The 8th ACM/EG Expressive Symposium
C. Kaplan, A. Forbes, and S. DiVerdi (Editors)

EXPRESSIVE 2019

Aesthetics of Curvature Bases for Sketches

K. Lippincott[†] and R. L. Hatton[‡] and C. Grimm[§]

Abstract

In this work we propose a curve approximation method that operates in the curvature domain. The curvature is represented using one of several different types of basis functions (linear, quadratic, spline, sinusoidal, orthogonal polynomial), and the curve's geometry is reconstructed from that curvature basis. Our hypothesis is that different curvature bases will result in different aesthetics for the reconstructed curve. We conducted a user study comparing multiple curvature bases, both for aesthetics and similarity to the original curve, and found statistically significant differences in how people ranked the reconstructed curve's aesthetics and similarity. To support adaptive curve fitting we developed a fitting algorithm that matches the original curve's geometry and explicitly accounts for corners.

1. Introduction

Curves are used in a wide variety of applications, from traditional sketching to animation paths to data fitting. Often an artist or designer first sketches [GJ12, BBS08] the curve (using a tablet pen or equivalent), at which point the curve is represented as a simple list of connected points. Rarely is the curve kept in this format; instead, it is processed into a higher-level representation (such as a spline) using a mix of smoothing and fitting. This processing serves two purposes; first, it smooths or filters the data, removing noise due to the capture process, and second, higher-level curve representations can be re-sampled at will and support mathematical operations such as taking derivatives. As has been noted before, different curve representations have a different "look and feel" [OEYK12, FRSW87, MS09, WKSM12]. Curvature clearly plays a role in this, as does the fitting or smoothing process used.

In this paper we ask: what role does curvature play in determining the aesthetics of a curve? There are three pieces we need to explore this question. First, we need a higher-level curve representation that lets us experiment with different curvature *bases*. Second, we need a fitting method that is *faithful* to an original, sketched curve (in an L^2 sense). Third, we need to define a *context* for comparing aesthetics — it's likely that a curve that is aesthetically pleasing in a drawing of a flower will not be so in a drawing of a building.

Traditionally, higher-order curves have been represented as splines — essentially, piece-wise polynomials. This representation — along with traditional fitting approaches — tends to minimize maximum curvature, smoothing out noise but also usually smoothing out high-curvature features. More recent approaches have explored representing curves as glued-together pieces of curves, each of which have linear curvature (clothoid curves [OEYK12, MS09, BLP10]) or come from a family of pre-defined "nice" curves [MS11]. This is the approach we take, only we broaden it to include a variety of curvature bases, from constant to sinusoid. We provide a general-purpose algorithm for fitting a curve of this form to an input sketched curve that takes into account corners.

To evaluate the aesthetics of these different curvature bases, we turned to a crowd-sourced survey (Mechanical Turk). Participants were asked to evaluate both the aesthetics of the curves (presented as a drawing) and how similar the curves were to the original (fidelity of fit). We chose to conduct our evaluations on complete drawings instead of single curves, under the hypothesis that the content of the drawing will have an influence on what is perceived as aesthetic. Obviously, an extensive evaluation of all types of drawings is not feasible; we settled for a set of drawings that span simple to complex, and natural to man-made.

Contributions: Our contributions are 1) An analytic representation for curves based on curvature bases, 2) An algorithm for fitting to a sketched curve with a desired level of accuracy, which accounts for corners in the curve; 3) A user study that demonstrates that curvature bases do show a measurable effect in terms of aesthetics.

The paper is laid out as follows: Section 2 discusses previous curve fitting methods and aesthetic studies; section 3 describes the curve representation and how to reconstruct the curve from the curvature bases; in section 4 we describe the user study that we used to evaluate the aesthetics of the resulting curves; section 5 discusses the results of that study; section 6 describes the details of the fitting algorithm; and finally, section 7 contains concluding remarks and future work.

† Pariveda Solutions
‡ Oregon State University, Ross.Hatton@OregonState.edu
§ Oregon State University, Cindy.Grimm@OregonState.edu

DOI: 10.2312/exp.20191081

2. Related work

Curve fitting and smoothing has a long history [FRSW87], and a full review is beyond the scope of this paper. We focus here on methods that explicitly reconstruct the curve from the curvature or from a "basis" set of curves. The inspiration for our work is curve design using piece-wise clothoid curves [MS09], and the idea that the choice of mathematical representation of curves may have an aesthetic component [WKSM12]. In fact, Shaheen et. al. give a method that distinguishes between the sketching style of different artists by comparing mathematical representations of their sketches [SRG15]. Baran et. al. give a method for approximating curves using clothoid segments [BLP10] and Orbay et. al [OEYK12] extend this to evaluating product designs.

Curvature manipulation has long been recognized as a means for smoothing, or neatening, curves [MR07, FRSW87, BAFS94]. Unfortunately, too much curvature smoothing also tends to remove the "character" of the line by reducing sharp features. Stroke dynamics (i.e., the speed at which a user sketches) have been used to reduce this problem [TSB11], although this technique requires temporal data. An alternative is to fit a curve basis set to the raw data — the most common approach being Bézier curves [BBS08], B-Splines [WPL06], or equivalent piece-wise polynomial representations. Usually, the degree of the polynomials and the number of segments is kept low, which has the effect of setting the nth derivatives to zero and capping the number of times the curve bends. Recent work has also incorporated curvature constraints into the spline formulation itself [MG16].

A less-used alternative is conic splines [Pav83]. McCrae [MS11, Sin99] used French curve segments to neaten drawings; although this set of curves has no formal mathematical description, the curve shapes themselves have been refined based on aesthetic properties.

Different approaches have been used to characterize the aesthetic nature of a curve. In [KNS] they classify curves as divergent, neutral, and convergent, based on a designer's classification, and show that this classification can be linked to curvature. Similarly, in [YS06] they use curve segments based on linear logarithmic curvature. In [XM09] they move particles along trajectories with constantly changing curvature, similar to how we reconstruct curves only with a more limited basis set. In [Zit13] they focus on handwriting, and averages similar bits and pieces of the hand-written text to create more regular results. None of these papers validated the relative aesthetics of the curves.

