Real-time Ornamental Calligraphic Pens

Lena Polke[1] and Jürgen Richter-Gebert[2]

Dept. of Mathematics, Technical University Munich, Germany
[1] polkel@ma.tum.de, [2] richter@ma.tum.de

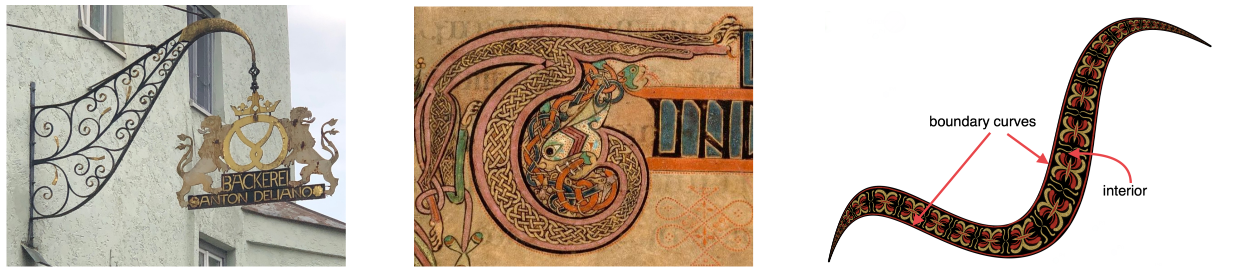
Abstract

This paper presents an approach to "rich" digital calligraphy pens. These pens draw strokes consisting of an elaborate ornamental structure in real time. The structure's density and complexity depend on the stroke width which itself is controlled by the pen pressure. In particular, we describe a process that, driven by pen data and applied during the drawing process, dynamically combines variable basic building blocks to create aesthetically pleasing ornaments within the stroke. In addition to the algorithmic challenges posed by the real-time scenario, we address several mathematical problems as diverse as discrete conformal mappings, the geometric properties of logarithmic spirals, and the generation of fractal branching structures.

Introduction and Motivation

During the last few decades, many areas of daily life have been digitized. The use of a tablet together with a digital pen opened up a variety of possibilities in the world of writing and drawing, among others. Applications in art are a particularly challenging field of development and research. To provide highly expressive pens and brushes, drawing programs ultimately need algorithms to convert the pen's motion into artistic output. To manage this, they have to answer the question of what a simulated stroke should look like.

There are many different kinds of real-world pens. Some pens draw a single curve such as pencils and ball pens for example. Others draw strokes having a fixed or variable width that sometimes depends on the pressure applied while drawing. These include fountain pens, brushes, and felt-tip pens. Such a stroke consists of two boundary curves and an enclosed interior as exemplary denoted in Figure 1c.



(a) *photograph from Wasserburg* **(b)** *extract from the Book of Kells* **(c)** *computed with given ornamental tile*

Figure 1: *Examples of strokes filled with ornamental patterns.*

Here, we aim for digital pens that carry some of the properties of real pens but we want to enrich them algorithmically. The special feature in this context is that the interior of the drawn stroke is filled with ornamental patterns in real time during the drawing process. Ornamental patterns are decorative designs that may have various forms. The ornaments have to be generated or adapted so that they adjust nicely to the available space of the stroke. Although these terms are difficult to pin down, harmony, balance, and evenness play a decisive role in our approach. At first glance, it may seem artificial since pens that fill a stroke with a pattern do not exist in the real world. However, in the art of calligraphy one often finds highly decorated single strokes of exactly this kind, drawn using elaborate penmanship. It is only natural to ask for a computer-aided hyperrealistic tool to provide this effect automatically.

Figure 1 shows examples of strokes whose interior is filled with different ornaments: 1a is a real-world example of wrought-iron artwork found in the German city of Wasserburg, 1b is an artwork from the famous Book of Kells where the ornamental knot patterns are inserted into the curved letter by hand, and in 1c an ornamental tile has been repeatedly inserted into a given stroke by our proposed algorithm discussed later. Observe how in examples 1a and 1b the intricacy and level of detail varies with the local width of the stroke.

The picture from the Book of Kells from the 9th century shows that the idea of inserting ornamental design into strokes was developed a long time ago. An early example of manipulating pictures along strokes with computer programs can be found in [4]. They define a skeletal stroke to be a picture around some reference line. Their algorithm deforms the pictures in such a way that the reference line follows a given path.

