	Path With Catmull-Rom Spline
	

Table 3.2: *This figure shows the trajectories generated (a) without Catmull-Rom spline, (b) with a Catmull-Rom interpolating spline incorporated.*

3.2.2 More on the Flash and Hogan mathematical model

To create a modeling scheme for minimum-jerk unconstrained point to point trajectories Flash and Hogan derived a relationship between the maximum hand speed (V_{max}), and the ratio of movement amplitude (A) to movement duration (t_f):

$$V_{max} = C\,\frac{A}{t_f} \tag{3.2}$$

where,

$$C = 1.875$$

Flash and Hogan [1985] measured the mean value of C (the criterion function) from 30 measured unconstrained point to point movements ranging up to twenty five centimeters in length, to be equal to 1.805; the mean error of C was 3.7% of the predicted value and the standard deviation was 0.153, 8.2% of its predicted value. The velocity measure were accurate to 4%, which ultimately prove that model and observations agree within experimental error.

Chapter 4

Capturing The Style: Texture Extraction and Analysis

In this chapter I describe the second half of the Human Line Algorithm (HLA); it includes texture observations derived from a user study and an algorithm for texture synthesis. I propose a texture synthesis procedure to produce aesthetically pleasing pencil line textures mimicking human drawn ones. In order to understand how to shade or texture these paths, I conduct an observational study on the lines gathered in the initial user study Section 3.1.

My approach is based on an observational model of graphite pencil on paper. The emphasis is on the graphite texture distribution and natural appeal, not on the paper quality and attributes to graphite deposits. The goal was to capture the natural appearance of graphite texture and develop a way to semi-accurately display them through a generated process.

The texture synthesis algorithm is divided into a two step process, as seen in Figure 4.1. Phase (A) gathers information from a wide range of graphite pencils, from soft to hard from scans of human drawn lines to create a statistical profile. I analyze the lines for their statistical information and store them as histograms and co-occurrence matrices. A co-occurrence matrix is defined as the tabulation of how often different combinations of pixel values (intensities) occur in any given image. Phase (B) takes in two inputs, a pencil type selection and the lines of the image to be rendered.

My algorithm creates the line texture after the path is created by distributing random grey values along the path, applying the CCM for the pencil type, and then applying a Gaussian filter.

The texturing method used here is inspired by the use of second order statistics, also called

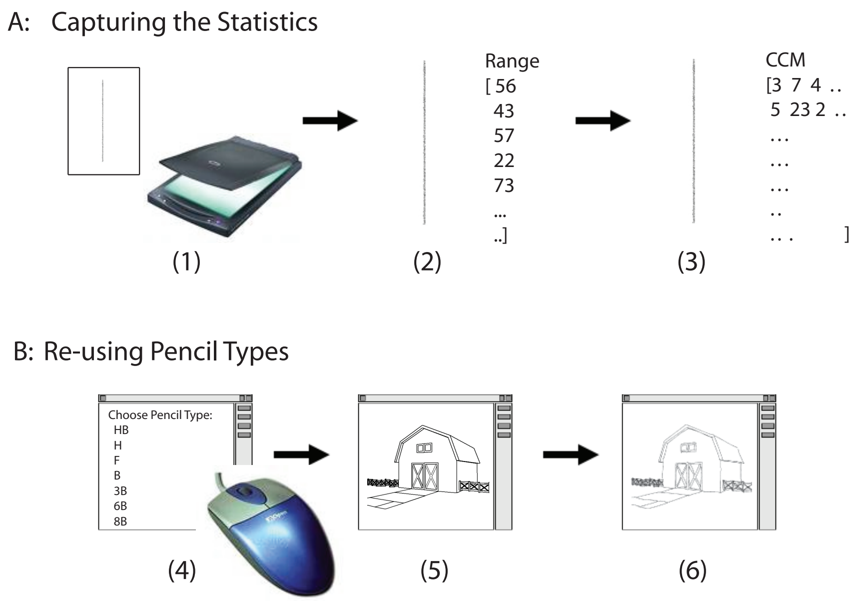


Figure 4.1: *An overview of the texture extraction and synthesis algorithm overflow. Part* A *(top), a one step process, capturing of the pencil style from an original drawing by scanning it in (1), capturing the dynamic range of the pencil line (2); and (3) capturing the CCM of the line and storing it together with the histogram. The second part* B *(bottom), selecting a pencil type in (4); reading in input data (5); and finally (6) the resulting rendered image.*

grey-level co-occurrence matrices (GLCM)([Gagalowicz and Ma 1985],[Copeland et al. 2001],[Zalesny and Gool 2001]).

Verification later reveals that the synthesized textures represent a striking similarity to human made ones Chapter 6.

4.1 Texture Extraction

Graphite pencils depending on their type are made up of combinations of different levels of graphite, clay and wax. The percentage of graphite and clay determine the hardness of the pencil. There are nineteen different types of pencils ranging from 9*H* to 8*B* depending on their hardness factor. The more graphite a pencil has, the softer and thicker that pencil becomes. I selected to base the work on classic wood-encased graphite pencils. This choice is sufficient for the purpose of generating human-like drawn pencil lines, as other pencils are more fitted for artistic use and not commonly used.

For each pencil type, I conduct a one time analysis on example lines in order to create a statistical profile. Texture samples for simulating hand drawn lines are obtained by scanning and analyzing the statistical properties of example lines drawn using a wide range of graphite pencils of the following specified types: *2H, H, HB, F, B, 3B, 6B, 8B*, see phase *A* in the system overview in Figure 4.1.

The pencil textures used for this extraction were acquired on plain, 20*lb* long grain, 100% recycled, *A*4 smooth white paper. I follow a simple set of procedures to ensure the extraction of all the textural properties needed for the reconstruction of the synthesized texture. They include multiple dynamic range histograms and co-occurrence matrices used in Section 4.2.

The following summarizes the steps that are taken to correctly capture the textural information properties of scanned in model textures:

- First, determine the range of grey levels for pixels along the approximate centre of the pencil line (non-repeating catmull-rom control points) and record them in a histogram within the range $[min_{mid}, max_{mid}]$. This serves as a starting point to imitate the bell-shaped curve property noticed when analyzing the statistical information of the sample pencil textures, later discussed in Section 4.2.

- Next, determine the histogram of the grey levels for the complete line image, assuming that white pixels surround each line segment when scanned in; since very few white pixels appear inside the line image, I do not consider white in the histogram. The histogram records intensities in the range $[min_{range}, max_{range}]$.
- Finally, compute the grey level co-occurrence matrix $(GLCM)$ for the line image, a detailed description of this process is given below.

The principle reason why I used second order statistics to synthesize lines resulted from studies done by Copeland et al. [2001]. They showed their work on psychophysical experiments of synthesis algorithms which proved that human vision is sensitive to second order statistics, were a texture similarity metric was shown to have a high correlation with human perception of textures.

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 neighbour in the positive y-direction, $(p,q+1)$ with value $j \in [0,255]$. The values (i,j) are indices to the appropriate cell of the co-occurrence matrix C defined over an n x m image I, parameterized by an offset (Δx, Δy) as in Equation 4.1:

$$C(i,j) = \sum_{p=1}^{n} \sum_{p=1}^{m} \begin{cases} 1 & if \;\; I(p,q) = i \;\; and \\ & I(p+\Delta x, q+\Delta y) = j \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

The *GLC* probabilities for any single displacement are tabulated in the form of a 256×256 matrix, with i serving as the row index and j serving as the column index. The next section describes the texture synthesis procedure built upon the information gathered here, where I use the *GLCM* to manipulate a random set of grey pixels to appear closer in similarity to the existing model texture.

4.2 Texture Synthesis

I use the dynamic range and the co-occurrence matrix of a target (model) pencil texture to synthesize a new texture. A predefined pencil width parameter is used when placing the histogram values onto the path. Values are added left to right incrementally from start point to end point.

For most pencil lines, the distribution of grey pixel values starts off with dark value in the middle of the line texture and falls off according to the bell-shaped curve in the direction normal to the line, see Figure 4.2.

I formulate a grey value distribution technique according to this statistical observation. From analyzing scanned lines I used visualization techniques to view the complete data set in 3D view. The Figure 4.3 shows the visualization of the line using *matlab* routines.

