Guide to free software for distance map contour textures

Stefan Gustavson 2009-10-06 (stegu@itn.liu.se)

In addition to the source code and data included in this package, you need some additional free software from other sources.

Texture creation

To create textures, you can use Matlab from MathWorks (http://www.mathworks.com) if you already have it. If not, the free Matlab-compatible GNU Octave works fine: http://www.gnu.org/software/octave/

Installation of Octave is quick and easy:

MacOS X and Windows: Complete Octave installation packages can be found here: http://octave.sourceforge.net/

Linux: For Debian-based Linux distributions, simply install the two packages "octave" and "octave-image".

With Octave (or Matlab) up and running, you need to compile the add-on function "edtaa3" to compute the modified Euclidean distance transform. This is performed from within Octave (or Matlab) by issuing the command "mex edtaa3.c" while in the directory with the files "edtaa3.c" and "edtaa3func.c".

Windows: If you are using Octave, you need to install a C compiler and configure Octave to use it. (Matlab for Windows comes with its own compiler.) Instructions on how to configure a compiler is in the Octave documentation.

Linux: GNU gcc is probably already installed as part of your system setup. If not, install it. The Debian package is "gcc". Octave and Matlab will find gcc if it is installed.

MacOS X: You might have gcc installed already. If not, it can be installed from the Apple installation DVD.

When you have Octave or Matlab up and running and have succeded in compiling the MEX file, you need to issue a few commands to create a texture file. A sample sequence of commands is in "createdemotextures.m", along with extensive comments. You can run it directly by typing "createdemotextures" in Octave to create four texture files for the demo program using the example images provided, and you can edit the script to create textures from your own images. Instructions on how to use the main function "makedisttex.m" are available from within Octave or Matlab by typing "help makedisttex". You can also read the source code, even if you are not familiar with Matlab programming. It's not magic, it's fairly basic image processing.

If you encounter any problems with my scripts or the MEX file, tell me and I'll try to help. However, I can't help you install, configure or learn Octave or Matlab. There are many good resources for that elsewhere.

Demo program experiments

The demo program is compiled for Windows, Linux and MacOS X. If your GPU and its OpenGL driver support GLSL, you should be able to run it. The demo reports the frame rate every second and allows you to pan by dragging with the mouse, and zoom and rotate by pressing "shift" or "ctrl" while dragging. You have a choice of four textures and two different shaders, selected by pressing "1" through "4" and "F1" or "F2". The textures reside in 8-bit TGA files named "dist1.tga" to "dist4.tga", and the reference bitmaps are in 8-bit TGA files named "ref1.tga" to "ref4.tga".

To experiment with the demo program, you can do a lot with a plain text editor by making changes to the text files "fragment1.glsl" and "fragment2.glsl" and simply restarting the demo. GLSL shaders are compiled by the OpenGL driver at runtime. If you make mistakes, the GLSL compiler messages are printed to the console.

To recompile the demo yourself, you first need a compiler.

Linux: Install "gcc" if you don't already have it. You also need the GLFW framework. This is provided in the Debian package "glfw2-dev". This will possibly install a few other packages required for OpenGL development on your particular system.

MacOS X: Install gcc from the installation DVD if you don't have it already. For the additional required frame-work GLFW, you can either try to find a pre-built installation package, or you can compile and install the GLFW library file yourself. If you have gcc it's actually quite easy. Instructions are in the GLFW source download available at: http://glfw.sourceforge.net

Linux and MacOS X: Once you have a compiler and GLFW installed correctly, just type "make" in a shell and the demo should compile.

Windows: The development environment used to create the demo for Windows was the free software package "Dev-C++" from Bloodshed Software: http://www.bloodshed.net/dev/devcpp.html Download and run the installation package that includes the free MinGW compiler. The source code for our demo includes a project file for Dev-C++ for your convenience. You also need the GLFW OpenGL framework. A DevPak for Dev-C++ is available from various sources on the Internet, but make sure you get the latest version. The current version is 2.6, and a copy is here: http://www.itn.liu.se/~stegu/OpenGLquickstart/glfw-2.6-3fun.DevPak

Good luck, and have fun! If you encounter any problems, tell me and I might be able to help, but I make no promises. In particular, I can not help you in configuring your own system to perform OpenGL development, like installing a hardware driver, finding the development libraries and figuring out which extensions are supported. Such problems can often be solved in the developer forums at http://www.opengl.org.