With the task of developing software that fills drawn strokes with artistically pleasing ornaments in real time, artists benefit from the variety of pens and ornamental patterns within one application that nevertheless leaves enough freedom to give free rein to their creativity. It is important to emphasize that the artistic intention is fully preserved throughout our enriching calculations. The stroke itself is not changed, only the interior is filled with different kinds of ornamental structure. This gives the user full artistic freedom, while the algorithmic enrichment is completely controlled by the underlying geometric principles. In addition, the mathematics behind the software represents a significant advance in the field of conformality. To the best of our knowledge, no method of real-time calculations exists for the treatment of constantly changing shapes while keeping distortion to a minimum. In the following, we will refer to existing research from time to time and point out the differences between these works and our approach.

In what follows, we first have to generate suitable simulations of calligraphy pens. We address this problem briefly in the next section since it is the basis of the following part of the processing chain. Second, an algorithmic approach is provided that maps given fixed ornamental patterns to strokes. Later on, it is explained how the ornamental design is to be created to fit the individually drawn stroke exactly.

Simulating the Pen

Electronic drawing is done with the help of a mouse, touchpad, or electronic pen. The data provided by these “input devices” is used together with an underlying (geometric) model to simulate the behavior of a pen. The position of the input device is given during the drawing process in the form of a registered sequence of points on the drawn curve. Based on this, the whole stroke is calculated depending on the information created by the input device adjusted by the user to their individual needs through some parameters.

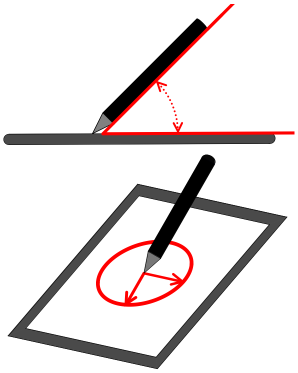
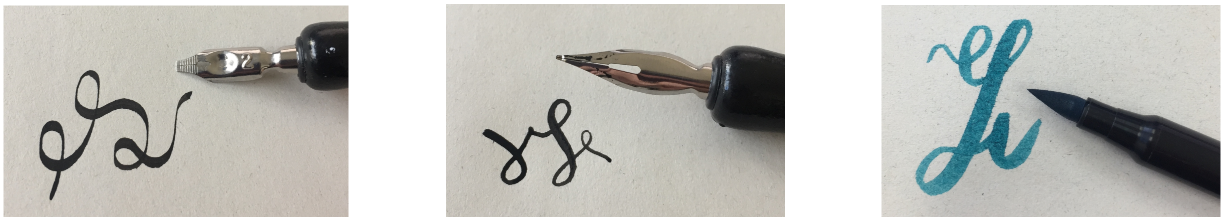


Figure 2: *Tilt and azimuth.*

Electronic pens provide additional information like the pressure applied to the drawing surface, the tilt of the pen relative to the tablet, and the orientation or azimuth of the pen, i.e., the direction in which the pen tip points (Figure 2). In what follows, we concentrate on the case of a user drawing with an electronic pen providing all this information.

There already are some advanced simulations of strokes making use of the features provided by electronic pens. For example, there are several studies of the generation of Chinese calligraphy from virtual brushes (see for example [7], [8]). Their approach, however, is not suitable for our task since they attach importance to the simulation of the behavior of the brushes’ bristles and the distribution of ink on the paper, for example. Here, we aim for a pen simulation that uses the technical data provided by the electronic pen to directly support our goal of mapping ornamental structure to the interior of the stroke during the drawing process.

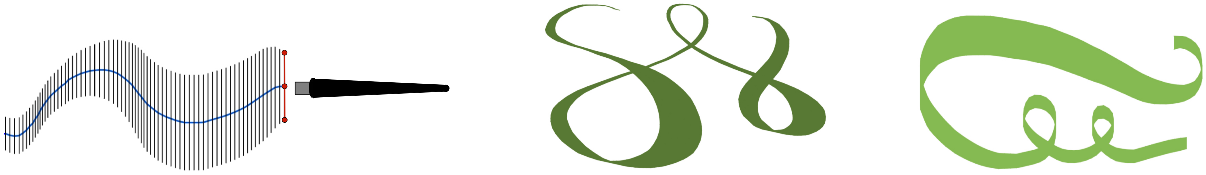
We paradigmatically chose three types of real calligraphic pens, the properties of which will guide the simulation: edged pens, fountain pens, and soft brush pens (Figure 3). The imitation of real pens is intended to evoke a natural feeling of drawing.