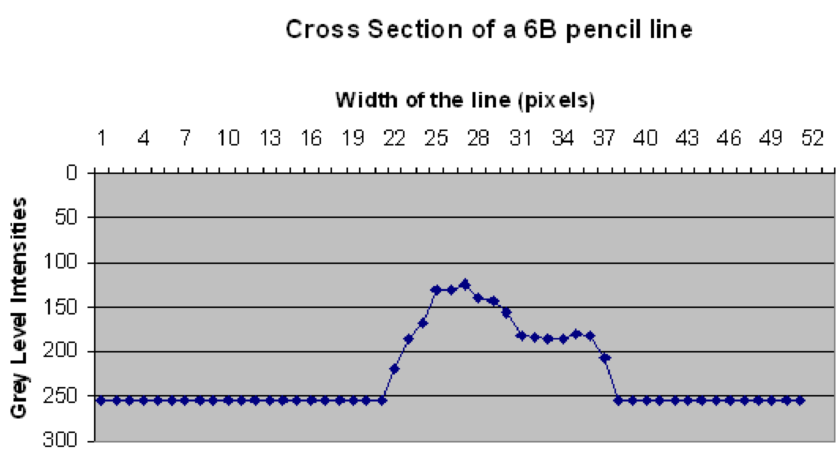


Figure 4.2: *The approximately bell-shaped curve showing intensity values of cross sections of a lines drawn with a 6B pencil.*

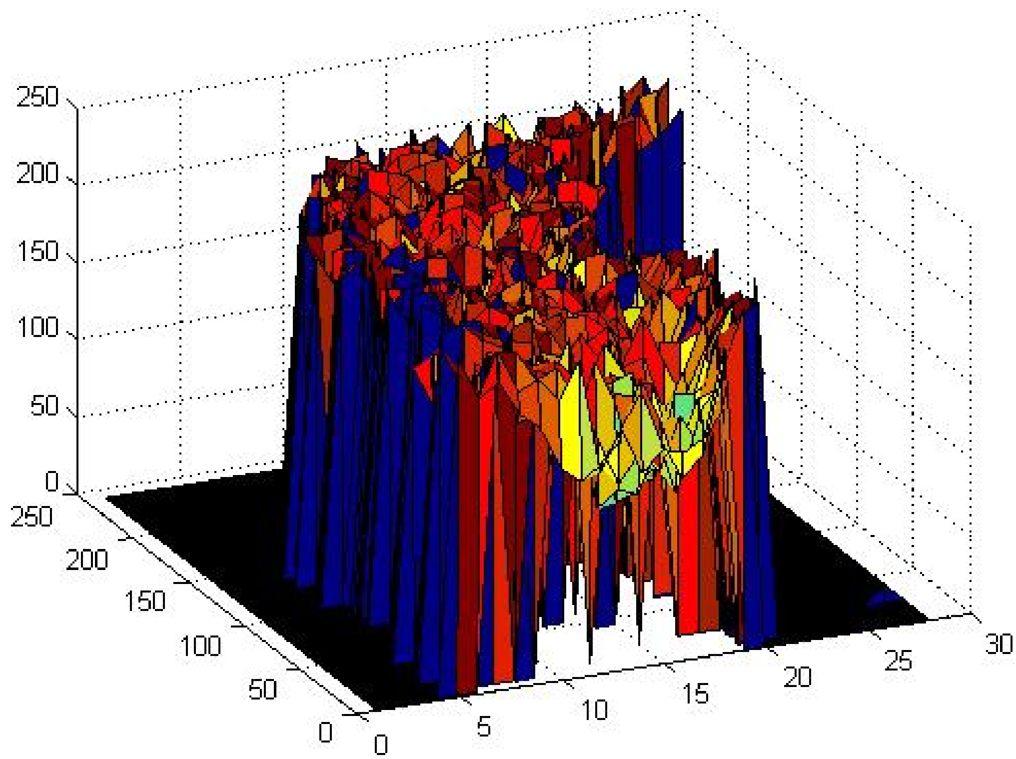


Figure 4.3: *A visualization of the grey level distribution across the width (x axis) and length (z axis) of the line. The height of the map presents the inverse of the grey values (255* - g-value*); you can notice the inverse bell shapes in the color ranges yellow and green.*

The initial line fill includes a random grey value fill of all points along the line and limited to the predefined width. The random function defined uses a flat, uniform distribution. This serves its purpose since the second step of the algorithm applies pixel manipulations to create a texture using the CCM.

First, I seed the line with initial random values (using the histogram of the pre-selected pencil type) and then apply the co-occurrence matrix. According to the bell-shaped fall off observed, I distribute the grey levels of the centre pixels of the generated spline, replicating the histogram in the range $[min_{mid}, max_{mid}]$. The approximate direction normal to the line is determined 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 4.4.

Figure 4.4: *Initial step line texture and partial close-up view of the line texture. Randomly placing the grey dynamic range values across the width and length of the path.*

The co-occurrence matrix is used next to re-organize the initial random values to a more suitable pattern that causes the synthesis of a texture that is similar to the sample texture.

The ultimate goal is to produce a final synthesis with a quality that has more visual similarity to the target texture, and not focus as much on the numerical measure of error between the *GLC*'s of the synthesized and model texture as in [Copeland et al. 2001]. I achieve this through synthesizing a texture from a *CCM*.

Next, the CCM is applied by comparing a 3x3 neighbourhood of pixels. If the combination of pixel intensities does not exist in the CCM, the neighbouring pixel's grey value is replaced with an existing neighboring value from the co-occurrence matrix. An existing value is represented by $C(i, j) == 1$; where i is the grey and index value of the point, and j is the grey and pixel value of the neighbour.

The co-occurrence process can be repeated over the entire line multiple times until the amount of pixel changes reach a minimum, indicating how well the synthesized co-occurrence matrix represents the model CCM (see Figure 4.5).

Because the model textures are taken from lines of set lengths, its is not possible to calculate a quantitative measure of error between the two matrices, the synthesized and the model.

Figure 4.5: *Line and partial close-up view of the line texture after CCM step. 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.*

Once complete, a final step is performed. I apply a single-pass Gaussian filter to the generated lines to reduce tone differences and filter out aliasing effects (Figure 4.6). 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, my texture synthesis model enables the synthesis of perceptually convincing textures of a pencil line as shown in Figure 4.6.

Figure 4.6: *Final pencil line and close-up view of the line after a 3×3 single-pass Gaussian filter is applied.*

The following Table 4.1 shows a summary of the texture generation steps side to side.

I presented in this chapter a simple texture synthesis algorithm that produces visually realistic pencil textures based on the CCM of model textures. The following table, Table 4.2, is a comparison between real texture and generated textures not using a CCM (without step 2 of the synthesis process). A second table is added here for comparison to the final HLA synthesized textures (with CCM), since the difference is not noticed unless the two textures are placed side by side as in Table 4.3. If analyzed closely it can be seen that

Grey filled texture	Zoomed in
	
Co-occurrence application to texture	Zoomed in
	
One pass Gaussian blur	Zoomed in
	

Table 4.1: *This table shows the three steps of the texturing synthesis process.*

the control of texture composition is more evident in the CCM synthesized texture. The texture is more homogenous and big variations between pixel grey value in surrounding neighbourhoods are not noticeable. While the textures not using the CCM step (only using the grey value distribution technique), are less faithful to textures generated by humans; The white and dark pixel arrangements give a sense of disorganization and do not represent the smoothness that can be seen in true graphite pencil drawn lines.

The results chapter, Chapter 5 offers more results from the HLA texturing algorithm with comparisons to real drawn textures and relevant discussions.

Line type	Real human line	"HLA"-generated lines NO CCM
H		
HB		
B		
3B		
6B		
8B		

Table 4.2: *Line samples: comparison of real hand-drawn lines with synthesized lines not using the CCM (deviation parameter set to zero).*

Line type	"HLA"-generated lines with CCM	"HLA"-generated lines NO CCM
H		
HB		
B		
3B		
6B		
8B		

Table 4.3: *Line samples: comparison of HLA synthesized lines (including CCM)and synthesized lines not using the CCM (deviation parameter set to zero).*

Chapter 5

Results and Discussion

The algorithm I present in this thesis is called “The Human Line Algorithm” (HLA). By combining the path and texture algorithms described in Chapter 3 and Chapter 4 the HLA algorithm is capable of creating convincingly acceptable appearance of human-like drawn pencil lines to be verified by an experiment explained in Chapter 6.

The following Table 5.1 shows a comparison of hand-drawn textured lines with lines that are synthesized by my ”HLA” system texturing technique.

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

Table 5.1: *Line texture samples: comparison of hand-drawn textured lines with synthesized textures (deviation parameter (squiggle) set to zero).*

All the lines present in Table 5.1 are created on straight paths; all deviation parameters are removed for the intention of presentation. The purpose of this table is to show similarity between both real and generated line textures without exterior artifacts.

It can be observed that the method used to manufacture the lines is almost undetectable. In Chapter 6, my verification procedure proved very successful and shows how well the *HLA lines* can pass for real lines by viewers.

All the results have been generated on a *2.00GHz* Intel core dual CPU workstation, and implemented using C++ with Qt software *2D* drawing API, without any additional hardware assistance.

