DD2423

Image Analysis and Computer Vision

Laboratory Report

Lab 2: Edge Detection & Hough Transform

Jiang, Sifan
sifanj@kth.se

April 7, 2019

1 Difference operators

- **Question 1:** What do you expect the results to look like and why? Compare the size of `dxtools` with the size of `tools`. Why are these sizes different?

  The x-wise derivative is expected to be an image with the edge of the original image in the x direction. The y-wise derivative is expected to be an image with the edge of the original image in the y direction. The result is shown in Figure 1.

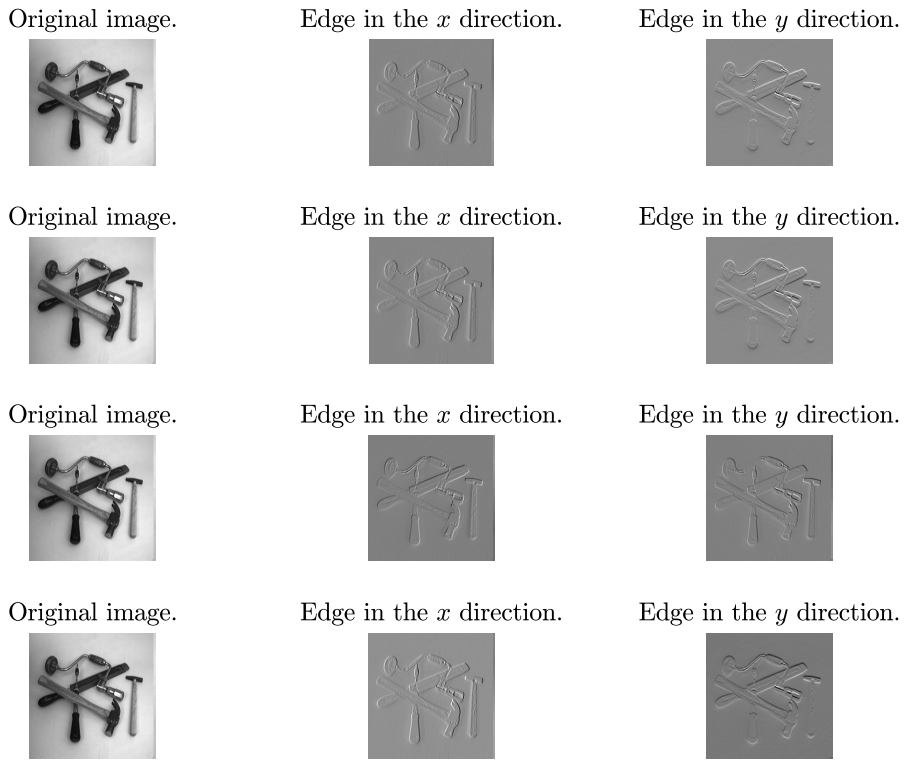

Figure 1: The first row of images is applied by simple difference operator. The second row is applied by central differences. The third row is applied by Robert's diagonal operator. The last row is applied by the Sobel operator.

The sizes of the derivative images are different from each other or from the original image because of the difference of the kernel sizes in each method. The image size is shown in Table 1.

Table 1: Image size.

Image	x direction ($y \times x$)	y direction ($y \times x$)
Original	256×256	256×256
Simple difference operator	256×254	254×256
Central difference operator	256×254	254×256
Roberts cross edge operator	255×255	255×255
Sobel operator	254×254	254×254

The reason why these sizes are different is because of the difference between the kernel size of each difference operators. If the image is of size $N \times M$ and in the case of the

simple difference operator, the kernel when considering the x-direction has size of 1×3. Since all the elements in the kernel should be multiplied by a element in the image, the kernel will fit the image N times in the y direction, but $M - 2$ times in the x direction.