(a) *an edged fountain pen* **(b)** *a pointed fountain pen* **(c)** *a soft brush pen*

Figure 3: *Examples of calligraphic pens and the strokes they produce.*

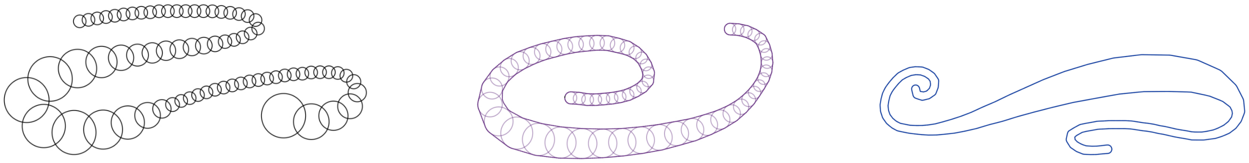
Our simulations are divided into two different types: the virtual edged pen and the so-called *round* pen. Both react to the pressure sensitivity of the input device by adapting the width of the drawn stroke to a user-defined intensity. The main difference is in the orientation sensitivity of the pen's tip in combination with the drawing direction. The orientation of the tip of an edged pen has a direct influence on the width and the boundaries of the drawn stroke. Think of an edged pen as a potentially pressure-sensitive highlighter. In contrast, a stroke drawn by a pointed or brush pen is not driven in a particular way by the coherence of the stroke's width and the tip's orientation on the paper. These pens (Figures 3b and 3c) give rise to the simulation of what we call the virtual round pen.



(a) **(b)**

Figure 4: *Simulation of an edged pressure-sensitive pen.*

For the simulation of the edged pen, the orientation of the tip gives us the necessary information to derive the boundary curves of the stroke. Two boundary points are calculated for every new point on the registered drawn curve. They are located parallel to the orientation of the pen's tip at a distance to the registered drawn curve that depends on the calculated width according to the settings and the pressure applied (Figure 4a). The total of all segments between two corresponding boundary points gives the interior of the stroke (Figure 4b).



(a) **(b)** **(c)**

Figure 5: *Simulation of a round pressure-sensitive pen.*

The calculation of the round pen's boundary curves is less straightforward; in particular, if it is supposed to be a suitable input for the conformal mapping techniques we apply later on. Again, there is a curve of directly registered points and the pen's information about tilt and pressure sensitivity. In accordance with its given name, we combine this information to generate an increasingly long collection of circles with their center on the registered drawn curve (Figure 5a). Consecutive circles are supposed to intersect at a constant angle (say 90°). We chose this approach with the expectation that we could later use the structure of the perpendicular circles to calculate a suitable coordinate system within the stroke. However, this is still work in progress. To generate the boundary curves, a hull is calculated that surrounds the constantly growing collection of circles (Figures 5b and 5c). This is done locally and thus the calculation effort is low.

Algorithms for Fixed Patterns

In this section, we focus on the case where a given rectangular ornamental tile is mapped repeatedly to the calligraphic stroke. One may think of equipping a stroke with a suitably deformed frieze pattern. Thereby, our approach differs a lot from the one in [11]. There, the authors cut out pieces from a given pattern and glue them together, whether or not the result fits together at the edges or within the given stroke. We want the ornament pattern to perfectly match the interior of the stroke, deforming with a minimum of local distortion.

The two different types of simulated calligraphic pens result in different methods with differing difficulty for mapping an ornamental tile to the corresponding stroke. However, they have in common that a local coordinate system is calculated in the stroke's interior. This is a uniform framework to which every kind of pattern or ornamental structure can be mapped later on. As already emphasized, all happens in real time, contrary to the approach of Adobe Illustrator [1], where the stroke is drawn entirely before it is equipped with some pattern or structure.

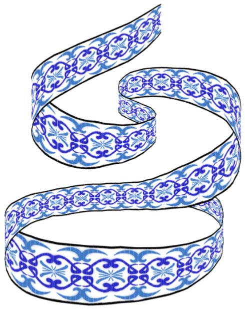


Figure 6: *A textured edged pen.*

In case of a virtual edged pen, the parallelism of the tip's orientation and the segments between corresponding boundary points already provides us with such a coordinate system. Also, the stroke can be considered as a ribbon that loops in space, so we take distortion into account when we map the pattern to it. This means, the more the stroke's course deviates from the axis perpendicular to the pen tip's orientation, the more the image is scaled down (Figure 6).