Individual lines can be interactively drawn while the system synthesizes the new path and texture. The system has shown to be successful at assisting users (including novice users) in easily producing a variety of images. All line drawing figures in this paper indicated as "generated by HLA" are drawn using the Human Line Algorithm system.

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

(1) Initialize the CCM and histogram of 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.

I have currently applied this algorithm into these four areas:

1. space filling curves,
2. architectural drawings (CAD, Google Sketch up),
3. line patterns generated from the Life Game, and
4. reproductions of artist's drawings.

The line generation is not limited to these fields but can be incorporated easily into existing graphics programs that use line end points, and is editable by artists or additional algorithms. My technique only requires the selection of a pencil type and specification of the end points of each line. For example, Figure 5.1 and Figure 5.2 show two space filling curves, Gosper's Flowsnake model [Gardner 1976] and the dragon curve, rendered with human like lines using the *B* pencil graphite setting. Each short individual line segments of each curve is synthesized using the Human Line Algorithm. The drawings could be improved by detecting co-linear segments and processing them as a longer line to better emulate a human artist.

In Figure 5.3 hatching lines are generated and replaced with synthesized lines; in this way I can simulate the effect of filling a part of the plane with human-like lines as shown.

In Figure 5.4 a hand drawn illustration is used to extract vector lines, then used as input to the "HLA" algorithm resulting in part (c) of the figure. I note that my result is quite different to the artist's lines, but point out that the result in (c) has more of an appearance of being hand-drawn, than the original (a) that seems to have been drawn with the aid of a ruler.

Figure 5.5 shows an example where the output from a CAD application has been replaced by my synthesized lines. A rendering of an object with 36-edges is shown using a simulated H type pencil.

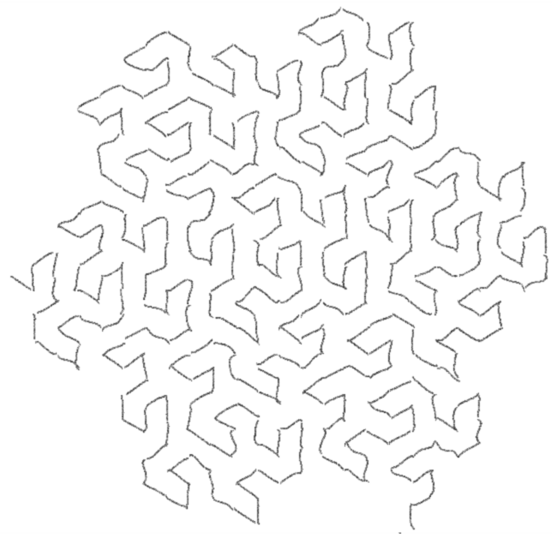


Figure 5.1: *Flowsnake space filling curve using a B pencil generated by HLA.*

In Figure 5.6 an architectural drawing has been rendered by reading in a postscript file input using the HLA 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 (vectorization), the pencil style can be applied to those drawings as well, shown below, Figure 5.7.

5.1 Efficiency

The algorithm has not been optimized to its full capacity. The speed of the algorithm can be improved dramatically if various memory accessing functions and options were considered when coding. Hence optimization is a key to enable faster implementation. Currently each pencil line can be rendered using the HLA in 0.24 of a second, not including the gaussian blur.

- 0.00503947 seconds, to generate the Flash and Hogan trajectory and Catmull-Rom

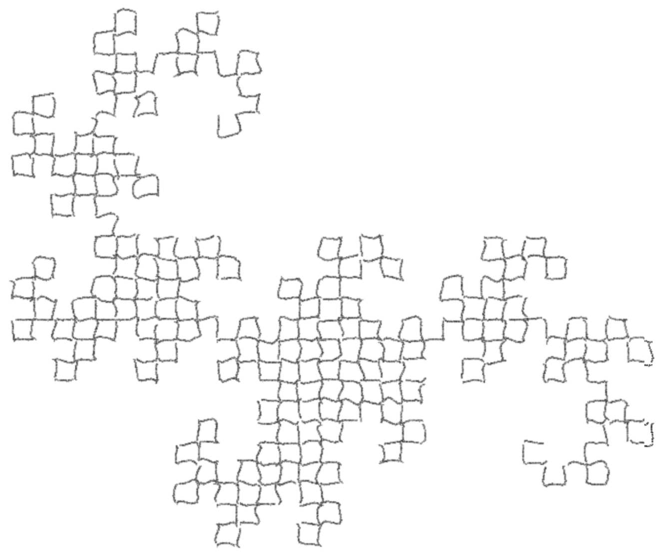


Figure 5.2: *Dragon space filling curve generated by HLA (B pencil).*

spline.

- 0.00686624 seconds, to fill the path with random grey values.
- 0.000768813 seconds, to apply the co-occurrence.
- 5.86667e-006 for various call routines.
- An extra 1.5 second is added to apply a one pass gaussian blur to the image.

As observed in Chapter 6, there is a slight amount of improvement possible to make generated pencil lines more similar to real pencil lines. This mainly lies in the style of termination of the lines at each of its endpoints. More research should be done to mimic the starting and ending of line segments for a more natural appeal.

Printing caused multiple artifacts to the synthesized images. Printer quality images appear different to real pencil lines that have been scanned and printed. Printer resolution did not resolve the issue of the drastic changes in the ranges of grey values which makes the

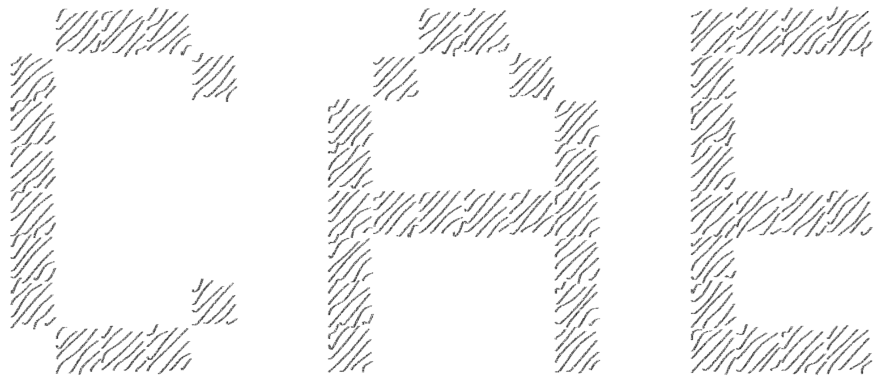
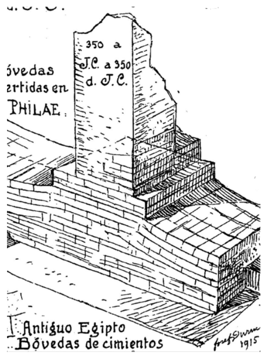
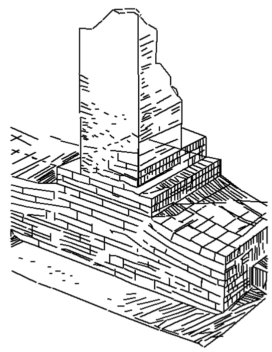


Figure 5.3: *Line hatching generated by HLA (6B pencil). CAE stands for Computational Aesthetics, the name of the conference where this work was published.*

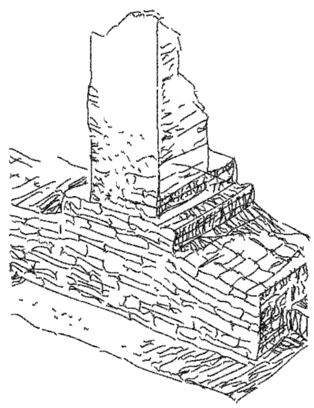
generated lines appear unreal to a certain extent. I recommend printing the images on a 600dpi printer at a minimum resolution of 300dpi, this again does not resolve the problem.



(a) Hand Drawn Illustration.

