Computer Vision Assignment 1: Filtering

Robrecht Jurriaans (5887380), Taco Cohen (6394590)

November 11, 2012

1 Gaussian Filters

1.1 1D Gaussian Filter

We implemented the 1D Gaussian in `gaussian.m`. We made sure the kernel size is about $3 * \sigma$ and is always odd (using the `ceil` function).

Because the filter has a finite size, the sum of the filter values will not be one in a naive implementation. For this reason, we must normalize the kernel after calculating the values of the Gaussian at each kernel entry. To save computation, we used the following equality:

$$\begin{aligned} \frac{G_\sigma(x)}{\sum_{x'=-h}^{h} G_\sigma(x')} &= \frac{\frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{x^2}{2\sigma^2})}{\sum_{x'=-h}^{h} \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{x'^2}{2\sigma^2})} \\ &= \frac{\exp(-\frac{x^2}{2\sigma^2})}{\sum_{x'=-h}^{h} \exp(-\frac{x'^2}{2\sigma^2})} \end{aligned}$$

That is, we leave out the normalization of each index, because it falls out in the normalization of the entire kernel anyway.

1.2 Convolving an image with a 2D Gaussian

We implemented this the obvious way, relying on `conv2` to convolve the image first with the 1D x-direction filter, and then with the 1D y-direction filter.

1.3 Comparing with Matlab's Gaussian Filter

Thanks to the inherent separability of Gaussians, the implementation that convolves on one dimension at a time is slightly faster than the normal 2D implementation. We expect this difference to increase on larger images, as the difference is one of $O(N^2)$ versus $O(N)$. We note that there is a slight difference, on the order of 10^{-17} sum of squared differences, between the resulting images. This is likely due to rounding errors.

1.4 Gaussian Derivative

We implemented the Gaussian derivative, see `GaussianDer.m`.

Figure 1: Original Matlab Filter

Figure 2: Filter based on separation

1.5 Gradient Magnitude and Orientation

We implemented the function `gradmag`. The result is shown in the figures below.

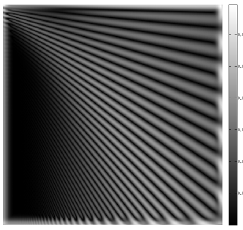

Figure 3: Magnitude image for $\sigma = 5$

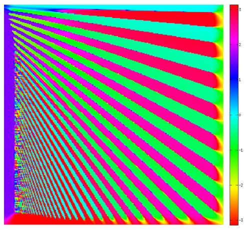

Figure 4: Orientation image for $\sigma = 5$

Figure 5: Magnitude image for $\sigma = 5$

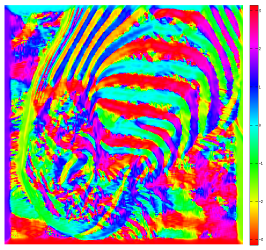

Figure 6: Orientation image for $\sigma = 5$

1.5.1 Quiver before my magnitude

See the quiver plots below.

1.5.2 Magnitude and orientation for different σ

Running `master.m` will result in gradient and orientation plots for various values of σ. It is clear to see that for larger values of σ, only larger structures are retained. The gradient is no longer affected by small details when σ is large. There is an obvious tradeoff here between informativeness on small details and sensitivity to noise. Likewise for the orientation plot.

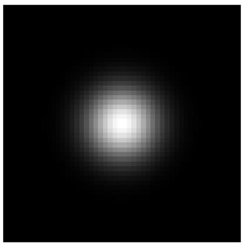

Figure 7: Original image

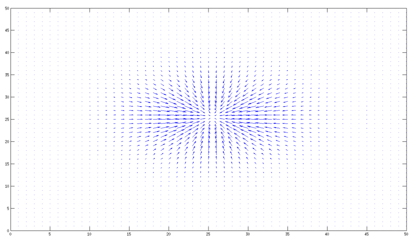

Figure 8: Gradient image for $\sigma = 1$

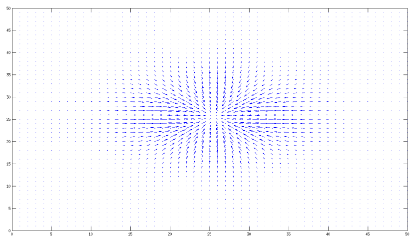

Figure 9: Gradient image for $\sigma = 3$

1.5.3 Threshold

See figure 13 for plots of thresholded gradient magnitudes for various thresholds and values of sigma.

1.5.4 Second Order Derivative

We implemented this function, see `ImageDerivatives.m`.

1.5.5 Impulse

When convolving with an impulse, the response should be the filter. This is what we see in figure 14. We notice that the shape of the kernel only depends on the order, while the size depends on the sigma.