3. Curvature basis representation

Figure 1 shows the basic format of our curve representation (upper left), and an example of each different curvature basis (upper right). Given a curve in 2D we can calculate the curvature at each point along the curve. Conversely, given the curvature, we can uniquely reconstruct the curve in 2D (up to a rotation and translation).

Rather than create a function that represents the curve's 2D geometry we create a function that represents the curve's curvature — and then integrate to get the 2D geometry. The middle row of Figure 1 shows curvature functions built using 8 different bases, and their corresponding reconstructed curves underneath. We define two *classes* of bases types — piecewise (the first 3) and multi-basis (the last 5).

We next provide more information on the bases functions and the reconstruction algorithm. In Section 6 we provide a windowing algorithm for fitting to a sketched curve — this algorithm was used to generate the examples in Figure 1.

3.1. Basis functions

Similar to splines, we can think of the curvature function as being broken up into n pieces, each of which is represented by a simple polynomial function (linear, quadratic) with positional or derivative constraints at the boundaries. We classify these as our piecewise polynomial bases. From left to right in Figure 1 we have 1) [PWL] piecewise linear — linear sections with positional constraints, 2) [QUAD] piecewise quadratic — quadratic polynomial sections with positional constraints, and 3) [QUADS] quadratic spline — quadratic polynomial sections with derivative constraints. In Figure 1 we have shown these with 8 segments each.

We can also represent each section as the *sum* of a set of weighted, ortho-normal basis functions. More formally, let $\{b_i : R \rightarrow R\}_{i \in [1,m]}$ be the set of basis functions, then we define the curvature as the weighted sum of the basis functions ($c(t) = \sum_{i \in [1,m]} w_i b_i(t)$). Figure 1 upper right shows our five types: Haar wavelets, Mexican Hat wavelets, Laguerre polynomials, Hermite polynomials, and sinusoid functions.

The multi-basis can also be split up into segments; in Figure 1 we use two segments, with the join indicated by the red line. We ensure positional continuity across the segments; this reduces the degrees of freedom by 1 (i.e., we have $m-1$ weights to set in the second segment, instead of m).

Our fitting algorithm optimizes both for the number of segments and the placement of the segment boundaries, taking into consideration both the curvature values and the original, 2D geometry.

3.2. Curve Reconstruction

We define each curvature function on the unit interval, centered at zero: $c(t) : [-0.5, 0.5] \rightarrow R$ so that we can reconstruct the curve from the middle outward (reducing computational error). The reconstruction algorithm starts at the origin and *integrates* a 2D coordinate frame forward and backward through time based on the curvature values.

We describe two methods for reconstructing the geometry of the curve from the curvature. This reconstruction is unique up to a rotation and translation; both methods start with a position and orientation of the $t = 0$ coordinate frame that defines the curve's position and orientation in space.

Our first method is function-based, and uses an ODE solver to integrate the coordinate frame through the curvature values. This method has the advantage that it returns a continuous fitted curve, but it tends to be slow and is not suitable for real-time non-linear optimization.

Our second method reconstructs the curve by inverting the discrete curvature calculation of section 6.1. Essentially, we use the

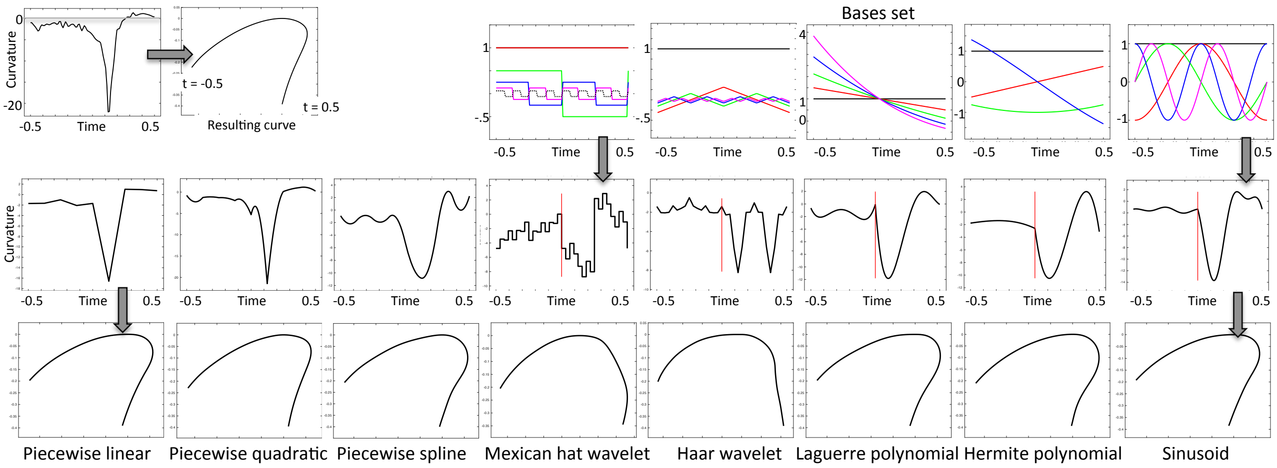

Figure 1: *Representing a curve from a curvature bases. In the upper left is a curve reconstructed from the curvature. We show the bases set for our multi-bases curves in the upper right. For each type we show the curvature function and the reconstructed curve. The piece-wise bases were split into 8 uniform segments, the multi-bases into two segments..*

curvature at a set of time points $t_j \in [0,1]$ to find the angle change at each point and add segments to the curve in the direction of this angle. To do this we need a good approximation of the length of the segment from t_j to t_{j+1}, which can be extracted from the original sketch if we are fitting to one. The two disadvantages of this method are 1) the accuracy of the reconstruction depends on the number of points reconstructed 2) it requires the segment lengths of the original curve. To address these issues we resample the original curve using arc-length parameterization using twice as many sample points as the original curve.

