ACM SIGGRAPH / Eurographics Symposium on Computer Animation (2024) Poster
M. Skouras and H. Wang (Editors)

# Organic Brushstrokes

William J. Joel
Computer Graphics & Animation Research Projects
wjoel@optonline.net

**Abstract**

*The main goal of this project was to determine (a) if operations could be developed for brushstrokes like those generally performed on 2D geometric objects (scale, rotate, translate, copy, etc.) and (b) if new operations could be developed that provide users new ways to manipulate brushstroke paths.*

**CCS Concepts**

• ***Computing methodologies*** *~ Computer graphics ~ Shape modeling • Applied computing ~ Arts and humanities ~ Fine arts*

## 1. Introduction

With respect to simulating brushstrokes, there have been numerous methods for creating images composed of strokes [WCW*18][CAS*97][MKT*97], as well as animating strokes [XXK*06][SWT*05][HL94]. However, methods for editing and manipulating brushstrokes, are lacking. Brushstrokes are generally represented as two-dimensional bit-mapped images. Though geometric data may be used to generate a brushstroke, once it is drawn this data is lost. A project was developed to see if brushstrokes could be edited like other geometric objects unique to brushstrokes. Brushstroke appearance issues were set aside as this has been well studied and documented [Str86][MSB*17].

## 2. Concept

Each brushstroke is defined by a spine that controls its shape, with attributes that address how the stroke is drawn. Though numerous authors describe rendering CGI brushstrokes [Str86][MSB*17], a discussion of how to manipulate strokes once drawn is lacking. By retaining spine information, each stroke can be manipulated in a manner similar to other geometric objects (polygons, curves, etc.). Brushstrokes often lack a defined boundary, making the brushstroke's edge essentially nonexistent. Similarly, given how the interior of a brushstroke depends on multiple factors, such as painting style, ink/paint type, paper type, the definition of a brushstroke's interior does not align with traditional models for defining a geometric object's visual appearance. Therefore, given the wealth of work focusing on the rendering of brushstrokes, the current project attempts to address the mechanics of using a spine to define a stroke's overall shape, and the editing methods that subsequently become available. These methods include selection, scaling, rotation, translation, reshaping, splits, joins, etc.

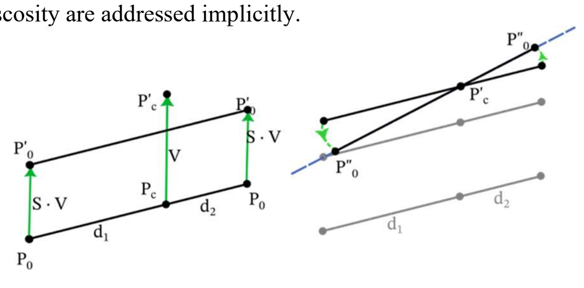
## 3. Operations

Previous work [Joe06] addressed a method for delaying the movement of a stroke when dragged. This new project extends the earlier project focusing on the interaction between a paint medium and a surface with regard to the medium's fluidity. Nodes are evenly distributed along the spine., with a sufficient quantity of nodes to reflect the stroke's flexibility. Too few nodes, and the stroke is too rigid. Too many, and the stroke bends too easily.

Five new operations were implemented during this project: push/pull, drag, split, join, extend.

**Push/Pull**

There are several examples of rendering brushstrokes using equations based on the physical state of the materials involved. These methods can produce wonderful results but are time-consuming. Further, they may not be necessary when the main goal is to edit a brushstroke's shape and location, ignoring the stroke's finer visual elements. A simpler approach is to incorporate material attributes at a conceptual level, avoiding the use of more intensive computations. Whereas the earlier paper mentioned above delayed the motion of a brushstroke as it was moved, the current model attenuates the "bend" of a stroke as it is being reshaped. Reshaping is therefore defined as moving a brushstroke by grasping a part of the stroke and moving it. During this process, the attraction/friction between the substrate and the paint medium, and the paint's viscosity are addressed implicitly.




(a) (b)

**Figure 1:** *Two-step movement of spine segment*

First, the spine segment closest to the cursor is determined, and a contact point, Pc, on the segment determined. Next, the distance, d, from the endpoint furthest from the contact point, P0, is computed. A vector, V, is obtained representing how far and in what direction the contact point has moved. A pre-determined percentage, S, of this vector is then applied to both segment endpoints.

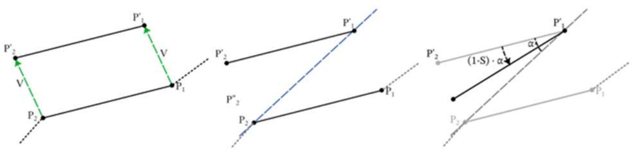


DOI: 10.2312/sca.20241164

Second, a line is found that passes through the new position of the contact point and the new position of the segment's endpoint furthest from the contact point. The farthest endpoint is then moved to this line, d units from the contact point. The other endpoint is also moved to this line. This provides the appearance that there is friction between the stroke and the substrate.

