**Nathan Sponberg – CS448 Pattern Recognition, Spring 2013 –**

**Edge Detection Project**

**Violet Images**

**Sobel:**

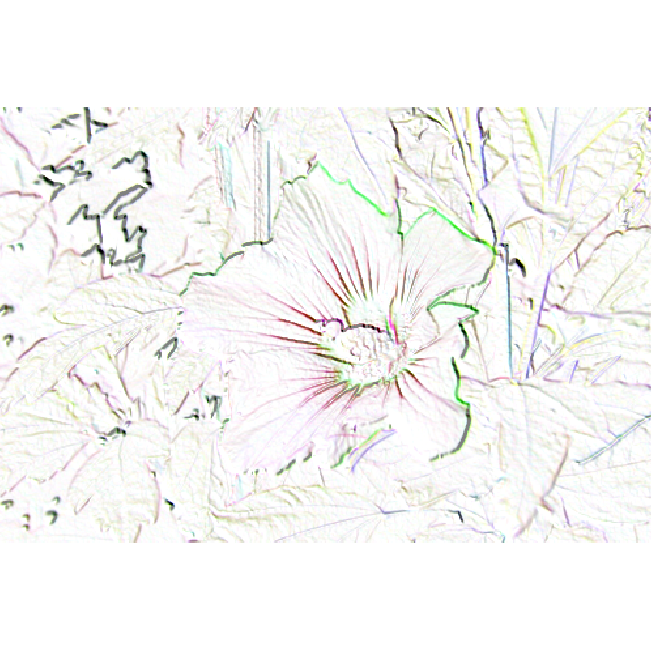


Figure 1 The image of the purple violet after the Sobel edge detection algorithm has been applied

to it. Note that this is the negative of the image output that is given by the Sobel algorithm.

**Canny:**

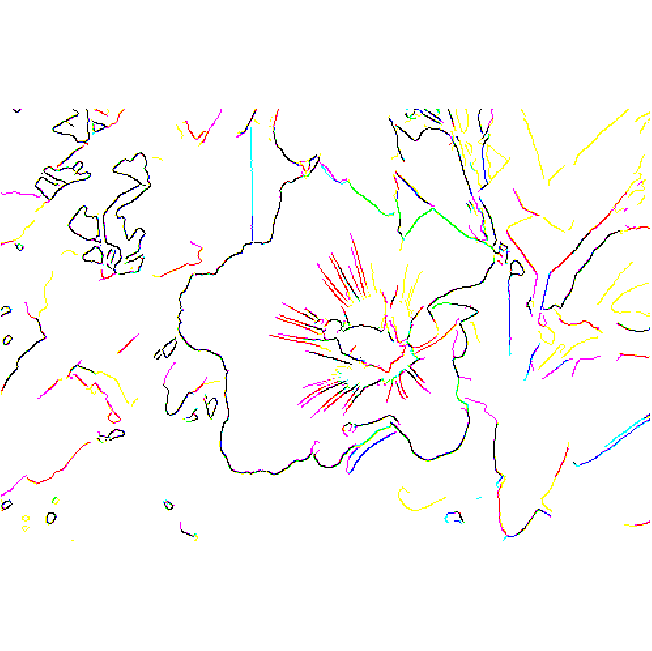


Figure 2 – The violet image with the canny edge detection algorithm applied to it. Observe how much

crisper the edges are.

Lenna:

**Laplacian**



Figure 3 – Image of Lenna with a unsmoothed laplacian edge detection algorithm applied to it.

**Smoothed Laplacian**



Figure 4 – Image of Lenna with a smoothed laplacian edge detection algorithm applied to it. The image was first smoothed using a gaussian convolution mask and then laplace edge detection was run on the smoothed image.

**Smoothed Laplacian with Boundary Increase**



Figure 5 – Image of Lenna that has been smoothed as in Fig. 4. The Laplacian Edge detection was then run on the smoothed image, however, in this case the lower boundary for pixel intensity was set to 100 as opposed to zero. Thus, any pixels that had an intensity value of 100 or less were replaced by an intensity of 0.

**Canny Edge Detection**



Figure 6 – Image of Lenna with the canny edge detection algorithm applied to it. Note that the image was not converted to grey scale before the algorithm was applied.

**Questions Section**

It is a little tricky to say which one of these algorithms worked better for this image. On the one hand, the canny algorithm produces much crisper edges that are only one or two pixels thick, the laplace algorithm creates messier edges. This is of course to be expected with any image. There are some edges on the top of the hat and the brim that canny captures more fully than laplace. However, there seems to be one problem that laplace deals with better than canny. Canny picks out the highlights on the nose, upper lip and top of the hat and puts two edges there, one the actual edge and the other the edge of the highlight. This is not very desirable and the laplace algorithm seems to do much better with this.

Why is this? The canny algorithm relies on the gradient of the pixel intensities which may make it more sensitive to these double edge highlights. The double boundary condition may also contribute to pick out two edges in this can this the change in intensity is so dramatic at the edge of a highlight. Also the fact that canny attempts to make thin edges based on the center of the intensity peak may be working against it in this case. The laplacian algorithm can make fatter edges and so will just create a thick edge where the highlight is (as the mask passes over this region most of the pixels in this highlight area will add up to have high intensity across the board and thus will be made into one edge). Over all though the canny is probably the one that I would choose to use. There are only a few areas were double edges appear and the rest of the edges seem to have a higher degree of fidelity that the laplacian image (and there is zero noise in the canny image). Some of the edges in the laplace image seem a bit washed out or are not as complete as in the canny. This is probably due to the lower threshold use on the laplace algorithm, which reduces noise but decreases sensitivity to less pronounced edges. The canny algorithm definitely outstrips the laplace in this respect. This is mostly likely due to the edge continuation threshold that it uses to see if the edge should be continued in the same direction of the edge gradient.