4. Curvature Aesthetics Study Design

The goal of our user study is to determine if the choice of basis function 1) has an effect on the aesthetics of a fitted curve and 2) the visual similarity of the fitted curve to the original. Recognizing that context has a role in aesthetics, we do not evaluate curves in isolation, but as a collection of curves making up a drawing. We reconstruct all of the curves with the same bases and ensure that the reconstruction for each type has approximately the same amount of error. For our survey, we use a forced-choice, inter-subject approach with two question types, one for aesthetics and one for similarity to the original.

We discuss the drawings, the questions, and the participants next.

4.1. The Drawings

Aesthetics are difficult to determine when considering a single abstract curve, thus we combined a variety of curves into recognizable drawings. Figure 2 shows the eight drawings that we created for the purposes of this study. These drawings were chosen in order to span a range of complexity. The hat drawing is the simplest drawing, made with only a few, relatively smooth curves. The elephant, boy, and cat represent a mid-range complexity. The bush, the Seattle skyline, and the mountainous scene represent more complex images with both linear elements and sharp curvature changes. By using multiple drawings we hoped to catch any differences in the aesthetics of the basis function that were due to the semantic information in the drawings and the types of curves they contain.

Figure 2: *The drawings used in the study. All eight drawings were made by tracing images found on publicly-available "how to draw" web sites. Shown are the original, unfitted drawings.*

4.2. The Questions

Our study contained two different types of forced-choice questions designed to measure both aesthetics and visual similarity. Question type 1 showed pairs of drawings and asked the user to select the one that seemed more aesthetically pleasing (see left of Figure 3). In question type 2 the user was asked to choose which of two reconstructed drawings was most similar to the original drawing (see right of Figure 3).

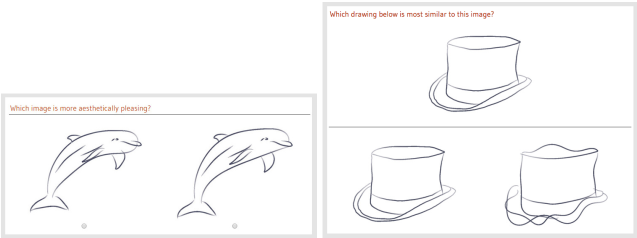

Figure 3: *Left: Question type 1, aesthetic. Right: Question type 2, similarity.*

For each of the eight drawings we generated nine variations of the drawing: the original drawing and one for each basis function type (the three piecewise polynomial and the five multi-basis). Each drawing variation was compared to all the others, with each combination shown in both orderings (left-right). This resulted in a total of 1152 questions (8 drawings $\times$ 9$\times$8 comparisons $\times$ 2 left-right).

Similarly, for the type 2 questions we generated all possible pairs (we included the original) with both left and right orderings.

From the full question set we uniform randomly selected 15 questions from each question type to avoid fatigue (30 total for each participant). In addition, we added in one attention-check question (type 2) with an obvious correct answer. This resulted in roughly 7-8 answers per question type.

We collected 466 responses via Amazon Mechanical Turk (AMT). No restrictions were placed on the participants.

5. Study Results

This section presents the results of the study, including our user attentiveness check and our check for order bias.

5.1. Terminology

We discuss results in terms of *appearances* and *votes*. We define an *appearance* to be any time a basis appears as an answer to a survey question and we define a *vote* to be any time a study participant selects the basis as the answer to a survey question.

In order to determine if a particular basis has a statistically-significant difference in number of votes than another basis we use a two sample t-test between them. Specifically, we construct an array of ones and zeros for each pair of basis functions. Each element in the array represents an appearance and each non-zero element in the array represents a vote for that basis. The mean of the array indicates which basis was voted for more often, and the p value indicates if this preference was statistically significant.

5.2. Attention Checks

We used four attention check questions in the survey — images such as the example shown in Figure 3 where (in preliminary tests) all of the participants picked the left image. Almost all of the survey participants correctly answered the attention check questions. Those who failed were still largely correlated with the overall averages. For this reason, we did not exclude any of the survey responses.

5.3. Order Bias

We placed each image on the left and right a roughly equivalent number of times to prevent order effects from skewing the results. We did find a small right bias throughout our survey, which we hypothesize is due to people picking the right image more often if they had no opinion. Here we compare the proportion that the right side image was picked to the expected proportion 50% which would result if no order bias were present.

When answering the first question group (aesthetics) users chose the right hand image 53.1% of the time. This was statistically significant with a z score of 5.12 and p-value of $1.52e-7$. When answering the second question (similarity to original), users chose the right image 51.1% of the time (z score of 1.574 and a p-value of 0.0577), which was less statistically significant.

We hypothesized that there is a stronger order bias in the questions where there was not a clear answer. We looked at each comparison between images and used a statistical test to see if one of the images in the comparison was chosen significantly more than the other (*i.e.* questions with a preferred answer). We then found the proportion of time the right hand image was picked for comparisons with and without a preferred answer. Figure 1 gives a summary of our results.

We found that the pictures where there was not a preferred answer had a much higher right bias. For instance, in the aesthetic question set, the p-value for the comparisons with a preferred answer is 0.0114, whereas the p-value for the comparisons that do not have a preferred answer is 0.000111, which is much lower. In the similarity question set there is a slight left bias (not statistically significant) for the pictures that have a preferred answer and a fairly strong right bias (p-value=0.0021) for pictures that do not have a preferred answer.

Since the right bias is much stronger in the aesthetic question set, we also speculate that, in general, this type of question does not have as strongly preferred answers. Indeed, it is more subjective and less clear which image is more aesthetically pleasing than which image looks more like an original image.

We conclude that the presence of an order bias indicates there were many questions that did not have a preferred answer. However, as noted earlier, our equal placement of images within the survey prevents any skewing in the data.

