Creating B-Spline Control Points

We initialize B-Spline control points along the corners of a simplified Voronoi diagram. We observe that the placement of these control points in a 2 x 2 node block solely depends on the presence of a diagonal connection between the block's nodes. This allows us to compute control-point positions in parallel by looking at each cell separately. We present a data structure that allows random access lookup to control points, based on the position of their generating node blocks in the planar graph. Based on the connectivity of each cell's nodes to surrounding nodes, we can identify neighboring control points lying on the same spline and store them along with each control point's position. In the same processing step, we are also able to identify control points that lie on intentionally sharp corners and duplicate control points that lie on T-junctions.

Optimizing B-Splines

The control points defining the B-splines still suffer from staircasing artifacts, which we mitigate by shifting each control point to reduce its corresponding curve segment's curvature.

Algorithm Overview

The general idea behind the algorithm is to create a resolution-independent, smooth representation of a given low-resolution pixel art image. The result should have sharp contours between regions of strongly dissimilar colors and smooth shading transitions between similarly colored regions. We show how to implement each of the steps of the original algorithm in a highly parallel fashion, exploiting modern GPUs.

Parallelization

In order to allow parallel processing, we split the algorithm up in a sequence of multiple parallel stages. Intermediate results are stored in buffers, allowing subsequent stages to continue processing.

Contact: falichs@gmail.com, kopp@microsoft.com, wimmer@cg.tuwien.ac.at