SIMULATING ARTISTIC BRUSHSTROKES USING INTERVAL SPLINES

Sara L. Su [1, 2]
sarasu@cs.washington.edu

Ying-Qing Xu [1]
yqxu@microsoft.com

Heung-Yeung Shum [1]
hshum@microsoft.com

Falai Chen [3]
chenfl@ustc.edu.cn

[1] Visual Computing Group
Microsoft Research Asia
Microsoft Corporation
Beijing, China

[2] Department of Computer
Science and Engineering
University of Washington
Seattle, WA, USA

[3] Department of Mathematics
University of Science and
Technology of China
Hefei, China

ABSTRACT

In this paper we present a novel method of simulating the elegant brushstrokes found in calligraphic lettering and painting. Such simulations have earlier been attempted with physically based or texture-mapped approaches, methods that succeeded in producing aesthetic, but also computationally intensive, results. We introduce a brushstroke model based on a parametric curve, the interval spline. By defining brush effects as mathematical constraints between knots, we generate artistic strokes that render faster than earlier methods and are also resistant to scaling. More significantly, because the construction of the interval spline curve is based on changes in the shape of the brush as it moves along a path, variations within a single stroke not fully modeled with earlier attempts are inherently captured by this representation. We present a system for painting with interval spline strokes and discuss a number of examples created with this method.

KEYWORDS

Non-photorealistic rendering, Stroke generation, Interval spline, Digital painting.

1 INTRODUCTION

Much NPR research effort has been devoted to simulation of traditional graphic media. We focus our attention on the elegant brushstrokes of calligraphic (or brush-pen) painting.

Classical calligraphy has long been one of the most respected of brush arts. In addition to describing a particular style of painting, calligraphy refers to the artistic form of lettering practiced by Asian and European scholars for centuries. The long, fluid strokes on a Chinese scroll and the even, controlled lettering on an illuminated manuscript are the results of years of disciplined practice fueled by artistic instinct. Though a computer cannot capture the creative spirit of an artist, it can simulate the brushstroke characteristics that are so important in its manifestation. The 8th-century Chinese painter and poet Wang Wei describes the importance of stroke shape in a painting:

> *"With a curved line I represent the Song mountain ranges… A swift stroke will be sufficient for the Taihua Mountain… With changes and variations in all directions, movement is created…"* [6]

In this paper we present our efforts to simulate some of the features that make calligraphic brushstrokes so compelling.

1.1 Related work

Our simulation of brushstrokes draws from a long line of earlier work in understanding the interactions of bristles and ink with paper. The various models of brushstrokes attempted in recent years range from texture-mapping, to application of arbitrarily deformable images along a stroke path, to physical simulations. There has also been a great deal of work in user interfaces for allowing an artist to input strokes into the computer. Here we briefly discuss the efforts that have laid the ground work for our research.

Texture-mapped methods. In rendering artistic silhouettes, Northrup and Markosian created the basis of an artistic stroke, the variable-width line primitive, by defining "rib vectors" composed of triangle strips [16]. Hsu et al. introduced "skeletal strokes", deformable images that can be anchored, scaled, or transformed by multiple factors at each control point [11]. Hsu and Lee went on to explore applications of such strokes in a drawing and animation systems [10]. While this push toward mathematically deformable models of brushstrokes has led to some promising results, these representations still fundamentally rely on texture mapping for rendering many of the artistic stroke effects.

Physically-based approaches. Though their watercolor simulator is empirically based, Curtis et al. make use a cellular automaton and other physical models to simulate the dynamics of water interacting with suspended pigment [5]. Their simulation drew from the work of Strassman, who produced "hairy brushes" simulating the interactions of bristles, ink, water, and paper [20]. Haeberli's Dynadraw connects a virtual mass to the mouse with a damped spring. As the user draws, the movement of the mass, not the mouse, is stroked [8]. The brushstrokes resulting from this early work were visually stunning but, due to the complexity of the physical simulation, also computationally intensive.

Stroke input interfaces. There has been nearly as much work on methods for inputting strokes as there has on rendering them. Bleser et al. gave the user a pressure- and angle-sensitive tablet on which to create "charcoal" sketches [2]. Chua and Winton devised a low-cost mouse input device allowing the user to change brush widths and angles as a stroke is constructed [4].