5.4. Basis Preference Ordering

When analyzing which bases were preferred aesthetically we looked both at each drawing individually and at all drawings collectively.

Figure 4 shows the overall aesthetic preference across all drawings. We measure preference as the total number of votes over the total number of appearances of each basis. The bars with matching

	Aesthetics Question Set			Similarity to Original Question Set		
	Total	Clear preference	Unclear Preference	Total	Clear preference	Unclear Preference
Prop. RS picked	0.531	0.520	0.543	0.511	0.488	0.545
z score	5.12	2.27	3.69	1.574	-1.267	2.8
p-value	1.53e-07	0.0114	1.11e-4	0.0577	0.8974	0.00211

Table 1: *Order bias tends to occur when the p-value is large and the bases are indistinguishable. There tends to be more bias toward the right side than the left.*

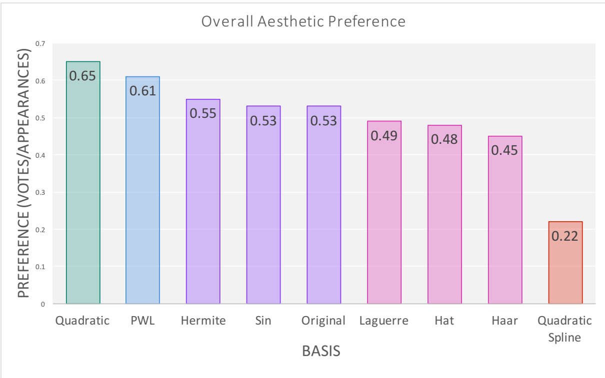

Figure 4: *Aesthetic preference for each of the basis types averaged across all pictures. Preference is measured as the number of votes over the number of appearances. Bars of the same color are not aesthetically different based on statistical significance.*

color are not preferred over each other with any statistical significance. This significance was determined using two sampled t-tests.

These combined results reveal that, in general, the quadratic is the most aesthetically pleasing basis and PWL is the second most aesthetically pleasing. Hermite, sinusoid, and the original drawing tend to have the same aesthetic quality, and so do Laguerre, Hat and Haar wavelets. The quadratic spline is the clear loser when it comes to aesthetics. We hypothesize that quadratic spline performs poorly because it does not allow for differential discontinuities which are necessary to create sharp corners in a curve.

This overall ordering describes a general trend in aesthetics, but the actual aesthetics vary from image to image, sometimes significantly so. For example, the PWL basis tends to do well and is the second or third most preferred basis for most drawings, but in the Seattle drawing, PWL is the third least preferred basis. Figure 5 shows the ordering for each of the drawings.

5.5. Similarity to Original

We analyzed the similarity of each basis to the original (as determined by our study participants) both across all drawings and for each drawing individually.

Figure 6 shows the similarity of each basis to the original sketch across all drawings. We defined similarity to the original to be the number of votes for a basis over the number of times it appears in the similarity question. That is, the similarity to the original is based on how many times a survey participant stated the basis was more similar to the original drawing than a drawing reconstructed from a different basis.

Looking at similarity across all drawings we see that the original is the most similar to itself by a significant margin, as one would expect. The sinusoid is the second most similar to original. Laguerre is the third, although not significantly more so than the Hat basis. Hat, Quadratic, Haar, and Hermite have a preference close to 0.5 suggesting that these all appear to be equally different from the original. This does not necessarily mean that they are indistinguishable from one another. These bases could all look different, but their relative difference from the original could all be roughly equivalent. The piece-wise linear basis is noticeably different from the original and all other bases, and finally, the quadratic spline is the most different-looking basis of all (again because the quadratic spline does not handle sharp corners well).

Figure 7 displays the similarity to original broken down by drawing. As expected, the original is selected as the most similar to itself in almost every drawing. In the more detailed pictures, the sinusoid basis is indistinguishable from the original (the Seattle, mountain, and tree drawings).

5.6. Differences Between Drawings

It is worth noting that some of the drawings seemed to have significant aesthetic differences when reconstructed using different basis functions and others did not.

The most complex drawings appeared to have little variation in aesthetic rankings. For example, the mountain drawing had no significantly difference aesthetics across all of the bases (except for the quadratic spline variation). The more complex drawings may have so many details that it is hard to pick out changes in aesthetic quality due to individual curve shapes.

The most simple drawing, the hat, also had little variation in aesthetics. The curves within this very simple drawing may not be significantly altered by different bases, and therefore the overall aesthetics do not change. This is corroborated by the fact that large groups of bases variations of the hat were perceived to be equally similar to the original hat picture.

5.7. Discussion of Survey Results

In sum, we can conclude that the basis function that is used to reconstruct the curve does, indeed, have an effect on the perceived aesthetics of the drawing. Part of this effect may simply be smoothing; as can be seen in Figure **??** where the original curves in the

