Automatic Pen Stroke Guidance by Example

Justin Solomon, Charlton Soesanto, and Haithem Turki

CS 221: Introduction to Artificial Intelligence

1 Aim

Using an artist's initial strokes as training data, we wish to train an automatic pen tool that places and shapes strokes with minimal guidance.
Such a tool could be used to add detail to rough sketches or texture regions that would otherwise be frustrating to fill in.

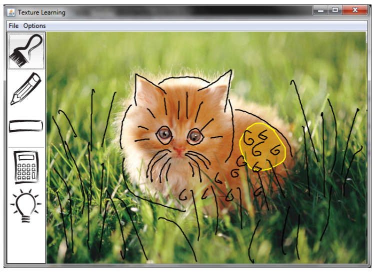


Figure 1: User interface; strokes are drawn in black, and the region of interest is a yellow polygon. The user clicks the calculator to run learning, can draw new regions of interest, and subsequently clicks the light bulb to use the learned model.

3 Results

Using the mixture of Gaussians Model, we were able to successfully select a starting point for the curve. Furthermore, choosing an orientation and length for the curve worked well. Curve shape was effective in some cases but future work is needed to choose the best possible regression technique.

Figure 5: Examples of algorithm output; black curves are examples and green curves are automatically produced by the model. These figures show the advantages and disadvantages of the approach at hand. Linear regression only allows for fairly simplistic spiral curves.

Even so, they capture basic curvature relationships between curvature and arc length, and they have reasonable shape. Placement works fairly well; for instance, the green spiral curves all start at the center of the image, and their initial directions are distributed evenly.

4 Discussion/Future Work

Although final results are mixed, they indicate that a refined version of our method shows promise for the implementation of a "smart" curve placement system.

This framework allows for a higher degree of artist control than previous approaches.

Goals for future research:

1. Find optimal combination of heuristics and regression techniques to compute new curves reliably
2. Find a model of curve density to determine when to start and stop drawing curves in a given region
3. Find the interface that is most natural for artists making use of such a tool
4. Integrate with existing "paint" software as a useful brush to be used alongside more traditional pen tools

2 Approach

Before we could learn from our drawn our own lines, we established a geometry in which to draw, define, and predict curves with the following:

- The **Fundamental Theorem of Plane Curves** (two curves γ1 and γ2 parameterized by arc length represent the same shape if and only if they have the same starting point, orientation, and curvature functions κ) shows that once we decide where to place a pen curve, we can decide on its curvature function to reconstruct curve shape.
- Curvature is also a local differential property, which means that the decision of a curvature value does not need any global curve information to maintain smoothness.
- Also, it is easily discretized on a curve composed of short line segments as the ratio of angle defect to segment length, which we do as shown in Figure 2.

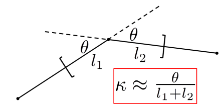
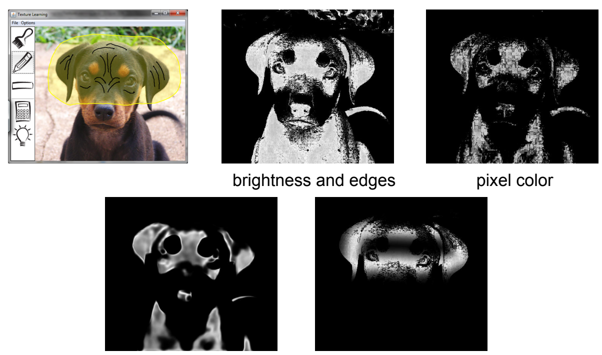


Figure 2: Set-up for computing discrete curvature at the join between two segments.

After we determined how to represent curves, we chose to model the decision-making process of drawing curves based on user-provided input in two stages, in analogy with how an artist might do the same:
1) the computer (or artist) picks an initial pen position and orientation.
2) the computer (or artist) guides the pen along the paper.

Choosing a starting point To choose a starting point for a curve, at each pixel we evaluate the curve placement heuristics to find a probability density value p_{place} representing the likelihood that the pixel is a curve starting point. We then sample a single (x, y) pair from the set of all pixels with probability proportional to p_{place}.

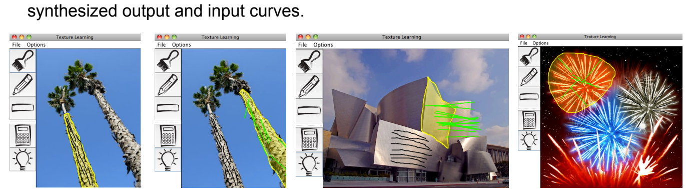
Figure 3 shows curve placement probabilities for the example curves in in the first image using the heuristics underneath the other images.The probabilities p_{place} are scaled to $[0, 1]$ and shown at each pixel.



brightness and edges | pixel color | blurred pixel | color brightness and distance to region of interest boundary

Choosing a length and initial direction: For curve length heuristics $H_1, \ldots, H_n$, our learning algorithm provided a mixture of Gaussians expression for the joint distribution $P(L,H_1,\ldots,H_n)$. To decide on a particular length, we evaluate $H_1, \ldots, H_n$ at (x, y) and then conditionalize the distribution to find the single-valued distribution $P(L|H_1,\ldots,H_n)$; this conditionalization can be carried out analytically using the common formula for the conditional distribution of a Gaussian. To choose a particular curve length at (x, y), we simply sample this conditional distribution. An identical process occurs for deciding the initial curve orientation.

Below is **Figure 4**, where choice of curve placement, direction, and length are rendered using line segments rather than pen curves; (a) and (b) show a training data and synthesized data pair with marked regions of interest, while (c) and (d) contain synthesized output and input curves.



(a) (b) (c) (d)

Curve construction: We construct the curve incrementally, adding vertices one at a time until we reach the length. For each vertex, we simply compute the curve guidance heuristics for the curve built so far and evaluate a learned function to estimate curvature. The turning angle representation of curvature allows us to determine the location of the next sample point, and the process is repeated.