1.2 Overview

In the following section, we discuss the specific properties of an artist's brushstroke that we seek to model, and in Section 3 we outline our approach to simulating such effects. Section 4 introduces the mathematical basis of our stroke model, the interval spline and interval spline curve. Section 5 discusses artistic stroke effects simulated using the interval spline model, and in Section 6, we present images generated by our artistic stroke system. Section 7 discusses our results and future work.

2 STROKE PROPERTIES

The style of Chinese painters ranges from the strong, explosive marks of the "ink-splash" method and to the informal, loose strokes of the "worn-out brush" manner. We focus on the brush-pen technique which incorporates both thin, continuous lines and broad, soft strokes. Large patches of "ink-splash" shading are also often used.

The shape of a stroke on paper depends a number of factors. These include the speed of painting, the texture of paper or canvas, the pressure applied at particular points, and the brush angle, and most of all, the artist. However, we identify a number of features shared across many styles: The initial stroke is typically the heaviest, and the amount of pigment tapers off as the stroke is completed. Ink is distributed more sparsely at stroke ends and edges and more densely at thick corners. In addition to the variations in transparency, we seek to model the veins of white along the length of the stroke where there is absence of ink.

3 SIMULATING ARTISTIC STROKES

Various methods have been attempted in recent years with the goal of simulating such strokes. Although the results of physical simulations and texture-mappings are attractive, there are a number of inherent problems that limit their practical usage in an interactive system.

The first is inefficiency. The texture-mapping process is computationally intensive as texture coordinates along the length of the stroke must be re-calculated each time the stroke is re-drawn. A physically accurate simulation is also necessarily time-intensive. In fluid simulations, for example, systems of Navier-Stokes equations must often be solved.

Another issue is physical scalability. It is a challenge to ensure the appearance of the texture-mapped stroke at all sizes and display resolutions.

Finally, because our goal is to allow a stroke's construction to proceed in a manner intuitive for the user, we seek a model extensible for use with a pen-based input device. There is currently no easy way to extend a texture-mapped stroke for such use. The shape of a stroke on paper depends on the speed at which it is drawn, the pressure applied at particular points, and other factors. With texture-mapped methods, the only way to achieve such variations *within a single stroke* is to apply multiple textures along segments of the stroke.

These three issues are addressed by our use of the interval spline, a parametric curve that stores dense vectors of information at its knots.

By defining artistic styles as mathematical constraints between curve points, we are able to generate brushstrokes that both resist scaling and render faster than texture-mapped lines. A basic variable-width line is easily rendered, and stylistic variations can be applied along the length of the stroke. Such artistic effects can be described quantitatively in terms of constraints between parameters stored at its knots. This vector-based approach simplifies the representation of a complex stroke, reducing data stored in addition to the rendering time, making them ideal for use in an interactive system.

The mathematical basis of an interval spline curve provides a solution to the third problem. The construction of the interval spline curve, discussed in detail in the next section, can be visualized as being based on control shapes (knots) defined though time as a brush is moved along its path. These snapshots of the brush's shape form the knots of the interval spline curve. Thus, brush variations within a single stroke are inherently captured by the interval spline model.

4 INTERVAL SPLINES AND CURVES

The interval spline curve is a parametric representation capable of a complete description of coefficient errors. Interval analysis first emerged as a tool in numerical mathematics to enable computers to execute algorithms capable of capturing round off errors automatically [15]. Related curves have in the past been primarily applied in the domain of computer-aided design (CAD) to remedy lack of robustness in design systems.

Sederberg and Farouki first proposed using interval Bezier curves to approximate arbitrary smooth functions [18]. The range of values represented by these intervals take into account all sources of measurement uncertainty. Tuohy et. al extended interval Bezier curves to interval B-spline curves to approximate the data points of a reverseengineered CAD model [22]. Others have studied the representation and geometric operations of interval B-spline curves, such as those used to solve CAD intersection problems [3,12,19].

We use the interval spline curve to approximate not the uncertainty in measured data, but rather the uncertainty of ink applied by a brush. Here we introduce the mathematics of interval splines and interval spline curves. Though the interval method applies to many types of splines, we discuss the particular cases of B-splines and interpolatory splines.