2 Point-wise thresholding of gradient magnitudes

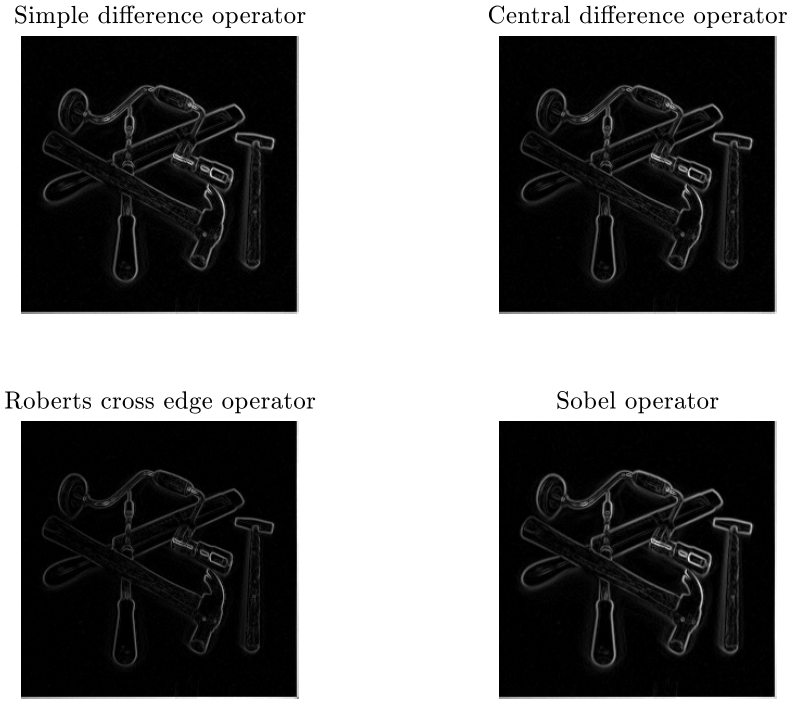

Figure 2: Gradient magnitude of the derivative images for `few256`.

- **Question 2:** Is it easy to find a threshold that results in thin edges? Explain why or why not!

  It is not easy to find a threshold that results in thin edges at the most time.

  - Applying a threshold to the whole image where the brightness for all edges are different would obtain thin edges for some objects while other edges could disappear or be too thick.
  - Different operator would give different result as illustrated in Figure 2. So the threshold for each operator should be different.
  - Noise with big magnitude could also make it hard to find a threshold that results in thin edges.

  However, in the case of `few256`, the histogram, Figure 3, accumulated to the left border of the histogram pretty well, so we can set the threshold based on the bin edges of the first cluster of the histogram. The threshold in the case of simple difference operator is 27.2. 13.6 for central difference operator. 32.7 for Roberts cross edge operator. 92.7 for Sobel operator.

