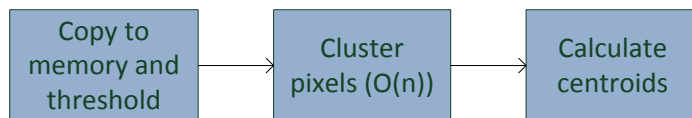


Digital Signal Processing – Thresholding, Clustering, Centroids

Submitted by Group 4 (Multi-touch table): Alix Krahn, Denis Lachance, Adam Thomson

This application note includes code for digital signal processing – particularly it thresholds data, clusters pixels, and calculates centroids of the pixels. Our code utilizes all three, but any of the three may be utilized independently.



To use the code:

- OpenGL must be installed. OpenGL is used to display the input from the camera alongside the processed image.
- Change the video feed being used. If running on Linux, then the video feed (if it is not the webcam on the laptop) is automatically detected, but this can be changed.
- Run make in the folder with the code.
- Run (type ./dsp)

The three processes are included in the functions:

```
threshold(unsigned char* pixels, int width, int height, int thresh)
```

```
get_centroids(unsigned char* pixels, int width, int height)
```

The code utilizes all three modules, and was used to detect touches on a multi-touch surface, which appeared as white blobs on a black background.

To see examples of how to use any function, check the unit tests for that function. Easier functions to implement such as clip_edges were tested on the fly and do not have unit tests assigned to them, but we can assure you that the harder functions to use have several unit tests assigned to them.

One of the trickier steps to running the code is installing the glfw3.h file. To do this you must install GLFW 3.0, which is typically downloaded and compiled. Here's a helpful guide:

<http://theanswers42.com/2013/12/08/building-glfw-3-on-ubuntu-13-10-or-how-i-learned-to-stop-worrying-and-love-the-cmake-file/>

In the main folder of this App Note, the ./dsp app runs as an OpenGL application which assigns a gesture to each number of finger presses:

- 1 finger is used to draw

- 2 fingers are used to change the brush color
- 3 fingers are used to change the brush size
- a large mass (such as a palm or fist) is used to clear the screen

Under the `mouse_control/` directory, we have included an example of how to move and control the mouse using C. This builds with `make` and runs with `./macro`. We used a PS2 controller to USB converter to get input so we could move the mouse. Comments indicate how this is done.

Under the `plan_b/` directory, we have redone the computer only dsp app using the DE2 accessed through a socket server. This builds with `make` and runs with `./dsp`. The drawing application is slightly different, since we didn't finish the DE2 – Computer interfacing, the app isn't finished yet.