4.1 Interval B-splines

Replacing the constant coefficient of the familiar B-spline basis function with an interval of real numbers results in an interval B-spline defined by a function of the following form

$$[B](t) := \sum_{i=0}^{n} (c_i + \varepsilon_i[e])\, N_i^k(t) = c(t) + [e]\varepsilon(t) \qquad (1)$$

where $c(t) = \sum_{i=0}^{n} c_i N_i^k(t)$ is the *center curve* and $\varepsilon(t) = \sum_{i=0}^{n} \varepsilon_i N_i^k(t)$, with $\varepsilon_i > 0$ and $[e] = [-1,1]$. $N_i^k(t)$ is the B-spline basis function of order k associated with knot sequence $U := \{t_o, t_1, ..., t_{n+k}\}$, where $t_o \le t_1 \le ... \le t_{n+k}$ and $t_i < t_{i+k}$ for i between 0 and n. $N_i^k(t)$ can be defined by the following recursive formula:

$$N_i^1(t) = \begin{cases} 1, t \in [t_i, t_{i+1}) \\ 0, t \notin [t_i, t_{i+1}) \end{cases} \qquad (2)$$

$$N_i^k(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_i^{k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1}^{k-1}(t)$$

where $i = 2,...,n$. The B-spline basis functions $N_i^k(t)$ are non-negative over $t \in (-\infty, \infty)$, hence

$$[B](t) = [B_{\min}(t), B_{\max}(t)] \qquad (3)$$

In this result, $B_{\min}(t) = \sum_{i=0}^{n} a_i N_i^k(t)$ is the lower bound of $[B](t)$ and $B_{\max}(t) = \sum_{i=0}^{n} b_i N_i^k(t)$ is the upper bound of $[B](t)$.

4.2 Interval B-spline curves

Building on the interval B-spline representation is the interval B-spline *curve.* An interval B-spline curve takes its coefficient values from control shapes, vectors stored at its knots.

$$[\mathbf{B}](t) := \sum_{i=0}^{n} [\mathbf{P}_i] N_i^k(t) \qquad (4)$$

where $[\mathbf{P}_i] = (c_i + \varepsilon_i[e]) \times (d_i + \delta_i[e])$.

4.3 Interpolatory interval spline curves

Though a basic B-spline is easily implemented, it is not intuitive for use by a non-mathematician as the curve does not pass though its knots. It can be difficult to determine the exact position of the curve by adjusting the knot values. We gain finer control through use of a curve that passes through each knot, an interpolatory spline.

An interpolatory interval spline curve [9] is defined such that its center curve interpolates a given set of points. Given a set of points $P_i, i = 0,1,\cdots,N$, we use a spline curve $c(t)$ to interpolate the given set of points. That is,

$$c(t_i) = P_i, i = 0,1,\cdots,N \qquad (5)$$

where the parameters are chord length parameters, that is, $t_i = \sum_{j=0}^{i-1} l_j$, l_j is the length of P_jP_{j+1}. If we impose two boundary conditions for the cubic spline curve $c(t)$, for example,

$$c''(t_0) = c''(t_N) = 0 \qquad (6)$$

$c(t)$ can be uniquely determined by solving a tridiagonal system of linear equations. Though one would expect such computations to be expensive, this system of equations can actually be solved in O(N) time. This fact and the ease of control offered by interpolatory spline curves make them the optimal curve representation for use in our system.

4.4 Visualization of interval spline curves

Interval spline curves consist of a family of interval splines whose control points are located inside the ranges defined by the interval control points. The interval spline itself is defined by two splines, an upper and lower bound. On the other hand, the interval spline curve should be considered as defined by the control shapes stored at its knots. The interpolatory interval spline curve is perhaps more easily visualized than the B-spline. Consider the example in Figure 1 on the following page. Construction of the curve is initiated when the pen is put to paper with the definition of the initial control ellipse. As the size of the brush changes as it is swept along the length of the stroke, the brush characteristics recorded at fixed periods become the control knots of the curve.

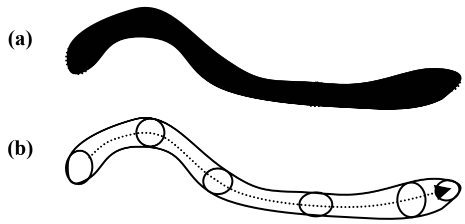

Figure 1 A brushstroke approximated by an interpolatory interval spline curve. **(a)** The original brushstroke. **(b)** Approximating (a) with an interpolatory interval spline curve defined by control shape (in this case, ellipses).