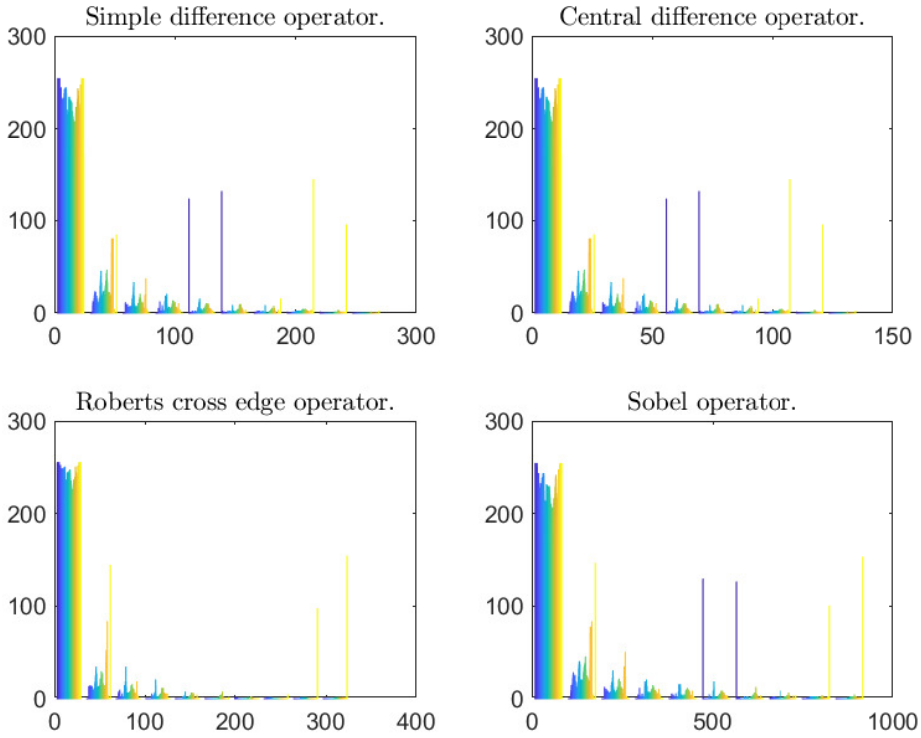

Figure 3: Histogram of gradient magnitude of the derivative images for `few256`.

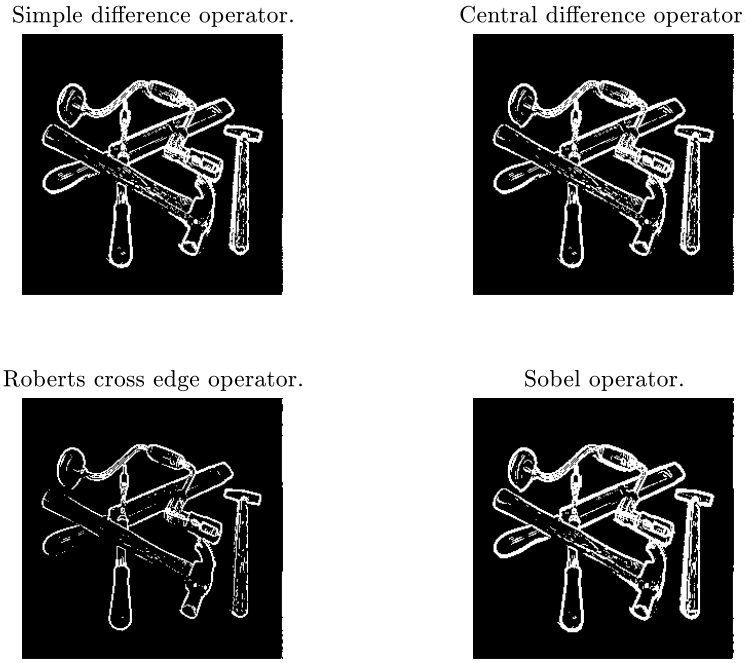

Figure 4: Gradient magnitude of the derivative images with different threshold for `few256`. The threshold in the case of simple difference operator is 27.2. 13.6 for central difference operator. 32.7 for Roberts cross edge operator. 92.7 for Sobel operator.

Figure 5: Gradient magnitude of the derivative images for `godthem256`.