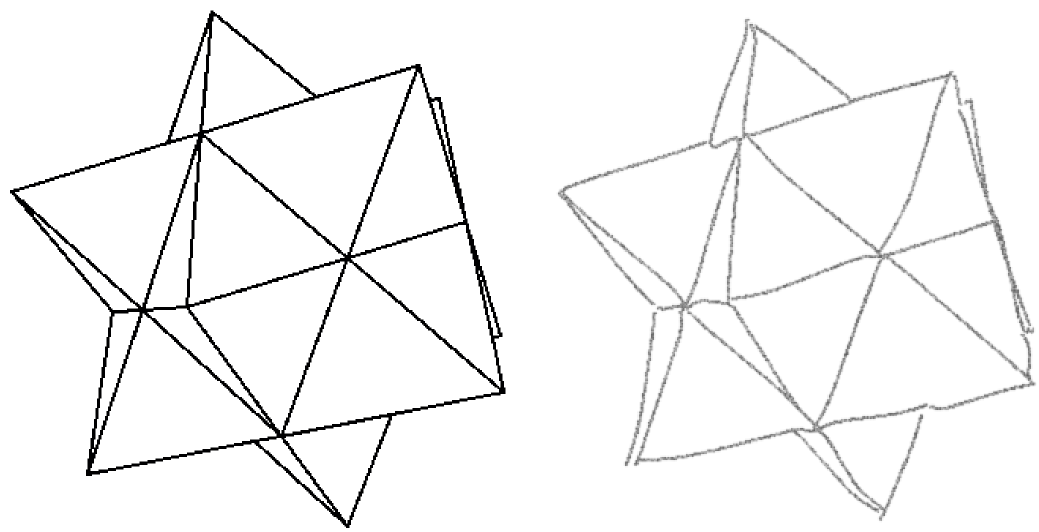


(b) Input Vector Lines.



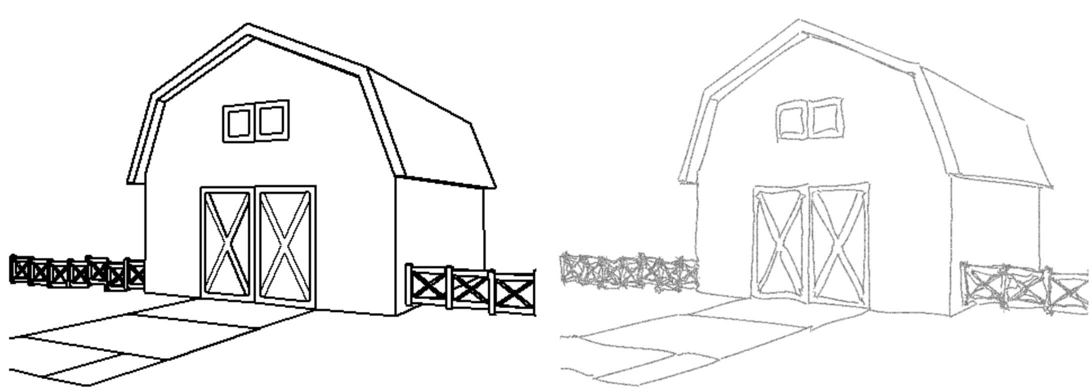
(c) Result generated by HLA pencil type B.

Figure 5.4: *My 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.*



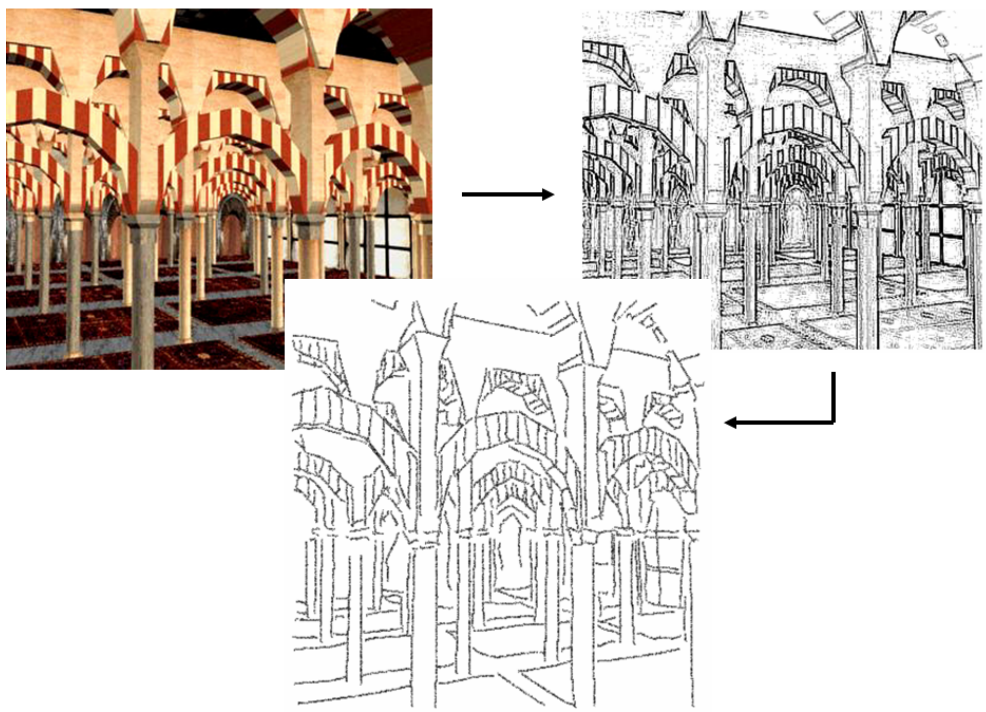
(a) Input Vector Lines. (b) HLA result.

Figure 5.5: *A 36-sided object generated by HLA using pencil type H.*



(a) Input vector Barn. (b) My Result, HLA Barn.

Figure 5.6: *A barn generated by HLA (H pencil).*



(a) Step by step vectorization of Islamic architecture.

Figure 5.7: *The process of vectorizing a photograph, first extract a sketch image from the photograph (use filters) then vectorize in Illustrator. The postscript output of this illustration was rendered using a H type pencil.*

Chapter 6

HLA User Studies

In this chapter I present a user study and verification process for analyzing lines generated from the Human Line Algorithm. The aim is to validate the closeness in nature of both real pencil drawn lines and computer generated lines synthesized from the HLA. I explain the user study in detail and then analyze the data gathered. The analysis and the statistical methods I use are explained.

6.1 HLA Verification

I designed a study to verify the ability of the *HLA* to replace hand-drawn lines. The study involves eighteen pre-rendered images. Nine of these images were scanned human made drawings, drawn using either a *HB*, *6B*, or a *8B* pencil. The other nine were exact replications of the scanned images; the replication images were rendered using the "HLA" line algorithm using the same mentioned pencil types. The images were at 800×800 pixels in scale. The resolution of the scanned in hand made images were at *300 dpi*.

The aim of this study is to evaluate whether my generated lines could pass for real human drawn lines.

I conducted the study on eleven subjects, all graduate and undergraduate students in the department of Computer Science at the University of Victoria.

Each participant spent in total about 2 minutes performing the study in front of a desktop computer. They were given a viewing time of three second for each image and then were asked to select *true* if they thought the drawing was hand-made, or *false* if they thought the drawing was computer generated. The timing is chosen empirically, such that participants have enough time to compare drawings without having too much time to closely analyze them; all the images were presented in random order.

The following tables, Table 6.1, Table 6.2, Table 6.3, show the images presented to the participants in the experiment using the above described scheme.

To interpret the results, with the help of Petra Isenberg, I ran the collected data through a paired sample T-test using SPSS statistical software. The following tables, Table 6.4, and Table 6.5 give the statistics used, the interpretations made, and the results carried out by the sample T-test.

Out of the total amount of choices made, 42% were incorrect. Out of the incorrect choices made by all subject decisions, 69.5% select images to be of hand drawn type (when they are actually computer-generated) and 30.5% select images to be of computer-generated type (when they are actually real).

The sample paired T-test shows a significant difference between computer–generated lines mistaken for real hand-drawn lines and the real hand drawn lines mistaken for computer-generated lines (paired $t(11) = 2.849$; $p < 0.05$). The experimental results (p-value) actually is $p = 0.017$; It shows that the HLA generated line drawings are good enough to pass for hand-drawn lines.