5 ARTISTIC INTERVAL SPLINES

We utilize both interval splines and interval spline curves to simulate artistic stroke effects. The interval curve is implemented as a family of splines constrained by control knots and rendered on-screen using a number of position evaluator functions.

5.1 Variable-width lines

Hand-drawn strokes are rarely as uniform in thickness as those generated by computer. Thus, a requirement in simulating this traditional media is the simulation of lines of varying width. Several such methods have been attempted in the past, including rib vectors [16] and skeletal strokes [11]. As discussed in the previous section, variable-width strokes can be easily described, with a minimal number of knots, as either interval splines or interval spline curves.

5.2 Stylistic effects

After defining the outline of a variable-width stroke, we can apply the stylistic variations to achieve the desired aesthetic effect. By storing width and color information in the control rectangles, the interval spline allows easy application of styles described in mathematical terms. The sketched style shown in Figure 2(a) is achieved by adjusting the number of curves in the family of the interval spline and the number of evaluated segments in those curves, and randomly perturbing the width and direction of these curves.

We simulate the painted style of Figure 2(b) by evaluating probabilities of ink density (in terms of color and transparency) along the lengths between control rectangles. The density of ink is often greatest when the brush initially touches paper, tapering off as the stroke is completed. Ink is distributed more sparsely at stroke ends and edges and more densely at thick corners. A degree of randomness controls the transparency, "flecks", and "veins" found along the length of a stroke and particularly common at endings.

Figure 2 **(a)** Pencil-sketched and **(b)** painted brush styles.

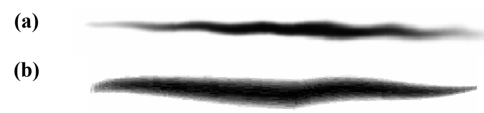

Figure 3 **(a)** Texture-mapped stroke by Northrup and Markosian [16]. **(b)** Stroke generated by our system with ink density controlled by probability distributions.

In addition to the default position evaluator, color and texture evaluators can be enabled to generate texture coordinate. Texture-mapping can be used along the stroke length to give the appearance of bristle lines or on stroke endings for a rough, "dry brush" appearance. Aesthetically, the results from the interval spline method compare favorably to those of texture-mapping as can be seen in Figure 3.

6 RESULTS

We measure the success of our model both quantitatively, in terms of rendering efficiency, and qualitatively, in terms of aesthetics.

6.1 Efficiency

Because a family of curves must be evaluated, one would expect the rendering of an interval spline curve to complete in real-time. In fact, interval spline curves render fast enough for use in an interactive system. While this has already exceeded our expectations, the fastest rendering speeds are achieved once the user has completed his composition and the final image is rasterized for display.

6.2 Painting with interval splines

The success of any simulation of traditional media is measured ultimately not by the ruler or stopwatch of an engineer, but rather by the eye of an artist. Here we present a system for users to "paint" in calligraphic styles with interval spline strokes.

The user first defines the curve outline by placing and adjusting the parameters of its control rectangles. Stylistic variations are then applied along the length of the

Figure 4 A swan painted in calligraphic style using artistic interval spline brushstrokes. Based on a painting by the Chinese artist Fang Zen [6].

outline. A drawing can be composed of a single continuous spline or can be composed of a number of discrete curves.As can be seen in Figures 4 and 5, a user can quickly create an impressive painting with this interactive system. Each image was created from scratch in only a few minutes.

Only a few expressive strokes are needed to suggest the image of the swan in Figure 4. Varying widths of probability-distribution fill strokes are used to suggest feathers, beak, and water.

Notice the mixture of stroke styles used to render Figure 5. Ribbon-like strokes are trace the folds of the wiseman's robe. Thin strokes of varying width are used for the delicate features of his head and face. A thick probability-distribution fill suggests the bushy beard.

7 DISCUSSION AND FUTURE WORK

We have approached brushstroke simulation not as a texture-mapping problem or physical simulation but rather as the 2D mathematical problem of rendering strokes using interval splines. Storing dense information vectors at its knots, the parameters of the interval spline are easily manipulated. The significance of the current approach is its ability to produce the artistic strokes faster and with greater flexibility. By defining styles as mathematical constraints between curve points, our system produces aesthetic strokes that both render faster than traditional strokes and resist scaling. The general nature of this approach suggests other applications to which it can be extended.