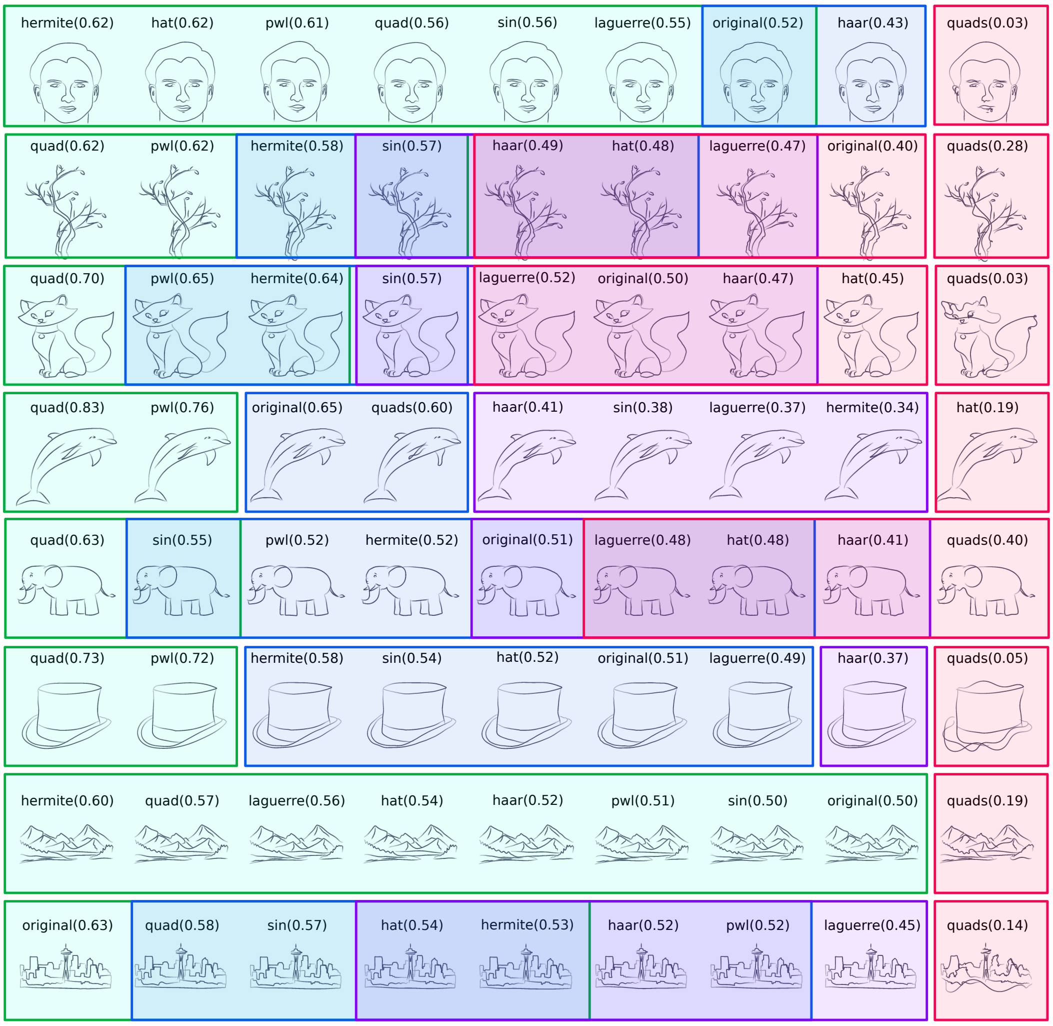

Figure 5: *Aesthetic ordering per drawing along with the the ratio of votes to appearances. Basis functions within the same colored box do not show a statistically significant difference in score.*

drawings contain a great deal of local curvature changes and peaks. However, although many of the basis functions eliminated these local curvature changes, there were still differences in perceived aesthetics, which indicates that using a specific basis does lead to perceivable differences.

The quadratic curvature basis generally produces drawings with the most aesthetically pleasing qualities and the piece-wise linear basis the second most. Further, we found that it varied from drawing to drawing how similar the quadratic basis appeared to the original drawing, and overall, the quadratic was just as similar to the original drawing as several other bases. This suggests that the aesthetic quality of the quadratic basis may be detected at a more subconscious level. Piece-wise linear, on the other hand, looked significantly more different from the original than the other bases. Participants could tell that piece-wise linear looked different and in general thought the different look was more aesthetically pleasing.

6. Curve Fitting

We now define our curve fitting algorithm. Given a sequence of 2D points which define a curve, the algorithm determines how many segments, where to place the boundaries (splits) between the segments, and then adjusts the free parameters of the bases in order to best-fit the original set of points in an L^2 sense.

We perform the fit in multiple stages, first fitting in curvature space (linear) then in geometry space (non-linear optimization). We use a windowing approach to speed up the non-linear optimization. We perform the non-linear optimization in geometry space because, in the end, we care about reconstructing the geometry, not the curve's curvature values.

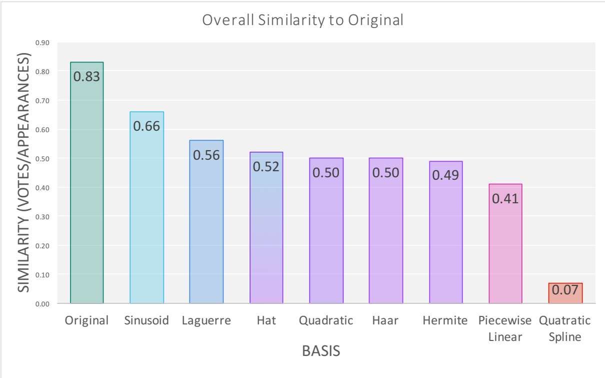

Figure 6: *The overall similarity of each basis reconstruction to the original drawing. This is the average across all drawings used in the survey. Bases with the same colored bars are not significantly different (determined by a two sample t-test). Note that Laguerre is not significantly greater than Hat, but it is significantly more similar to the original than quadratic.*

6.1. Calculating Curvature

We use the discrete calculation outlined in [MS09] ([KK06] presents an alternative method), which measures the change in angle $\Delta\theta$ between two adjacent line segments divided by the arclength Δs of one of those segments,

$$\kappa = \frac{\Delta\theta}{\Delta s}. \quad (1)$$

The curvature values for the first and last points of the curve are undefined, so we set them to the values of the adjacent points.

6.2. Fitting to the Curvature

Once the curvature is calculated we break the curve up into segments, each of which can be approximated with a simple basis function (see Section 6.3). For our piecewise bases this means each segment can be represented by a piecewise linear function or a quadratic polynomial; for our multi-basis, this means the segment can be represented by a sum of weighted basis functions. Our task is to set the coefficients of the polynomials (piecewise bases) or the weights (multi-basis) so that the resulting function approximates the calculated curvature.

Each segment has a start and end split point. This creates a unique assignment to each segment of one (or more) data points. We fit all segments at once using a least-squares approach.