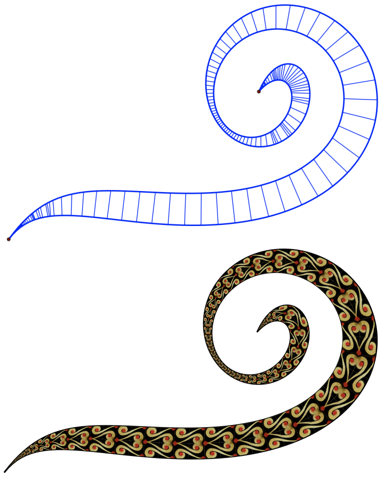
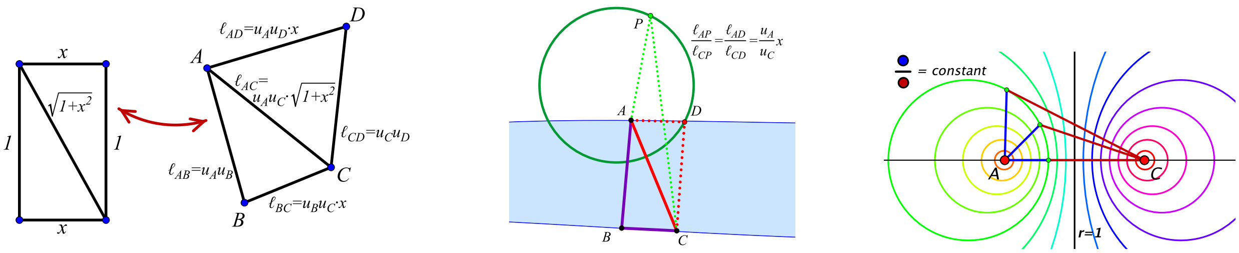


Figure 7: *Mapping ornamental pattern to quadrangles in a stroke.*

For round pens too, the requirement we place on the filling algorithm is that the artistic intention of the frieze pattern be only slightly disturbed as the ornament bends along the stroke. To achieve that, we aim for a close-to-conformal mapping of the pattern. In an ideal world, one could imagine an infinite stripe mapped conformally, i.e., angle-preserving, to the stroke, with the two infinite points mapped to the beginning and end of the stroke. Such a mapping exists as a consequence of the Riemann mapping theorem. However, two aspects of the problem force us to deviate from this idealized situation. Firstly, the real-time requirements allow only local computations although conformality is a global property. Secondly, while the beginning of the stroke may be considered to be infinitely small, the current position of the pen should be considered as a "loose end" that is subject to change as the stroke progresses. Since we do not want to re-calculate the conformal Riemann mapping with every change in the stroke, we have to define a local approximation of conformality to fulfill our requirements. Therefore, we use a local coordinate system.

We propose the following algorithm to fill a given stroke with a predefined ornament. Assume the stroke to be non-self-intersecting and non-branching. The algorithm takes only the next few boundary segments into account for its calculations and not the complete boundary curves. The outcome of the algorithm is a close-to-conformal quadrangulation of the stroke (Figure 7 above) calculated using results from the theory of discrete conformality [3]. The given rectangular ornamental tile is then mapped to these quadrangles so that the images fit together continuously along edges as in Figures 1c and 7 below. Notice that the quadrangles are refined in sharply bent regions of the stroke to reduce distortion in the pattern. The refinement is also applied to the pattern so that smaller parts of the ornamental tile get mapped to finer quadrangles.

For the following, consider Figure 8b. To obtain the quadrangulation, we place one starting edge AB inside the stroke. This and all future placed edges have to be as perpendicular to both boundary curves as possible while not leaving the stroke's interior. To find the position of the second edge closing one quadrangle, two points C and D have to be determined, one on each of the boundary curves. Therefore, we divide the quadrilateral by its diagonal into two triangles and we use the results from discretely conformally equivalent triangulations following [3]. As mentioned before, approximation of conformality is crucial here in order to get minimal local distortion of the filling ornament. Two combinatorially equivalent triangulations are defined to be discretely conformally equivalent if all edge lengths are related by scalar factors induced by weights associated with the vertices. For every pair of edge lengths, ℓ_{ij} and ℓ'_{ij} of the two corresponding triangulations, the condition $\ell'_{ij} = u_i u_j \ell_{ij}$ has to hold for (so far unknown) real weights u_i, u_j attached to vertices i and j of one of the triangulations. In our setting, the image triangulation is the one we want to generate inside the stroke. The reference triangulation refers to the given sequence of divided rectangular tiles that have to be mapped to the stroke.



