Rendering Handwritten Characters

Sara Su, Chenyu Wu[1], Ying-Qing Xu[2], Heung-Yeung Shum[3]

Introduction: As users of pen-based computers rely increasingly on freehand interaction, they become ever more sensitive to the on-screen appearance of their digital handwriting. We describe a method for improving the rasterized appearance of handwritten characters by borrowing strategies from the field of digital typography.

Approach: The majority of fonts in use today represent glyphs as outlines formed by knots and splines. At rasterization time, these outlines are scaled to the desired size on the pixel grid and each pixel lying within the outline is turned on. *Hinting* significantly improves the readbility of text by giving the typographer finer control over the final appearance of each glyph beyond this simple filling algorithm. With these grid-fitting instructions, the typographer can specify rendering constraints (between knots of a glyph or between a knot and a gridline) to prevent much of the aliasing that would otherwise occur at small point sizes [2]. Though it is a tedious manual process, hinting is essential for clear rendering of glyphs. Stroke width uniformity, stroke continuity, glyph spacing: all are controlled by hinting.

Hypothesizing that such grid-fitting instructions can similarly improve the appearance of digital handwriting, we have developed a system for automatically hinting handwritten characters by considering features of each glyph against knowledge of previously-hinted TrueType fonts [3]. This approach to transferring hints between template fonts and target characters is motivated by earlier work on shape-matching for example-based hinting [1, 4].

We use an aggressive matching strategy to find correspondences between templates and inputs, considering both global and local glyph features. We first calculate global correspondences between an input glyph and the same glyph from the template set and then calculate local correspondences through comparisons to analogous curves of other template glyphs. This hybrid approach allows us to find matches even for input/template glyph pairs that are topologically dissimilar.

After knot correspondences have been found, translation of hint instructions from template to input is relatively straightforward. In addition to glyph-specific hints, the description of a TrueType font contains references to global data in the control value table (CVT) to unify structural elements across glyphs. It could be argued that the CVT, whose apparent purpose is to impose uniformity, is not useful when dealing with the irregularities of handwriting. However, though the constraints are much less rigid than those of Roman typefaces, there still exists a degree of uniformity across most handwritten characters. Recognizing that stroke-width statistics, among the data stored in the CVT, are key characteristics of an individual's handwriting style, we attempt to preserve as much CVT data as possible in translation. We transfer CVT entries from the template, re-estimate their average values based on measurements at input knots, identify clusters of similar references to table entries, and create new entries when needed. This clustering and branching is a key advantage of our approach as it retains much of the CVT information that other hint-transfer methods discard.

Progress: We have implemented a prototype auto-hinting system for handwriting [3]. A manually-hinted TrueType font was used as the template, and input data was collected from volunteers writing on tablet computers. The results are encouraging. At sizes of 24pt and smaller, the auto-hinted handwriting exhibits fewer aliasing artifacts than unhinted text, and the appearance of individual characters improves with the combination of hinting and a simple *dropout control* mechanism. When part of a stroke is thinner than one pixel, the resulting hole or 'drop' in the raster image can be disruptive to perception of the character. Dropout control, incorporated in most auto-hinters, prevents these artifacts by detecting drops and inserting an extra pixel at the site.

Future: We have discussed example-based methods of improving the appearance of handwriting in the context of TrueType font hinting. The availability of many high-quality hinted TrueType fonts makes them

[1] Tsinghua University, China
[2] Microsoft Research Asia, China
[3] Microsoft Research Asia, China

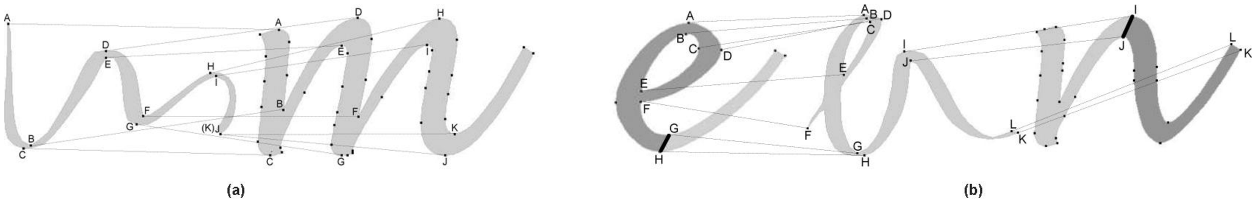

Figure 1: Results of the (a) global and (b) local correspondence searches for two input characters.

ideal templates. However, we would like to explore these methods in the more general context of intelligent rendering of handwriting and consider what additional information can be encoded in templates.

Research Support: This work began while the first and second authors were interns at Microsoft Research Asia.

References:

[1] Roger D. Hersch, Claude Bétrisey, "Model-Based Matching and Hinting of Fonts," In *Proceedings of SIGGRAPH 91*, 25(4), pp. 71-80, 1991.

[2] Beat Stamm, "The Raster Tragedy at Low Resolution," Microsoft Corporation, Excerpts from a presentation to the Microsoft Typography Group, http://www.microsoft.com/typography/tools/trtalr.htm, 1997.

[3] Sara Su, Chenyu Wu, Ying-Qing Xu, and Heung-Yeung Shum, "Rendering Handwritten Characters," Under review.

[4] Douglas E. Zongker, Geraldine Wade, and David H. Salesin, "Example-Based Hinting of TrueType Fonts," In *Proceedings of ACM SIGGRAPH 2000*, pp. 411-416, 2000.