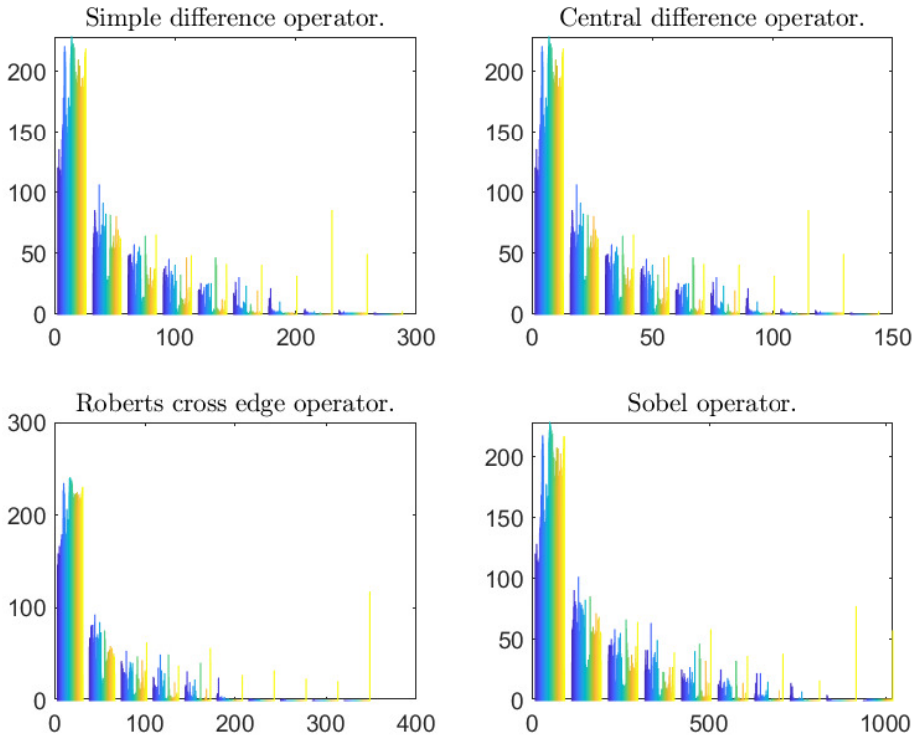

Figure 6: Histogram of gradient magnitude of the derivative images for `godthem256`.

Simple difference operator.

Central difference operator.

Roberts cross edge operator.

Sobel operator.

Figure 7: Gradient magnitude of the derivative images with different threshold for `godthem256`. The threshold in the case of simple difference operator is 29.1. 14.6 for central difference operator. 35.2 for Roberts cross edge operator. 103 for Sobel operator.

Simple difference operator.

Central difference operator.

Roberts cross edge operator.

Sobel operator.

Figure 8: Smoothed gradient magnitude using Gaussian filter with $\sigma^2 = 2.25$ for `few256`.

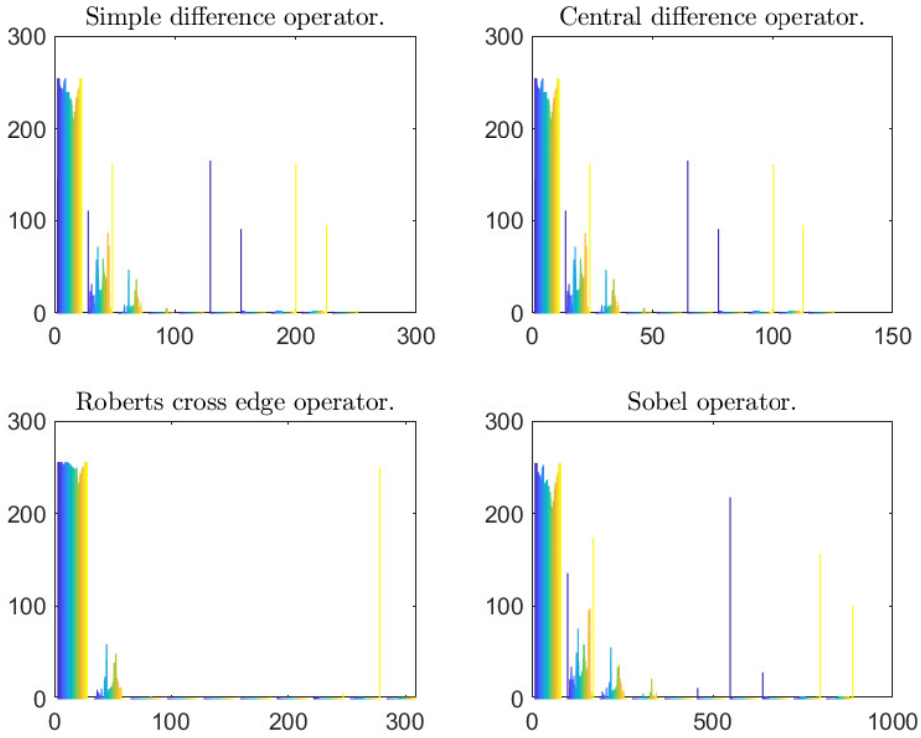

Figure 9: Histogram for smoothed gradient magnitude using Gaussian filter with $\sigma^2 = 2.25$ for `few256`.