We formulate the fit as $Ax = b$, where x is either the weights (multi-bases), the y values of the piecewise join points, the control points of the spline (quadratic spline), or the coefficients of the quadratic polynomials (quadratic). Each row of A corresponds to one time, curvature pair (t_j, κ_j). For the multi-basis, the row sets $\sum_i b_i(t_j) = \kappa_j$, where t_j is the arc-length distance along the curve of the point j.

For the multi-basis and quadratic with positional constraints, we use a modified version of the standard least-squares approach that includes a second set of (underconstrained) equality constraints, $Qx = c$ [HH81]. This second set of constraints is used to ensure that the end point of one segment is the same as the start of the second $p_i(t) = p_{i+1}(t)$.

6.3. Finding Split Points

Our goal with creating the initial set of split points is to keep the number of segments small while still limiting the total amount of curvature *change* within each segment. We make two observations: First, noise in the curvature (small wiggles) can largely be ignored, since that primarily arises from noise in the geometry that we are trying to smooth out. Second, corners in the original curve tend to produce a "spike" in the curvature. To accurately capture this spike it is best to create split points at the inflection points as well as the peak of the curvature; otherwise, the reconstruction will smooth out that corner.

We pre-process the sketch by arc-length sampling it and scaling it so that the curve fits in the unit square (normalizing the curvature values, since curvature depends on scale). We add split points at the geometric corners, the extrema, and to split up segments.

Geometric corners: Sharp corners in the original curve should be modeled as delta functions in the curvature plot. We explicitly find these corners using the Short Straw Algorithm [**?**] and add three split points (representing a delta function) which are each two sample points apart, centered on the corner.

Extrema: We identify extrema as points who's absolute curvature value is both greater than an absolute threshold ($|k| > 1$) and in the top 1/4 of absolute curvature values for the entire curve. Similar to the Short Straw algorithm, we remove the smaller of extrema that are too close together (within 0.05 of each other in absolute curvature value or closer than 4 sample points).

Linear fit: We recursively subdivide the remaining segments until they are sufficiently well-modeled with a linear segment or the resulting segment would have fewer than two data points in it. We define "sufficiently close" using a user-defined threshold d_m based on the overall range of curvature values.

$$d_m = \varepsilon |\max_i \kappa_i - \min_i \kappa_i| \quad (2)$$

We set $\varepsilon = 0.1$ for the examples in this paper.

This initial fit is now refined using non-linear optimization in *geometry* space, splitting segments further if necessary. We perform this as a sliding window for computational efficiency.

6.3.1. Windowing

The window consists of four segments (see Figure 9). The window that is being fitted (red) consists of three split points and the preceding segment. To the left of the window is the currently fixed part of the curve — the split points that have already been fitted (blue). To the right of the window are the remaining split points on the curve that have not yet been fitted (gray). Note that only the part of the curve that can change (red) is included in the optimizer error calculation.