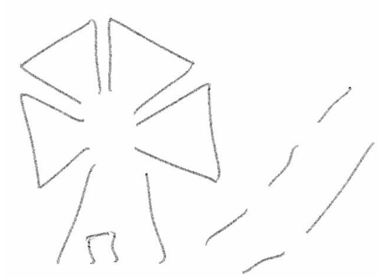
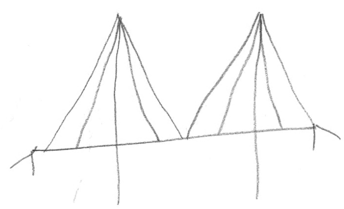
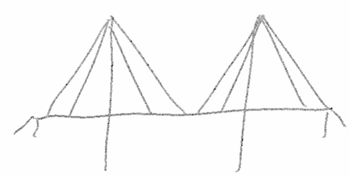
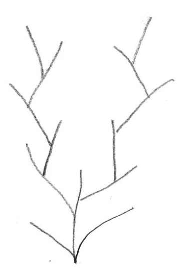
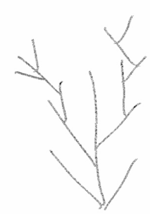
Pencil Type	Real Human Made Image	"HLA" Generated Image
HB		
HB		
HB		

Table 6.1: *Verification Image Samples (1): comparison of hand-drawn images with synthesized images used in the verification experiment. Each of these images were done using a HB pencil type.*

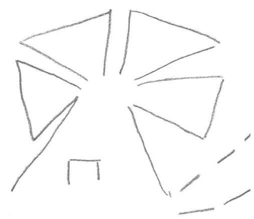
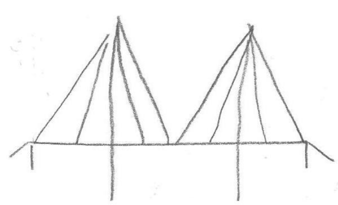
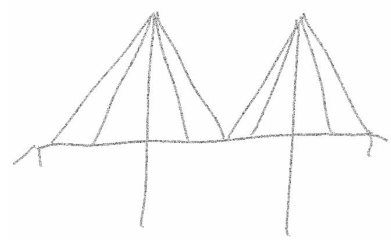
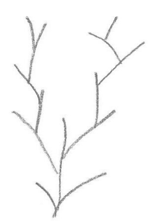
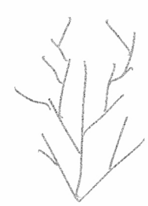
Pencil Type	Real Human Made Image	"HLA" Generated Image
6B		
6B		
6B		

Table 6.2: *Verification Image Samples (2): comparison of hand-drawn images with synthesized images used in the verification experiment. Each of these images were done using a* 6*B pencil type.*

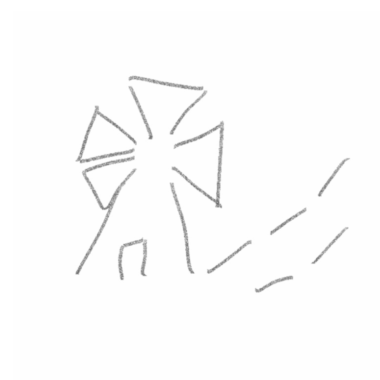
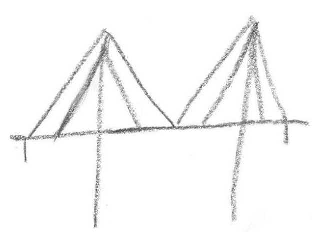
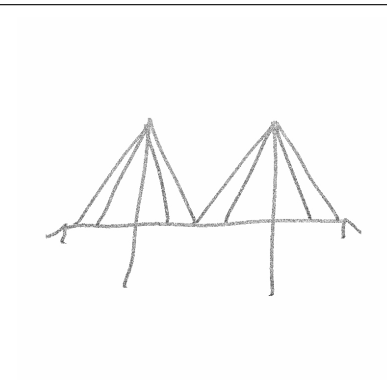
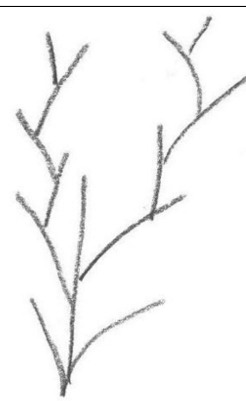
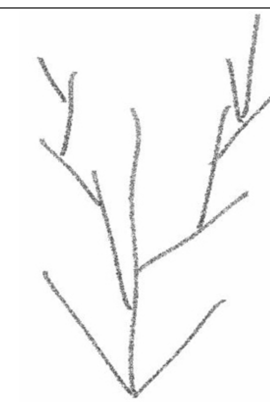
Pencil Type	Real Human Made Image	“HLA” Generated Image
8B		
8B		
8B		

Table 6.3: *Verification Image Samples (3): comparison of hand-drawn images with synthesized images used in the verification experiment. Each of these images were done using a* 8*B pencil type.*

Participant number	Generated Images	Real Hand-drawn Image
P1	7	4
P2	0	0
P3	3	4
P4	4	2
P5	4	2
P6	8	3
P7	2	4
P8	8	3
P9	5	3
P10	9	0
P11	9	1
Sum Errors	59	26
% of all errors	0.694117647	0.305882353

Table 6.4: *This table shows the results of the selection of the eleven participants who performed the study. The values of the real and generated images show the number of incorrect choices made by the participants. The percentages were used to acquire the p-value of the data.*

	Variable 1	Variable 2
Mean	5.363636364	2.363636364
Variance	9.254545455	2.254545455
Observations	11	11
Pearson Correlation	-0.075628218	
Hypothesized Mean Difference	0	
df	10	
t Stat	2.848640746	
P(T $<=$ t) one-tail	0.008645167	
t Critical one-tail	1.812461102	
P(T $<=$ t) two-tail	0.017290335	
t Critical two-tail	2.228138842	

Table 6.5: *The results of the paired sample T-test.*

Chapter 7

Conclusions and Future Work

A system has been presented that will serve as a high quality pencil line reproduction agent, creating aesthetically pleasing human drawn pencil lines using a texture synthesis method and a human arm movement replication model. The major advantage of the algorithm is that it avoids computationally expensive techniques and large storage space.

The mathematical model for human arm trajectory formation that I have adopted gives a simple and efficient way to calculate a realistic path preserving the look of a typical human movement. The algorithm offers the user multiple options for line lengths, short, medium, and long. In addition to that it offers various options for widths of the lines, and a degree of inaccuracy that can be applied onto the trajectory path.

The texture synthesis method I apply provides a fast and simple way to produce convincing pencil textures that have the same qualities and similar statistical values taken from predefined sample textures. This in turn makes it possible to continuously produce new unique textures that are visually indistinguishable from real drawn pencil lines.

The principal advantages of my method are its ability to re-create unique instances of aesthetically pleasing pencil lines, and its capability to be incorporated into raster applications. Using the Flash and Hogan [Flash and Hogan 1985], the human mathematical arm trajectory model gives a guaranteed unique trajectories and is done in linear time. Further, the texture synthesis technique I use allows for further optimization to improve the performance of pencil line generation.

I found that this approach provides convincing results in comparison to previous work on pencil line generation. Using a cubic Catmull-Rom spline to form a smooth base line for the path, then the application of the texture on top of it proved to help this fact.

The system is not intended to replace artists or illustration, but to provide a tool for users with no training to produce traditional images using a pencil medium.

There are a few limitations of the system: The speed, and the number of styles produced. I would like to extend the system to include different types of styles, as in [Kalnins et al. 2002]. The current system only supports one type of drawing style that uses my texture synthesis technique. This limitation is not critical for the proposed work, but by including other styles like the various parameters defined by Sousa and Buchanan [1999b] I can increase the sense of more expressive artistic features that can be found in real hand drawn images. This can be done by editing the co-occurrence matrices before synthesizing the texture. I can also with more observations and work mimic line starting and ending points (were the pencil is placed and lifted when drawing a line) using the co-occurrence manipulation technique.

Many characteristics of basic drawn straight lines observed from the initial user study Section 3.1 can be summarized into a few points based solely on observations, these characteristics are:

- paths end up higher and lighter than the ideal path between two points for a line drawn from left to right.
- paths end up lower and darker than the ideal path for lines drawn from right to left.
- paths tended to deviate less for lines drawn towards the participant than lines drawn away from them.
- lines alternate above and below the ideal path for either case.
- line could start off accurate and dark and then deviate after the midway point on the

path.

These observations can form a basis for future co-occurrence matrix manipulation for future style generation. By manipulating the co-occurrence matrix I could also make the system include the ability to capture line qualities as in Sousa and Buchanan [2000], such as pencil angle, tip, and pressure would enhance the illustration system by allowing more expressive renderings. This would be more appropriate if used with curved lines than straight line drawings, since more expression is found in non-straight pencil lines as per my observations in the conducted studies (Chapter 6).

Another area I would like to extend the system to include is drawing curved lines, as in example Figure 7.1, and develop expressive ways to present them. In Figure 7.1 I used the mathematical model by [Flash and Hogan 1985] for curved lines to produce a human-like expressive look to the image. The curved lines seen here, were textured using my straight line texturing technique; it is necessary to conduct further texture synthesis investigation 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 5.3 is my first attempt to consider hatching as an application, more experiments and analysis are needed to present accurate human hatching techniques.

The system can also be extended to support direct (or interactive) edge extraction from a read in image. The system can then read the start and end points of each segment and apply the rendering, eliminating the need of reading in postscript outputs.

Figure 7.1: *Example curve generated by HLA, using Flash and Hogan curve optimization methods, inspired by [Finkelstein and Salesin 1994].*

Bibliography

Bibliography

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

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

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

[Contreras-Vidal et al. 1997] 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 et al. 2001] 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.

[Curtis et al. 1997] CURTIS, C. J., ANDERSON, S. E., SEIMS, J. E., FLEISCHER, K. W., AND SALESIN, D. H. 1997. Computer generated watercolor. In *Proc. SIGGRAPH*, ACM Press, New York, 421–430.