x
$\sqrt{1+x^2}$
1
1
x
$\ell_{AD}=u_A u_D \cdot x$
D
A
$\ell_{AC}= u_A u_C \cdot \sqrt{1+x^2}$
$\ell_{CD}=u_C u_D$
$\ell_{AB}=u_A u_B$
C
B
$\ell_{BC}=u_B u_C \cdot x$
P
$\frac{\ell_{AP}}{\ell_{CP}}=\frac{\ell_{AD}}{\ell_{CD}}=\frac{u_A}{u_C}x$
A
D
B
C
= constant
A
C
r=1

(a) *discrete conformal equivalence* **(b)** *the fourth point of a quadrangle* **(c)** *Apollonian circle pencil*

Figure 8: *Mathematical background of the ornamental algorithm.*

For every quadrangle in our stroke, this results in five equalities that have to be satisfied (Figure 8a). Since edge AB was preset, the weights u_A and u_B are known. It turns out that positioning one of the two unknown points (say C) on one of the boundaries provides us with additional information about the related triangulations. From this, we can deduce that the ratio between the unknown edge lengths ℓ_{AD} and ℓ_{CD} is fixed by the weights associated with vertices A and C (Figure 8b). The theory of Apollonian circles [2] tells us that the locus of all points satisfying this ratio lies on a unique common circle with its center on the line joining the two given triangle points A and C (Figure 8c). Furthermore, there is exactly one such circle for every possible ratio. For a ratio equal to 1, the circle degenerates to a perpendicular bisector of the segment between points A and C. The more the ratio tends to zero or infinity, the smaller the circle gets and the more it is bent around one of the points. Hence, we can move C to a location on one boundary curve so that this circle intersects with the other boundary curve and the constraint of perpendicularity of the resulting edge is fulfilled in the best way possible. The intersection point is defined to be the fourth point D (Figure 8b). We repeat this search with the latest inserted edge CD as the next starting edge and go on until the whole stroke is divided into quadrangles. The quadrangulation of the stroke establishes a local coordinate system to which the ornament is mapped.

Ornamental Strokes

Besides the mapping of a fixed ornamental tile to the stroke, it is desired to develop an adaptive system that creates ornamental designs directly in the given coordinate system inside the stroke. This design then depends on the space available and on the drawn stroke itself.

The idea of filling strokes with a varying ornamental pattern is not new as we could see in Figures 1a and 1b. But our approach of doing so in real time and our requirements on the ornamental design differ a lot from existing implementations for expressive pens.

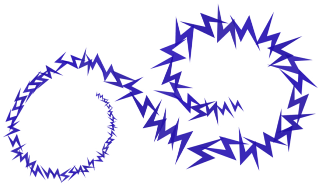


Figure 9: *A procedural pen from [5].*

Figure 9 shows an example of a stroke drawn by a procedural pen in the software Procreate [5] which places a flash design randomly along the drawn curve, with size proportional to the pressure of the pen. In contrast, our simulation is meant to satisfy the following properties: The ornament gets adjusted to the space available in the stroke's interior, and the more space there is, the more intricate the ornaments. The second requirement is in contrast to what we did in the last section where the same pattern is repeated (see Figure 7). Furthermore, the ornamental structure is to be generated making use of creation rules in such a way that the result looks aesthetically pleasing. Of course, the simulation happens in real time during the drawing process.