After the segments corresponding to the area under the window

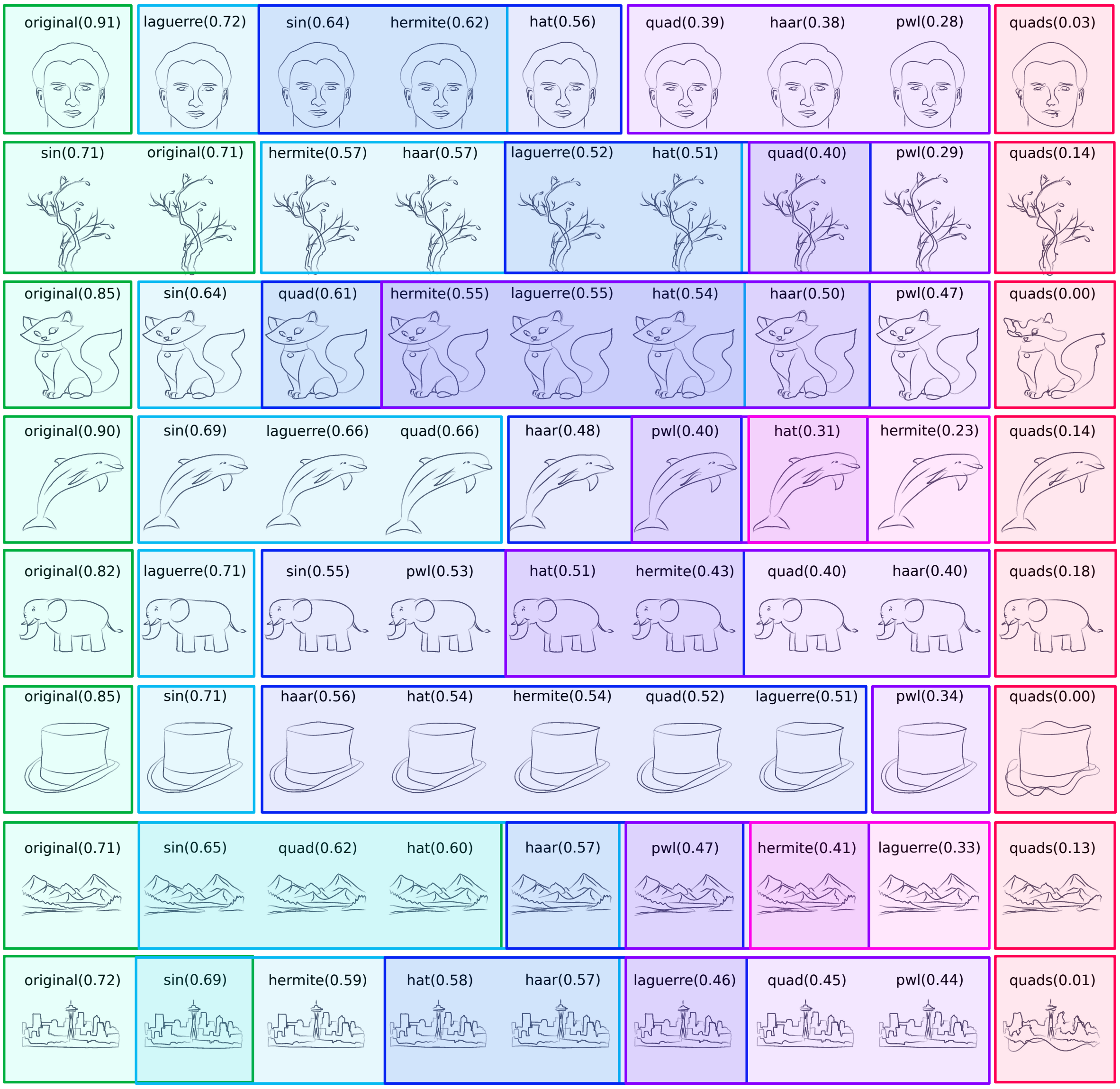

Figure 7: *Similarity to the original broken down by drawing. The number in parenthesis is our standard metric votes versus appearances which measures the number of times participants choose the drawing as being more similar to the original than another basis.*

have been optimized, and additional split points added if necessary, the window moves one split point toward the end of the curve. This is repeated until the window has reached the end and all split points become fixed points.

6.4. Non-Linear Optimization

The Nonlinear Optimizer (NLO)) uses a general-purpose optimizer to adjust the parameters of the curvature basis functions. We used MATLAB's `fminsearch}` with standard settings, which implements a version of Nelder-Mead's simplex method [LRWW98].

Error function: The curve is reconstructed using the discrete method in 3.2. This reconstructed curve is then compared to the original in geometry-space. The error is the sum of the distance between the reconstructed curve and the original data points, modified slightly to account for parameter slide (no backtracking) and the windowing.

6.5. Implementation Details

Reconstruction from curvature is unique up to a rotation and translation. In practice, we always start the ODE reconstruction by centering the curve over the origin with the tangent vector pointing in the positive x direction. The ODE reconstruction then starts in the middle of the curve and integrates out in both directions, in order

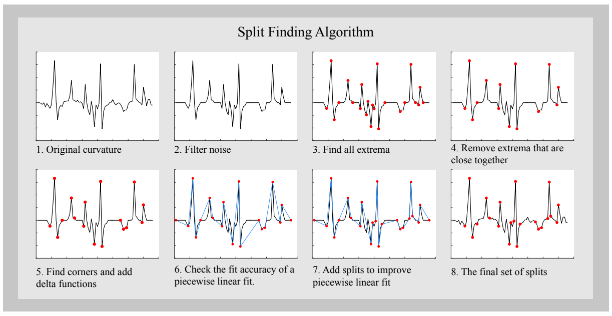

Figure 8: *Identifying good candidates for split points. Left: Identify points with high curvature, then filter out ones that are too close together (extrema). Right: Approximate each segment with a linear function and iteratively add splits at the points that are furthest from the linear fit.*

to minimize drift. We start the discrete reconstruction with the first point of the curve at the origin and the initial vector in the positive x direction.

We find an initial rotation and translation that take the reconstructed curve to the original's location by aligning the corresponding points and vectors. Since determining the corresponding x vector in the original curve can be problematic due to noise, we then apply a non-linear optimization procedure to solve for the best rotation and translation, this time matching the reconstructed curve's geometry to the original in a least-squares sense. This also addresses issues with global drift.

As a pre-processing step we resample the curve using arc-length sampling, filtering out duplicate points. This reduces error in the curvature calculation and provides one level of smoothing. For pen input we apply a Gaussian filter two times to reduce noise due to pen jitter.

6.6. Timing

There is a wide range in computation time involved for different drawings and different basis functions. At one extreme, the Seattle drawing takes close to an hour to fit the entire drawing using the Laguerre basis. At the other extreme, it only took 33 seconds to fit the dolphin picture using the piece-wise linear basis (all un-optimized MATLAB code). In interactive applications this could be amortized by fitting as the user draws the curve.

7. Concluding Remarks

We describe an algorithm for reconstructing 2D and 3D sketched curves using a variety of different bases in curvature space. Using a simple forced-choice test, we demonstrated that different curvature bases produce drawings that have a noticeable difference in aesthetics.

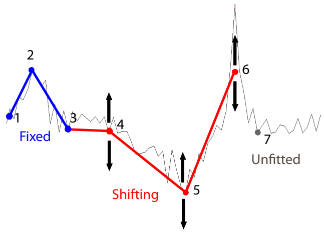

Figure 9: *Using a window to adjust four adjacent segments of the fitted curve. The left end point of the first segment is fixed.*

References

[BAFS94] Burchard H. G., Ayers J. A., Frey W. H., Sapidis N. S.: *1. Approximation with Aesthetic Constraints*. 1994, pp. 3–28. URL: http://locus.siam.org/doi/abs/10.1137/1.9781611971521.ch1, arXiv:http://locus.siam.org/doi/pdf/10.1137/1.9781611971521.ch1, doi:10.1137/1.9781611971521.ch1. 2

[BBS08] Bae S.-H., Balakrishnan R., Singh K.: Ilovesketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proceedings of the 21st annual ACM symposium on User interface software and technology* (New York, NY, USA, 2008), UIST '08, ACM, pp. 151–160. URL: http://doi.acm.org/10.1145/1449715.1449740, doi:10.1145/1449715.1449740. 1, 2

[BLP10] Baran I., Lehtinen J., PopoviĂĎÂĞ J.: Sketching clothoid splines using shortest paths. *Computer Graphics Forum 29*, 2 (2010), 655 – 664. URL: http://proxy.library.oregonstate.edu.ezproxy.proxy.library.oregonstate.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=aph&AN=50891843&site=ehost-live. 1, 2, 7

[FRSW87] Farin G., Rein G., Sapidis N., Worsey A. J.: Fairing cubic b-spline curves. *Comput. Aided Geom. Des. 4*, 1-2 (July 1987), 91–103. URL: http://dx.doi.org/10.1016/0167-8396(87)90027-6, doi:10.1016/0167-8396(87)90027-6. 1, 2