[Finkelstein and Salesin 1994] FINKELSTEIN, A., AND SALESIN, D. H. 1994. Multiresolution Curves. In *Proc. SIGGRAPH*, ACM Press, New York, 261–268.

[Flash and Hogan 1985] 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 et al. 2003] 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 and Ma 1985] 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] GARDNER. 1976. Mathematical games: In which “monster” curves force redefinition of the word “curve”. *SCIAM: Scientific American 235*, 124–133.

[Gooch and Gooch 2001] GOOCH, B., AND GOOCH, A. A. 2001. *Non-Photorealistic Rendering*. A K Peters, Ltd., Natick.

[Hertzmann et al. 2002] HERTZMANN, A., OLIVER, N., CURLES, B., AND SEITZ, S. M. 2002. Curve Analogies. In *Rendering Techniques*, EUROGRAPHICS Association, Aire-la-Ville, Switzerland.

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

[Isenberg et al. 2006] ISENBERG, T., NEUMANN, P., CARPENDALE, S., SOUSA, M. C., AND JORGE, J. A. 2006. Non-Photorealistic Rendering in Context: An Observational Study. In *Proc. NPAR*, ACM Press, New York, 115–126.

[Jodoin et al. 2002] 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 et al. 2002] 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.

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

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

[Mazzoni et al. 1991] 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.

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

[Salisbury et al. 1994] 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.

[Samavati and Bartels 1999] SAMAVATI, F. F., AND BARTELS, R. 1999. Multiresolution Curve and Surface Representation. *Computer Graphics Forum 18*, 2 (June), 97–119.

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

[Schumann et al. 1996] SCHUMANN, J., STROTHOTTE, T., RAAB, A., AND LASER, S. 1996. Assessing the Effect of Non-photorealistic Rendered Images in CAD. In *Proc. CHI*, ACM Press, New York, 35–42.

[Sechrest and Greenberg 1981] SECHREST, S., AND GREENBERG, D. P. 1981. A visible polygon reconstruction algorithm. *SIGGRAPH Comput. Graph. 15*, 3, 17–27.

[Shirley et al. 2005] 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.

[Simhon and Dudek 2004] SIMHON, S., AND DUDEK, G. 2004. Sketch Interpretation and Refinement Using Statistical Models. In *Rendering Techniques*, EUROGRAPHICS Association, Aire-la-Ville, Switzerland, 23–32.

[Sousa and Buchanan 1999a] 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 and Buchanan 1999b] SOUSA, M. C., AND BUCHANAN, J. W. 1999. Computer-Generated Pencil Drawing. In *Proceedings of SKIGRAPH 1999, Tenth Western Computer Graphics Symposium (Banff, Canada, March 1999)*.

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

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

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

[Vermeulen and Tanner 1989] VERMEULEN, A. H., AND TANNER, P. P. 1989. PencilSketch—A Pencil-Based Paint System. In *Proc. Graphics Interface*, Morgan Kaufann, San Francisco, 138–143.

[Winkenbach and Salesin 1994] WINKENBACH, G., AND SALESIN, D. H. 1994. Computer-generated pen-and-ink illustration. *Computer Graphics 28*, Annual Conference Series, 91–100.

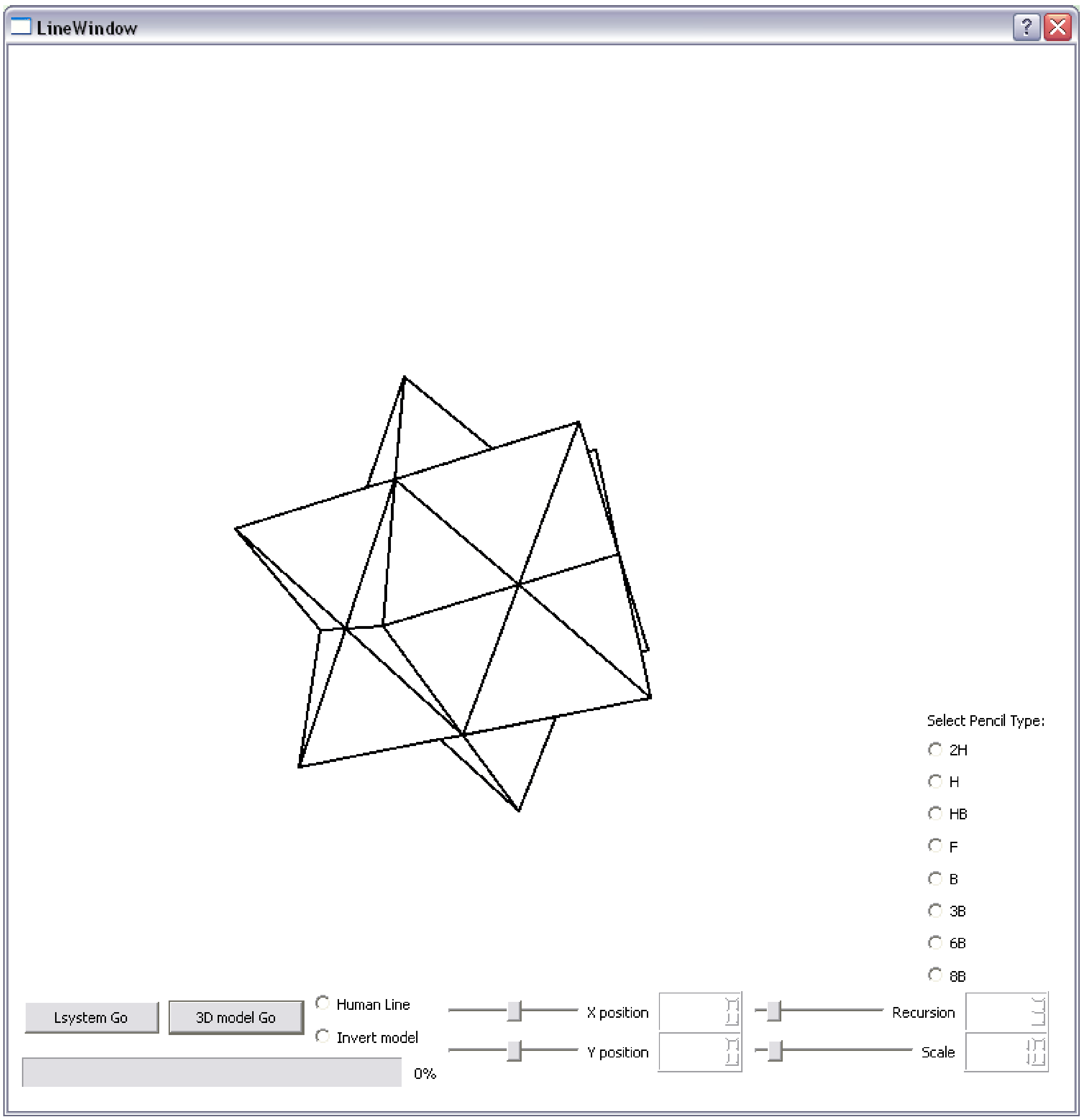
[Woodworth 1899] WOODWORTH, R. 1899. The Accuracy of Voluntary Movement. *Psychological Review: Monograph Supplements 3*, 1–114.

[Zalesny and Gool 2001] 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.

[Zelaznik 1996] ZELAZNIK, H. N. 1996. *Advances in Motor Learning and Control*, 1 ed. Human Kinetics, Champain, IL.

The System Interface

System Interface


LineWindow
Select Pencil Type:
2H
H
HB
F
B
3B
6B
8B
Lsystem Go
3D model Go
Human Line
Invert model
0%
X position
Y position
Recursion
Scale

Appendix A

Ethics Approval

The following Appendix presents the ethics documentation process in the following structure:

1. Human Research Ethics Board Certificate of Approval.
2. Letter of Information for Implied Consent.
3. Participant Consent Form.
4. Verbal Script Consent Statement.
5. Questionnaire.

Human Research Ethics Board
Office of Research Services
University of Victoria
Technology Enterprise Facility, Room 218
Tel (250) 472-4545 Fax (250) 721-7836
Email ethics@uvic.ca Web www.research.uvic.ca

University of Victoria

Human Research Ethics Board
Certificate of Approval

Principal Investigator	Department/School	Supervisor
Zainab Meraj **Master's Student**	**COSI**	**Brian Wyvill**

Co-Investigator(s):

Tobias Isenberg, Co-investigator, University of Calgary
Amy Gooch, Co-investigator, University of Victoria
Bruce Gooch, Co-investigator, University of Victoria

Project Title: **Human Line Drawing**

Protocol No.	Approval Date	Start Date	End Date
07-200	**26-Jul-07**	**26-Jul-07**	**25-Jul-10**

Certification

This certifies that the UVic Human Research Ethics Board has examined this research protocol and concludes that, in all respects, the proposed research meets appropriate standards of ethics as outlined by the University of Victoria Research Regulations Involving Human Subjects.