**Drag:** Defined as moving a brushstroke by grabbing a terminal endpoint and moving it, thereby moving the entire stroke. When a brushstroke is moved by pushing/pulling a segment, the sub-spines on either side are also moved using this method.

First, the selected endpoint, P1, is moved, and a vector, V, is determined representing how far and in what direction the endpoint moved. This vector is then applied to the next node, P2, in the spine. The old and new positions for P2 are stored for later use. A line is determined that passes through the new endpoint position, P'1, and the old subsequent node's original position, P2. Finally, P'2 is rotated towards this line using the percentage, S, introduced above. This method is then applied to all remaining interior nodes.



(a) (b) (c)

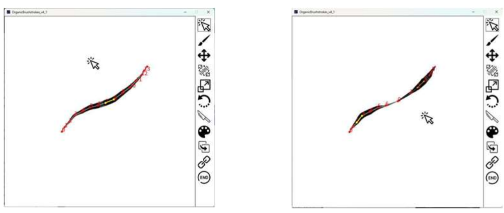
**Figure 2:** *Method for dragging a brushstroke*

**Split:** Determine a point on a segment where the cursor path intersects with the segment. Vertices below and above this intersection point define two new strokes respectively, with the intersection point becoming a new endpoint for each new stroke spine.

**Join:** Select two strokes and determine the endpoints, one from each stroke, closest to each other. Next, a new spine segment is inserted between the two closest endpoints. Additional nodes may be added if the distance between these endpoints is too big.

## 4. Discussion

After deciding to render strokes based on spines tangential issues arose. For example, when pushing/pulling spine segments, it became clear that the width of the stroke needed to be considered.



(a) (b)

Figure 3: Method for splitting a brushstroke

Displaying vertex indices led to reversing the order of vertices stored for a given stroke to simplify joins, turning the problem into concatenation of two sequences. If a user wished to extend a stroke from what is initially vertex 0, vertices in the stroke were reversed, allowing the extension to become a continuation of the original stroke drawing action.

Once the push/pull and drag operations were implemented, the movement of the brushstrokes did not seem to reflect the adhesion of a stroke's medium to a surface. An option was added to the above two operations such that when a node is moved, the distance between it and the subsequent node can increase or decrease, depending on the movement's direction. This modification was tied to the stiffness percentage, *S*, previously mentioned.

## 5. Conclusions

A variety of 2D operations were implemented working with and manipulating brushstrokes similar to other geometric objects. Future work will explore how these new operations can be implemented within various paint programs. Though graphic design software can often accommodate these new operations, and some already have similar operations, it is hoped that these new methods for working with brushstrokes can be added to image processing software. Future work will also focus on identifying and implementing additional operations that could benefit users.

## References

[XXK*06] Xu, S., Xu, Y., Kang, S.B., Salesin, D., Pan, Y., Shum, H-Y. Animating Chinese Paintings Through Stroke-Based Decomposition. *ACM Transactions on Graphics*, 25(2). April 2006, Pages 239–267.

[SWT*05] Seah, H.S., Wu, Z., Tian, F., Xiao, X., Xie, B. Artistic Brushstroke Representation and Animation with Disk B-Spline Curve. *Proceedings of ACE 2005*, Valencia, Spain, pp. 88-93.

[WCW*18] Wu, R., Chen,Z., Wang, Z., Yang, J., Marschner, S. Brushstroke Synthesis with a Generative Adversarial Network Driven by Physically Based Simulation. *Proceedings of Expressive '18*, August 17–19, 2018, Victoria, BC, Canada.

[CAS*97] Curtis, C.J., Anderson, S.E., Seims, J.E., Fleischer, K.W., Salesin, D.H. Computer-Generated Watercolor. SIGGRAPH '97: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, August 1997, pp 421–430.

[HL94] Hsu, S.C., Lee, I.H.H. Drawing and Animation Using Skeletal Strokes. SIGGRAPH '94: *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, July 1994, pp 109–118.

[MKT*97] Markosian, L., Kowalski, M.A., Trychin, S.J., Bourdev, L.D., Goldstein, D., Hughes, J.F. Real-Time Nonphotorealistic Rendering. SIGGRAPH '97: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, August 1997, pp 415–420.

[Str86] Strassmann, S. Hairy Brushes. SIGGRAPH '86: *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, August 1986, pp 225–232.

[MSB*17] Montesdeoca, S.E.,Seah, S.H., Bénard, P., Vergne, R., Rall, H-M. Edge- and substrate-based effects for watercolor stylization. *Proceedings of NPAR'17*, July 29–30, 2017, Los Angeles, CA, Article No.: 2, pp. 1–10.

[Joe06] Joel, W. Delayed feedback in a Chinese brush painting interface. *SIGGRAPH '06*: ACM SIGGRAPH 2006 Research posters, July 2006, pp 144–es.


Proceedings published by Eurographics - The European Association for Computer Graphics.