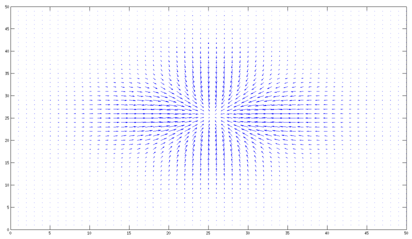

Figure 10: Gradient image for $\sigma = 5$

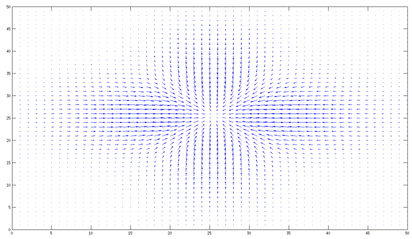

Figure 11: Gradient image for $\sigma = 7$

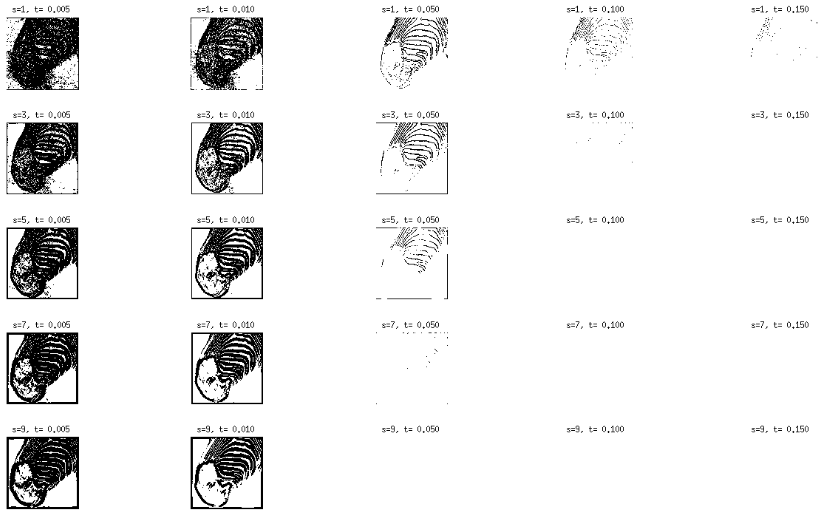

Figure 12: Threshold images for various σ and thresholds

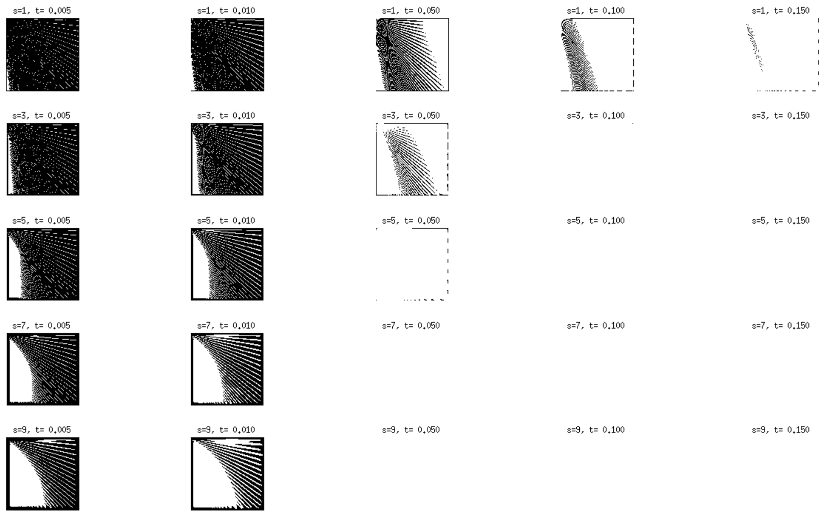

Figure 13: Threshold images for various σ and thresholds

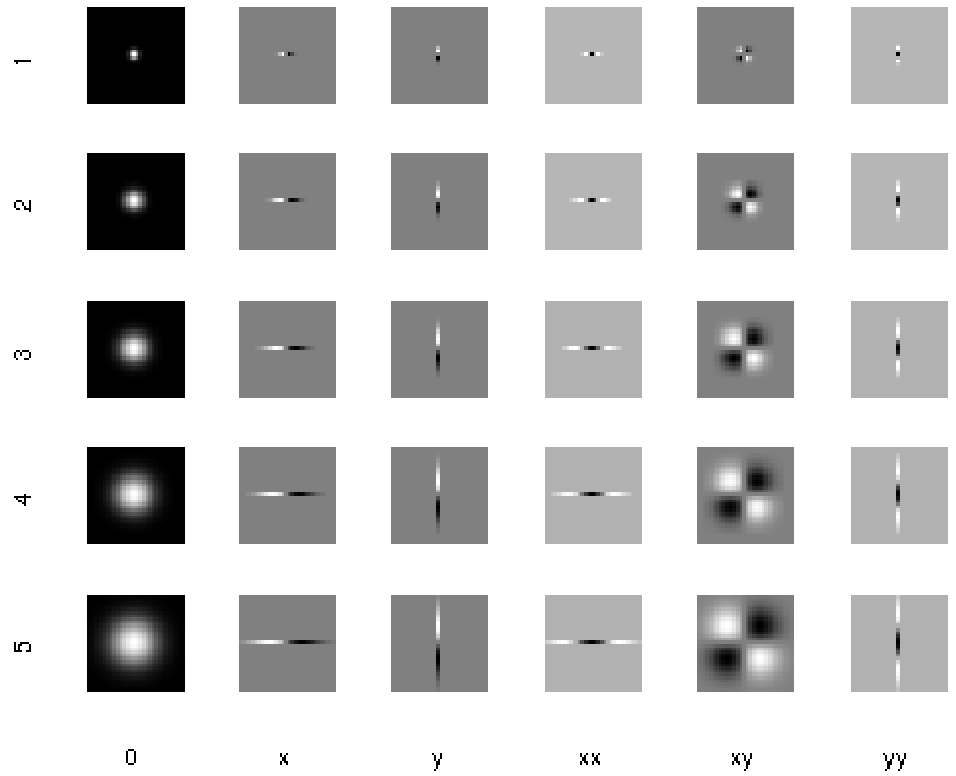

Figure 14: 30x30 impulse image convolved with various filters with $\sigma \in 1, 2, 3, 4, 5$