Dr. Richard Keeler
Associate Vice-President, Research

This Certificate of Approval is valid for the above term provided there is no change in the procedures. Extensions or minor amendments may be granted upon receipt of a "Research Status" form.

07-200 Meraj, Zainab

Department of Computer Science
University of Victoria

Letter of Information for Implied Consent

[Emulating Human Drawn Pencil Lines]

You are invited to participate in a study entitled **[Emulating Human Drawn Pencil Lines]** that is being conducted by Zainab Meraj.

Zainab Meraj is a graduate student in the department of [Computer Science at the University of Victoria and you may contact her if you have further questions by email: zmeraj@csc.uvic.ca

As a graduate student, I am required to conduct research as part of the requirements for a degree in Masters of Computer Science. It is being conducted under the supervision of Professor Brian Wyvill. You may contact my supervisor at (250) 472-5760.

Purpose and Objectives
The purpose of this research project is as follows:
"We want a line (path) drawing algorithm to emulate human drawn straight lines"

Importance of this Research
Research of this type is important because Non-Photo realistic Rendering (NPR) has used up to now random methods to introduce variance in lines to make them look real, and natural, while this study will base these values on statistics and produce relatively accurate lines, indistinguishable than that of humans.

Participants Selection
You are being asked to participate in this study because you are within the search area, (building of computer science and engineering), and that you are to be of a group of either male of female.

What is involved
If you agree to voluntarily participate in this research, your participation will drawing on several pieces of paper straight and curved lines of various lengths while the whole process is video recorded.

Inconvenience
Participation in this study may cause some inconvenience to you, including the video recording, but we assure no identity will be revealed.

Risks
There are no known or anticipated risks to you by participating in this research

Benefits
The potential benefits of your participation in this research include the advancement of science in both computer graphics, non-photorealistic rendering, and visualization. And in whole to the computer science field.

Voluntary Participation
Your participation in this research must be completely voluntary. If you do decide to participate, you may withdraw at any time without any consequences or any explanation. If you do withdraw from the study your data will only be used if you decide to allow us to do so. <u>The time the study will take around 20 minutes.</u>

Anonymity

In terms of protecting your anonymity if faces are recorded. All documents are classified as confidential and no one other than the prime researcher has access to the recordings.

Confidentiality
Your confidentiality and the confidentiality of the data will be protected by: The only people with access to the data are the investigator, the advisor and two university professors included as observers. The study material will only be stored in one office, and no one is to remove them from that office.

Dissemination of Results
It is anticipated that the results of this study will be shared with others in the following ways:
Statistics gathered from the study will be plotted and evaluated and eventually used as part of an algorithm. No actual papers drawn by the volunteers will be physically used or presented.

Disposal of Data
Data from this study will be disposed of eventually by being shredded and if kept indefinitely, will be stored as confidential records in the department.
.

Contacts
Individuals that may be contacted regarding this study include the investigator Zainab Meraj, and Advisor Mr. Brian Wyvill.

In addition, you may verify the ethical approval of this study, or raise any concerns you might have, by contacting the Human Research Ethics Office at the University of Victoria (250-472-4545 or ethics@uvic.ca).

By completing and submitting the questionnaire, **YOUR FREE AND INFORMED CONSENT IS IMPLIED** and indicates that you understand the above conditions of participation in this study and that you have had the opportunity to have your questions answered by the researchers.

Please retain a copy of this letter for your reference.

Department of Computer Science
University of Victoria

Participant Consent Form

[Emulating Human Drawn Pencil Lines]

You are invited to participate in a study entitled **[Emulating Human Drawn Pencil Lines]** that is being conducted by Zainab Meraj.

Zainab Meraj is a graduate student in the department of [Computer Science at the University of Victoria and you may contact her if you have further questions by email: zmeraj@csc.uvic.ca

As a graduate student, I am required to conduct research as part of the requirements for a degree in Masters of Computer Science. It is being conducted under the supervision of Professor Brian Wyvill. You may contact my supervisor at (250) 472-5760.

Purpose and Objectives
The purpose of this research project is as follows:
"We want a line (path) drawing algorithm to emulate human drawn straight lines"

Importance of this Research
Research of this type is important because Non-Photo realistic Rendering (NPR) has used up to now random methods to introduce variance in lines to make them look real, and natural, while this study will base these values on statistics and produce relatively accurate lines, indistinguishable than that of humans.

Participants Selection
You are being asked to participate in this study because you are within the search area, (building of computer science and engineering), and that you are to be of a group of either male of female.

What is involved
If you agree to voluntarily participate in this research, your participation will drawing on several pieces of paper straight and curved lines of various lengths while the whole process is video recorded.

I agree to be video-taped, and audio-taped. ☐ Yes ☐ No
Initials:______

I agree to let my conversation during the study be directly quoted anonymously, in presentation of the research results.
☐ Yes ☐ No
Initials:______

I agree to let video-tapings and audio-taping be used for presentation of the research results

Yes No
Initials:______

Inconvenience
Participation in this study may cause some inconvenience to you, including the video recording, but we assure no identity will be revealed.

Risks
There are no known or anticipated risks to you by participating in this research

Benefits
The potential benefits of your participation in this research include the advancement of science in both computer graphics, non-photorealistic rendering, and visualization. And in whole to the computer science field.

Voluntary Participation
Your participation in this research must be completely voluntary. If you do decide to participate, you may withdraw at any time without any consequences or any explanation. If you do withdraw from the study your data will only be used if you decide to allow us to do so. The time the study will take around 20 minutes.

Anonymity
In terms of protecting your anonymity if faces are recorded. All documents are classified as confidential and no one other than the prime researcher has access to the recordings.

Confidentiality
Your confidentiality and the confidentiality of the data will be protected by: The only people with access to the data are the investigator, the advisor and two university professors included as observers. The study material will only be stored in one office, and no one is to remove them from that office.

Dissemination of Results
It is anticipated that the results of this study will be shared with others in the following ways:
Statistics gathered from the study will be plotted and evaluated and eventually used as part of an algorithm. No actual papers drawn by the volunteers will be physically used. If any papers are used, scans will be published in anonymous form.

Disposal of Data
Data from this study will be disposed of eventually by being shredded, and if kept indefinitely, will be stored as confidential records in the department.

Contacts
Individuals that may be contacted regarding this study include:
- The investigator Zainab Meraj, email zmeraj@csc.uvic.ca, phone (250)472-5868.
- The Advisor Mr. Brian Wyvill, email blob@cs.uvic.ca, phone (250)472-5760
- Co-Inverstigators:
    - Amy Gooch, email agooch@cs.uvic.ca, phone (250)472-5784
    - Tobias Isenberg, email Isenberg@cs.rug.nl, phone +31-50-3633800

In addition, you may verify the ethical approval of this study, or raise any concerns you might have, by contacting the Human Research Ethics Office at the University of Victoria (250-472-4545 or ethics@uvic.ca).

Your signature below indicates that you understand the above conditions of participation in this study and that you have had the opportunity to have your questions answered by the researchers.

Name of Participant	*Signature*	*Date*

A copy of this consent will be left with you, and a copy will be taken by the researcher.

Verbal Consent:

Would you like to volunteer to complete an experiment for the purpose of observing how people draw?

All information gathered is confidential. If you wish to pull out at any time you have the choice to allow us to use the information or disregard it.

It will take in total about 20 minutes of your time.

We will be video taping the session and hope that it will not be an inconvenience to you. If you don't wish to, then its your choice to change your mind and not participate.

Please read the letter of consent and if you wish to volunteer then you can sign the form.

Thank you.

User Study Participant Questionnaire:

1. **Participant number:** __________
2. **Age group** ☐ **20-25** ☐ **26-30** ☐ **31-40**
3. **Gender:** ☐ **Male** ☐ **Female**
4. **What was the first language you learnt to write?**

   ________________________ ☐ **Left to Right** ☐ **Right to Left**

   Country: ____________________________

   Your ethnic background: _________________________

   Comments: ____

5. **Are you** ☐ **Left handed.** ☐ **Right handed.**
6. **Do you have corrected vision?** ☐ **Yes** ☐ **No**

   Comment: ____

7. **Do you experience any weaknesses in your selected writing hand or arm?**

   ☐ **Yes** ☐ **No**

   If yes, what kind: ____

8. **Do you practice drawing or sketching often?** ☐ **Yes** ☐ **No**

   Comment: ____

9. **How would you describe your level as an artist:**

   ☐ **Excellent**

   ☐ **Advanced**

   ☐ **Amateur**

   ☐ **Know basics**

   ☐ **I don't know much**

Thank you for participating in this study. Have a great day.