Additional artistic effects. We have simulated a number of them with this system, but it would be worthwhile to attempt others in the future. Specifically, there are a number of characteristics of the "dry brush" method not sufficiently captured by our current stroke model. It would be interesting to integrate our representation with other related systems, such as hairy brushes or computer-generated watercolor.

Integration with pen-based devices. Though a skilled user can quickly create an expressive composition with our system, the interface is hardly as intuitive to use as brush and paper. Thus, our future work involves integrating this system with a pen-based input device capable of recording angle, width, and pressure. The appearance of these curves would more easily incorporate the intent of the user as the setting of the control knots are set with this additional information about the user's application of the stroke measured through time.

Acknowledgements

The authors wish to thank the anonymous reviewers for the valuable feedback they provided.

Figure 5 A wiseman rendered in Eastern calligraphic style using our artistic stroke system.

References

[1] T. Baudel. A mark-based interaction paradigm for free-hand drawing. In *Proceedings of UIST 1994*, November 1994.

[2] T.W. Bleser, J.L. Sibert, and J.P. McGee. Charcoal sketching: returning control to the artist. *ACM Transactions on Graphics, 7*(1), pp. 76-81, 1988.

[3] W.Cho, T.Maekawa, N.M. Patrikalakis, and J. Peraire. Topologically reliable approximation of trimmed polynomial surface patches. In *Graphical Models and Image Processing*, Vol. 61, 1999.

[4] Y.S. Chua and C.N. Winton. User interface for simulating calligraphic pens and brushes. In *Proceedings of the ACM Sixteenth Annual Conference on Computer Science*, 1988.

[5] C.J. Curtis, S.E. Anderson, J. E. Seims, K.W. Fleisher, and D.H. Salesin. Computer-generated watercolor. In *Proceedings of SIGGRAPH 1997*, pp. 421-430, 1997.

[6] W.T. De Bary, W.-T. Chan, and B. Watson. *Sources of chinese tradition* (Columbia University Press, 1960).

[7] F. Zen. *Paintings of Fang Zen* (1999).

[8] P. Haeberli. Dynadraw. Silicon Graphics Corporation, 1989.

[9] J. Hoschek and D. Lasser. *Fundamentals of computer aided geometric design*, A.K. Peters Ltd., 1993.

[10] S.C. Hsu and I.H.H. Lee. Drawing and animation using skeletal strokes. In *Proceedings of SIGGRAPH 1994*, pp. 109-118, July 1994.

[11] S.C. Hsu, I.H.H. Lee, and Neil E. Wiseman. Skeletal strokes. In *Proceedings of UIST 1993*, November 1993.

[12] C.-Y. Hu, T. Maekawa, N.M. Patrikalakis, and X. Ye. Robust interval algorithm for surface intersections. In *Computer Aided Design*, 1997.

[13] J. Lansdown and S. Schofield. Expressive rendering: a review of nonphotorealistic techniques. *IEEE Computer Graphics and Applications*, *15*(3), pp. 29-37, May 1995.

[14] L. Markosian, M.A. Kowalski, S.J. Trychin, L.D. Bourdev, D. Goldstein, and J.F. Hughes. Real-time nonphotorealistic rendering. In *Proceedings of SIGGRAPH 1997*, pp. 415-420, August 1997.

[15] R.E. Moore. *Interval analysis* (Englewood Cliffs, NJ: Prentice Hall, 1966).

[16] J.D. Northrup and L. Markosian. Artistic silhouettes: a hybrid approach. In *Proceedings of NPAR 2000*, pp. 31-38, June 2000.

[17] S. Schlectweg. Lines and how to draw them. In *Norsk smarbeid inner grafisk databehandling NORSIGD Info*, February 1997.

[18] T.W. Sederberg and R. Farouki. Approximation by interval bezier curves. In *IEEE Computer Graphics and Applications*, 1992.

[19] G. Shen and N.M. Patrikalakis. Numerical and geometric properties of interval b-splines. In *International Journal of Shape Modeling*, Vol. 4, 1998.

[20] S. Strassman. Hairy brushes. In *Computer Graphics*, *20*(4), pp. 225-232, August 1986.

[21] S.T. Tuohy, T. Maekawa, G. Shen, and N.M. Patrikalakis. Approximation of measured data with interval b-splines. In *Computer Aided Design*, 1997.