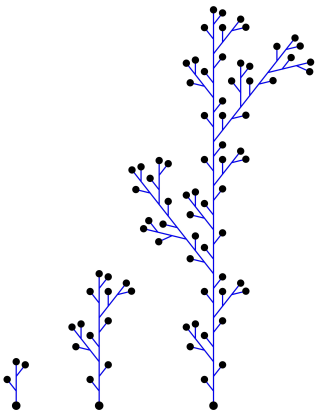


Figure 10: *L-system plant.*

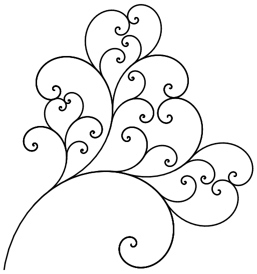


Figure 11: *Flourishing design.*

To define general semantics that creates ornamental designs according to the individual demands of the drawn stroke, we use different existing approaches to build our own “ornamental language”. One of them is Lindenmayer systems [6], L-systems for short. They rely on recursive rewriting rules that are used to develop complex structures from simple ones. A particularly pleasing aspect of such an approach is self-similarity. There are many fields of application such as the modeling of plants or the production of fractals. Figure 10 shows an example. We want to adapt the underlying logic in L-systems to define rules for the combination of simple building blocks so that the resulting image is an aesthetic ornament adjusted to the drawn stroke. The resulting texture can, for example, be built from floral elements like spiraling curves or from straight line segments that end up in an intertwined edged composition. Floral ornamentation has been researched by [9]. It is a very insightful work that algorithmically generates beautiful ornaments fitting any given region of the plane. However, their approach is not suitable for our problem since the whole region is known before the calculations start.

The composition of logarithmic spirals results in flourishing ornaments when some rules of substitution are defined properly. Figure 11 shows the result of experiments with logarithmic spirals with several iterations according to the growth rules of L-systems. Notice the visual similarity to the real-world example in Figure 1a. To obtain the visible curvature continuity and touching behavior, the geometric properties of the logarithmic spirals are combined with the combinatorial properties generated by the replacement system. The relation between different parts of the structure is intentional. This is in contrast to the approach of magnetic curves in [10] where decorative curves are created by particle tracing. The exact course of the curves cannot be fully controlled as it follows the physical model.

A general logarithmic spiral is defined by

$$f(\varphi) = ae^{k\varphi}e^{i(\varphi+\gamma)} + P$$

with $\varphi \in \mathbb{R}$ the independent variable, i the imaginary unit of complex numbers in $\mathbb{C}$, $a \in \mathbb{R}_+$ the scaling factor of the spiral, γ an angle of rotation relative to the x-axis, P the position of the pole, i.e., the translational factor, and $k \in \mathbb{R}\backslash\{0\}$ a parameter defining the angle between a radial line and a tangent of the spiral by $\arctan(\frac{1}{k})$. See Figure 12 for a visualization of the nomenclature.

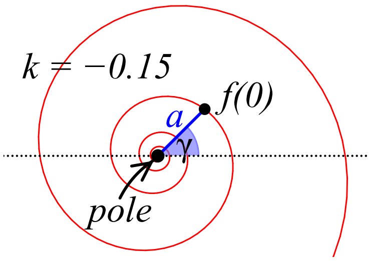
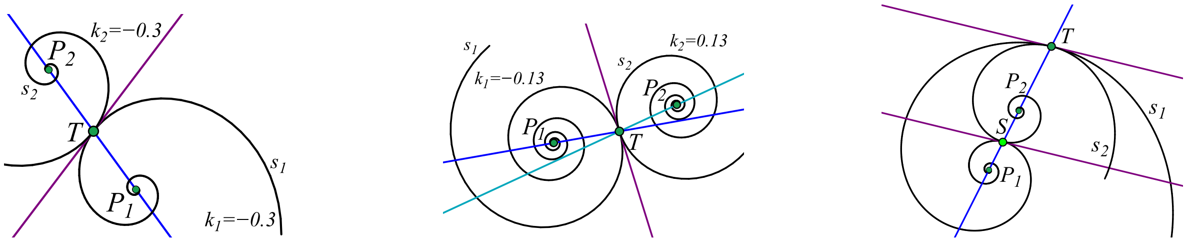


Figure 12: *A logarithmic spiral.*

If we follow the logic of L-systems and put together logarithmic spirals defined by the same parameters in an artistically pleasing way, one quickly comes to the conclusion that tangential spirals look very harmonious. This has interesting geometric implications. Two spirals s_1 and s_2 are tangential at some point T if and only if they share the same tangent at the common point. This sounds trivial but it gives rise to the conclusion that the tangential point T and the poles P_1 and P_2 of the two spirals lie in a special relationship with each other if both spirals have the same parameter k of equal or opposite sign.



(a) **(b)** **(c)**

Figure 13: *Tangential logarithmic spirals showing the geometric relations of poles and points of contact.*

There are several constellations of pairs of logarithmic spirals: they have the same (Figure 13a) or opposite rotation direction (Figure 13b) and they have either one or more than one point of tangentiality as, for example, in Figure 13c. For the same rotation direction and parameter k, the poles P_1 and P_2 and the tangential point T lie on a common line (see Figure 13a). If the rotational directions are opposite, the pole of the second spiral lies on the line P_2T which is the reflection of the line P_1T at the common tangent as the mirror axis (see Figure 13b).