Appendix B

Initial User Study

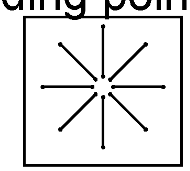
The following study was designed and conducted by the author. The main purpose was to collect data for observational puposes. The concluded observations were presented in this thesis.

Participant # 1

Line Study:

The Objective (1):

- To draw straight lines between dots.
- Don't use a dot more than once.
- Example format of sketches expected from you are shown next to task (like below).
- Starting points are circles ○ , ending points are squares □



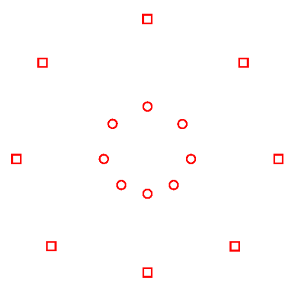
<u>The Rules</u>

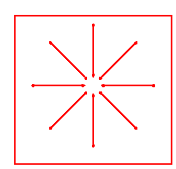
– Sit up straight.
– Hold pencil **steadily.**
– Draw the lines with in a comfortable pace.
– Leave paper in the **same** position **always**.

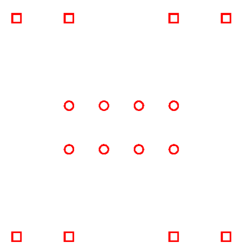
– Don’t draw over a line a second time
– Do not erase anything.

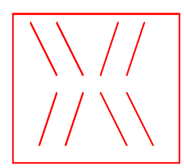
○ Start Point

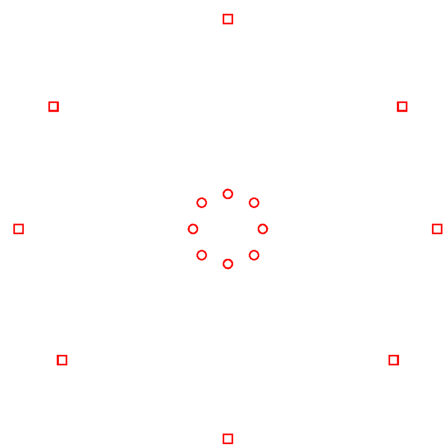
□ End Point

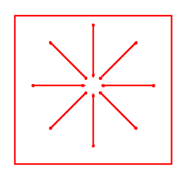


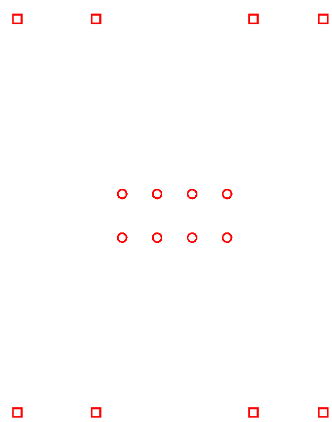


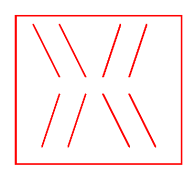


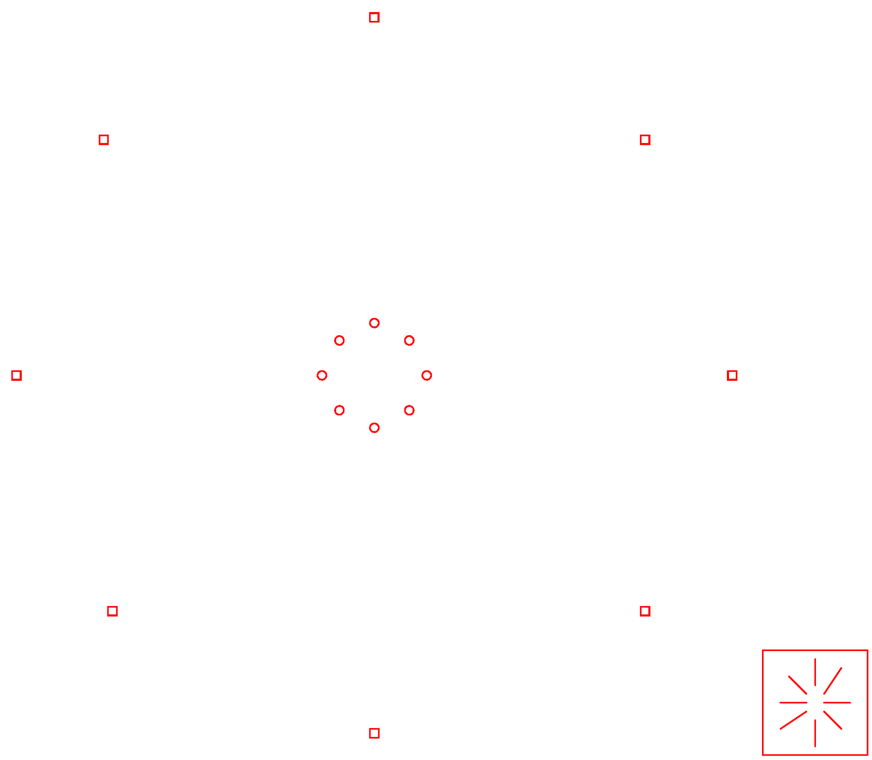


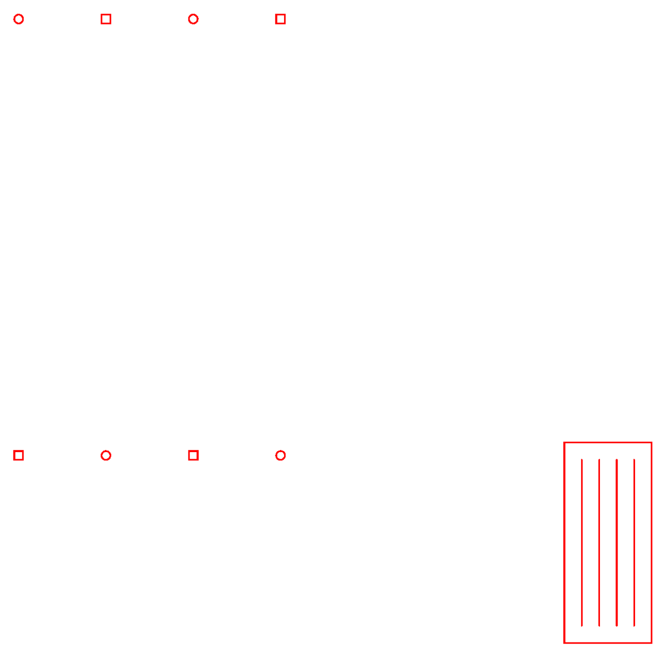


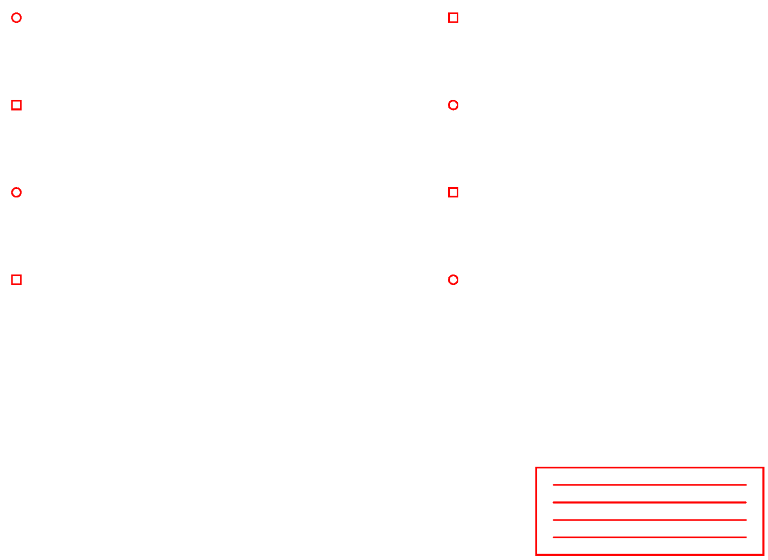








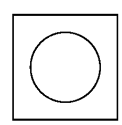


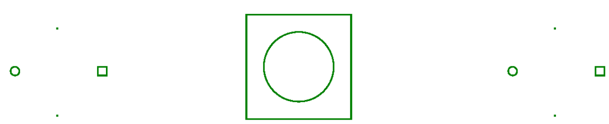


Curve Study:

The Objective (2):

- To draw curved lines between dots.
- Don’t use a dot more than once.
- Example format of sketches expected from you are shown next to task (like below).
- Starting points are circles ○ , ending points are squares □





Bottom of page