[GJ12] Grimm C., Joshi P.: Just drawit: a 3d sketching system. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling* (Aire-la-Ville, Switzerland, Switzerland, 2012), SBIM '12, Eurographics Association, pp. 121–130. URL: http://dl.acm.org/citation.cfm?id=2331067.2331084. 1

[HH81] Haskell K., Hanson R.: An algorithm for linear least squares problems with equality and nonnegativity constraints. *Mathematical Programming 21*, 1 (1981), 98–118. URL: http://dx.doi.org/10.1007/BF01584232, doi:10.1007/BF01584232. 7

[KK06] Kim D. H., Kim M.-J.: A curvature estimation for pen input segmentation in sketch-based modeling. *Computer-Aided Design 38*, 3 (2006), 238 – 248. URL: http://www.sciencedirect.com/science/article/pii/S0010448505001843, doi:http://dx.doi.org/10.1016/j.cad.2005.10.006. 7

[KNS] Kanaya I., Nakano Y., Sato K.: Simulated designer's eyes -classification of aesthetic surfaces. In *Proc. VSMM2003, 2003, pp 289-296. Computer-Aided Design and Applications*, pp. 457–464. 2

[LRWW98] Lagarias J. C., Reeds J. A., Wright M. H., Wright P. E.: Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM J. on Optimization 9*, 1 (May 1998), 112–147. URL: http://dx.doi.org/10.1137/S1052623496303470, doi:10.1137/S1052623496303470. 8

[MG16] Miura K. T., Gobithaasan R. U.: *Aesthetic Design with Log-Aesthetic Curves and Surfaces*. Springer Singapore, Singapore, 2016, pp. 107–119. URL: http://dx.doi.org/10.1007/978-981-10-1076-7_12, doi:10.1007/978-981-10-1076-7_12. 2

[MR07] Mullineux G., Robinson S. T.: Fairing point sets using curvature. *Comput. Aided Des. 39*, 1 (Jan. 2007), 27–34. URL:

http://dx.doi.org/10.1016/j.cad.2006.09.002, doi:10.1016/j.cad.2006.09.002. 2

[MS09] McCrae J., Singh K.: Sketch-based interfaces and modeling (sbim): Sketching piecewise clothoid curves. *Comput. Graph. 33*, 4 (Aug. 2009), 452–461. URL: http://dx.doi.org/10.1016/j.cag.2009.05.006, doi:10.1016/j.cag.2009.05.006. 1, 2, 7

[MS11] McCrae J., Singh K.: Neatening sketched strokes using piecewise french curves. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling* (New York, NY, USA, 2011), SBIM '11, ACM, pp. 141–148. URL: http://doi.acm.org/10.1145/2021164.2021190, doi:10.1145/2021164.2021190. 1, 2

[OEYK12] Orbay G., Ersin YüMer M., Kara L. B.: Sketch-based aesthetic product form exploration from existing images using piecewise clothoid curves. *J. Vis. Lang. Comput. 23*, 6 (Dec. 2012), 327–339. URL: http://dx.doi.org/10.1016/j.jvlc.2012.07.001, doi:10.1016/j.jvlc.2012.07.001. 1, 2

[Pav83] Pavlidis T.: Curve fitting with conic splines. *ACM Trans. Graph. 2*, 1 (Jan. 1983), 1–31. URL: http://doi.acm.org/10.1145/357314.357315, doi:10.1145/357314.357315. 2

[Sin99] Singh K.: Interactive curve design using digital french curves. In *Proceedings of the 1999 Symposium on Interactive 3D Graphics* (New York, NY, USA, 1999), I3D '99, ACM, pp. 23–30. URL: http://doi.acm.org/10.1145/300523.300525, doi:10.1145/300523.300525. 2

[SRG15] Shaheen S., Rockwood A., Ghanem B.: Sar: Stroke authorship recognition. *Computer Graphics Forum* (2015), n/a–n/a. URL: http://dx.doi.org/10.1111/cgf.12733, doi:10.1111/cgf.12733. 2

[TSB11] Thiel Y., Singh K., Balakrishnan R.: Elasticurves: Exploiting stroke dynamics and inertia for the real-time neatening of sketched 2d curves. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2011), UIST '11, ACM, pp. 383–392. URL: http://doi.acm.org/10.1145/2047196.2047246, doi:10.1145/2047196.2047246. 2

[WKSM12] Wyvill B., Kry P. G., Seidel R., Mould D.: Determining an aesthetic inscribed curve. In *Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging* (Aire-la-Ville, Switzerland, Switzerland, 2012), CAe '12, Eurographics Association, pp. 63–70. URL: http://dl.acm.org/citation.cfm?id=2328888.2328899. 1, 2

[WPL06] Wang W., Pottmann H., Liu Y.: Fitting b-spline curves to point clouds by curvature-based squared distance minimization. *ACM Trans. Graph. 25*, 2 (Apr. 2006), 214–238. URL: http://doi.acm.org/10.1145/1138450.1138453, doi:10.1145/1138450.1138453. 2

[XM09] Xu L., Mould D.: Magnetic Curves: Curvature-Controlled Aesthetic Curves Using Magnetic Fields. In *Computational Aesthetics in Graphics, Visualization, and Imaging* (2009), Deussen O., Hall P., (Eds.), The Eurographics Association. doi:10.2312/COMPAESTH/COMPAESTH09/001-008. 2

[YS06] Yoshida N., Saito T.: Interactive aesthetic curve segments. *The Visual Computer 22*, 9 (2006), 896–905. URL: http://dx.doi.org/10.1007/s00371-006-0076-5, doi:10.1007/s00371-006-0076-5. 2

[Zit13] Zitnick C. L.: Handwriting beautification using token means. *ACM Trans. Graph. 32*, 4 (July 2013), 53:1–53:8. URL: http://doi.acm.org/10.1145/2461912.2461985, doi:10.1145/2461912.2461985. 2