Logarithmic spirals' special property of exponential radius growth makes it possible to calculate the exact positions of the pole or the contact points of tangential spirals. Therefore, given one spiral and searching for a tangential spiral inserted into it as in Figure 13c at a certain position $T = s_1(\phi)$, the following two formulas give the unknown spiral's pole P_2 and the second tangential point S:

$$S = s_1(\varphi - \mathrm{sign}(k) \cdot 2\pi) \qquad P_2 = T + \frac{e^{|k|\pi}}{e^{|k|\pi} + 1} \cdot (S - T)$$

This knowledge was used to define rules to combine several spirals tangentially to each other so that the result fits into the drawn stroke. Since spirals with the same parameter k in different sizes and positions result from each other only by translation, rotation, reflection, and scaling, it is easy to see that they fit perfectly in the scheme of basic building blocks of L-systems.

This results in images like the one in Figure 14. Notice that the degree of shown details in the spiraling ornamentation varies with the width of the drawn stroke. This corresponds to the application of more recursions of the defined rewriting rules on the basic building blocks.

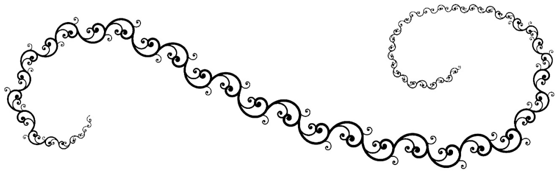


Figure 14: *Virtual calligraphic pen with an ornamental design of tangential logarithmic spirals.*

We aim to extend these methods in the future to an even more sophisticated ornamental combination of logarithmic spirals as well as to other basic structures that we can fit recursively into our strokes.

Summary

In this paper, we presented an ornamental calligraphic pen simulation. Two virtual pens – an edged pen and a so-called round pen – are presented as well as algorithms that allow the strokes to be filled with ornamental structures of different kinds. The prominent feature of this approach is the real-time requirement. The important concepts used to generate or manipulate ornamental patterns to fit a drawn stroke are the theories of discrete conformality, Lindenmayer systems, and logarithmic spirals. They describe the principles for the development of new mathematical semantics for the beauty of ornamental design.

References

[1] Adobe Illustrator User Guide. Website. https://helpx.adobe.com/illustrator/user-guide.html/illustrator/using/brushes.ug.html.

[2] R. Bamler, C. Reiher et al.. “Ein-Blick in die Mathematik.” *Aulis Verlag Deubner*, 2005.

[3] A. I. Bobenko, U. Pinkall, and B. A. Springborn. “Discrete conformal maps and ideal hyperbolic polyhedra.” *Geometry & Topology*, vol. 19, no. 4, 2015, pp. 2155–2215.

[4] S. C. Hsu, I. H. Lee, and N. E. Wiseman. “Skeletal strokes .” *Proceedings of the 6th annual ACM symposium on User interface software and technology*, 1993, pp. 197–206.

[5] Procreate. Website. https://procreate.art.

[6] P. Prusinkiewicz and J. Hanan. “Lindenmayer Systems, Fractals, and Plants.” *Springer Science & Business Media*, vol. 79, 2013.

[7] J. Shin, C. M. Kim, and M. A. Rahim. “Simulating Oriental Brush Character Considered with Aerial Action of Pen Tablet.” *Multimedia Tools and Applications*, 2019.

[8] H. T. F. Wong and H. H. S. Ip. “Virtual Brush: a Model-Based Synthesis of Chinese Calligraphy.” *Computers & Graphics*, vol. 24, no. 1, 2000.

[9] M. T. Wong, D. E. Zongker, and D. H. Salesin. “Computer-Generated Floral Ornament.” *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998, pp. 423–434.

[10] L. Xu and D. Mould. “Magnetic Curves: Curvature-Controlled Aesthetic Curves Using Magnetic Fields.” *Computational Aesthetics, 9*, 2009, pp. 1–8.

[11] S. Zhou, A. Lasram, and S. Lefebvre. “By-example synthesis of curvilinear structured patterns.” *Computer Graphics Forum*, Oxford, UK: Blackwell Publishing Ltd, vol. 32, no. 2pt3, 2013, pp. 355–360.