A better distance transform for distance map contour textures

The distance map contour texturing method published by Green in 2007 has found widespread use in the gaming industry. However, the distance transformation method used by Green for creating the distance map was an exhaustive search method, suffering from extreme inefficiency. In this context, that is not a big problem in itself and, if need be, more efficient methods can be found in literature and in free software libraries for image processing. However, one clear disadvantage remains: traditional distance transform methods accept only binary images as input (black and white only, no grayscale), and therefore a very high resolution contour image is required to create an accurate enough distance map for Green's method.
The new distance transform method presented here uses an anti-aliased alpha mask to generate a distance map of much higher precision and accuracy than what is possible from a binary image. The method is an improvement over the state of the art in image processing an pattern recognition literature, and has been submitted for publication in Pattern Recognition Letters (submission date: June 2009).

Anti-aliased distance transform

Traditional distance transform methods choose to reduce the problem to a binary input image, where each pixel belongs either entirely to the foreground or entirely to the background. This is not a good model of a real contour with arbitrary position and slope, where the position of the edge can be anywhere within a pixel, and pixels can straddle the edge with an arbitrary area coverage. By using a properly anti-aliased input image, where the grayscale value of each pixel encodes the relative area coverage of the object within that pixel, we can do a lot better. Both the accuracy and the precision of the distance map can be improved, with faster processing and less input data.
Our contribution is a modified measure of distance, where the grayscale value of an edge pixel is taken into account. This modified distance measure can be used for many different distance transformation algorithms, among them the classic SSED8 algorithm presented here. It is fast, well tested, reasonably simple and well suited to this modification. The method is not described in detail in this document. A formal and detailed research article is currently under review for publication.

Source code and documentation

Full source code in C is available on the link below, with a MEX function wrapper to use the modified transform as a plug-in replacement for the built-in distance transform method BWDIST in Matlab and the free alternative GNU Octave. The core algorithm is a separate C function which can be used in a stand-alone fashion, without the MEX wrapper. The software is released under the terms of the GNU General Public License, version 3 (see http://www.gnu.org/licenses/gpl.html).

http://www.itn.liu.se/~stegu/aadist/ (Link to this document and the source code)

To use the code from within Matlab or Octave, move to the directory where the files ”edtaa3.m”, ”edtaa3.c” and ”edtaa3func.c” are located and type ”mex edtaa3.c”. For help, type ”help edtaa3”. To use the C function in ”edtaa3func.c”outside of the Matlab or Octave environment, you need to familiarize yourself somewhat with the source code. If you need more help, limited support is available through a forum on the adress below. Feel free to use direct e-mail communication with me if you want to, but for support questions of general interest, please use the forum.

http://contourtextures.wikidot.com/ (Wiki site with a forum for support and feedback)

Stefan Gustavson (stegu@itn.liu.se) Linköping University 2009