

Excerpt from technical report on JustDrawIt: a 3D sketching system

sbim paper 1004

March 9, 2012

The following is an excerpt from a technical report that will accompany the “JustDrawIt: a 3D sketching system” paper. Here we provide implementation details for subsection 4.1 and general implementation details for Section 4 (which will be a new subsection 4.5) from the paper. In the tech. report, these implementation details will be in addition to the text in the paper.

Addendum to Sec. 4.1: Preprocessing for back and forth scratching

More specifically, we define an averaging function that takes in a set of sections and a point. The point is projected to each of the sections, and only those points that project to the interior of the section are kept. The section points are weighted by their distance to the input point and averaged.

To initially extend a section, construct a Hermite curve that joins the end point to the average curve. One end of the Hermite curve is the end point and tangent of the section end. The other end is found by taking a point d distance along the tangent and projecting it using the averaging function above (see Figure 8, middle). If the tangent extends past the end of the average curve, just use the point along the tangent. The distance d is the selection distance size. The tangent at the second point is the average of the tangents at the projection point. Both tangents are scaled to be length d . Sample the Hermite curve with 6 points and add those points to the section.

To blend a section with the average curve, we move each point on the section towards the projection point. The middle 50% of the curve (by arc length) is moved 0.25 towards the projected point, the ends are moved by

0.75 (the movement amount is linearly interpolated). Each section is also smoothed before averaging. We apply the projection operator 3 times before extracting the final curve.

This fold-over processing is used in one other place – if the current stroke overlaps more than 90% with the last *stroke* (not curve). If this is the case, we walk back in the list of strokes until we have a complete list of all of the overlapping strokes. Then we merge all of the strokes into a single, non-folding stroke and then proceed as before (see far right of Figure 8). This is preferable to treating these strokes as overlaps because the new stroke should not over-ride the last one, but should be blended with it.

Addendum to Sec. 4:

4.5 Implementation details and parameters

Recall that all curves and strokes are stored simply as lists of 3D or 2D points. Because different devices have different amounts of noise and sampling rates, we need some mechanism for controlling the quality of the samples and their frequency. The user supplies two parameters (set via sliders): The selection distance d as a number of pixels, and a smoothing rate N . As the strokes are processed, we perform the following three actions: 1) inserting samples to ensure the sampling rate is at least 10 points per screen distance d (in case the pen input “skips”, 2) smoothing some amount based on both N and the quality of the data, and 3) re-sampling to ensure even-spacing in 3D.

Smoothing is applied in three places: 1) To the ends of the stroke before computing merge and join information, 2) to the projected depth values while promoting the stroke to 3D, and 3) to a region around the joins. The system applies N rounds of smoothing to the stroke before processing, $2 + N$ rounds of smoothing for the depth, and $1 + N + 40q$, where q is the merge or join score from Section 4.3 for the join (q is typically in the range 0.01 to 0.1). One round of smoothing moves the point 0.2 towards the average of the neighbors. For the depth smoothing, we construct the 3D points, average, then re-project the point to the ray, rather than filtering the depth values directly.

We re-sample the strokes after promoting them to 3D. d is a screen-based sampling rate; by projecting a segment of length d onto a plane perpendicular to the view, centered in the middle of the curves, we can convert d to a 3D absolute distance d_3 . When re-sampling, we ensure they are spaced $0.1d_3$

apart. Since d reflects, in some sense, how “accurate” the 2D device is, and how close is close enough, it is a good measure for how accurately to sample the data.

To speed up the stroke to curve calculations we keep a 3D bounding box for each curve. If the stroke lies a distance d outside of the projected bounding box then we ignore that curve.