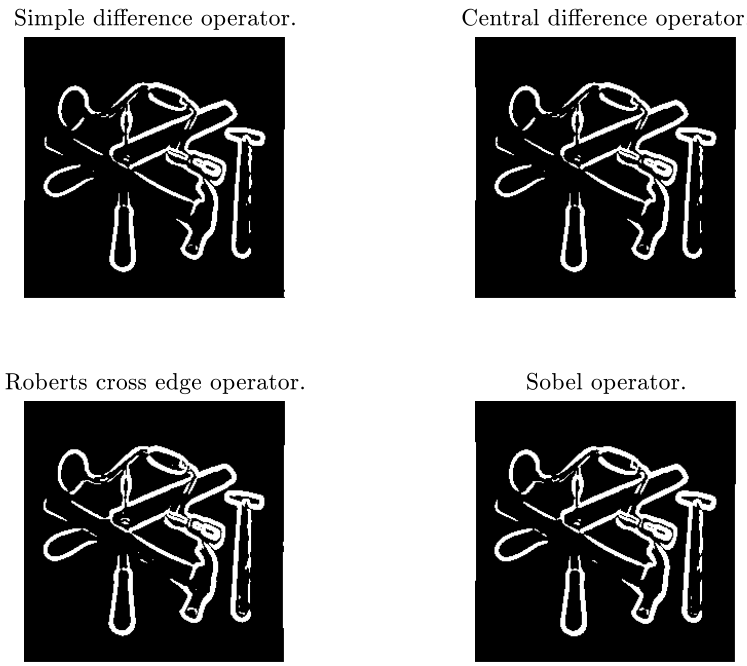

Figure 10: Smoothed gradient magnitude using Gaussian filter with $\sigma^2 = 2.25$ and different threshold for `few256`. The threshold in the case of simple difference operator is 25.4. 12.7 for central difference operator. 16 for Roberts cross edge operator. 89.9 for Sobel operator.

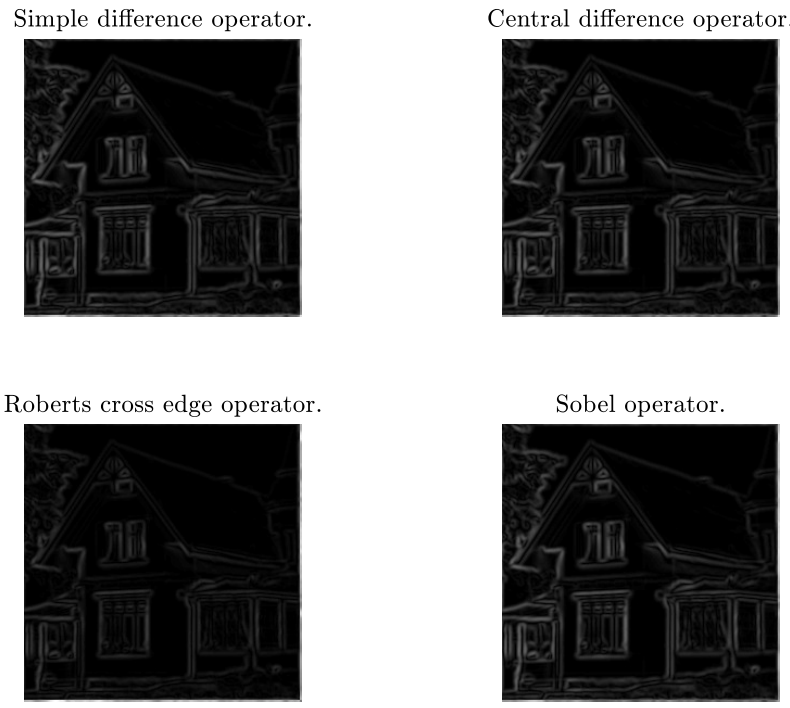

Figure 11: Smoothed gradient magnitude using Gaussian filter with $\sigma^2 = 2.25$ for `godthem256`.

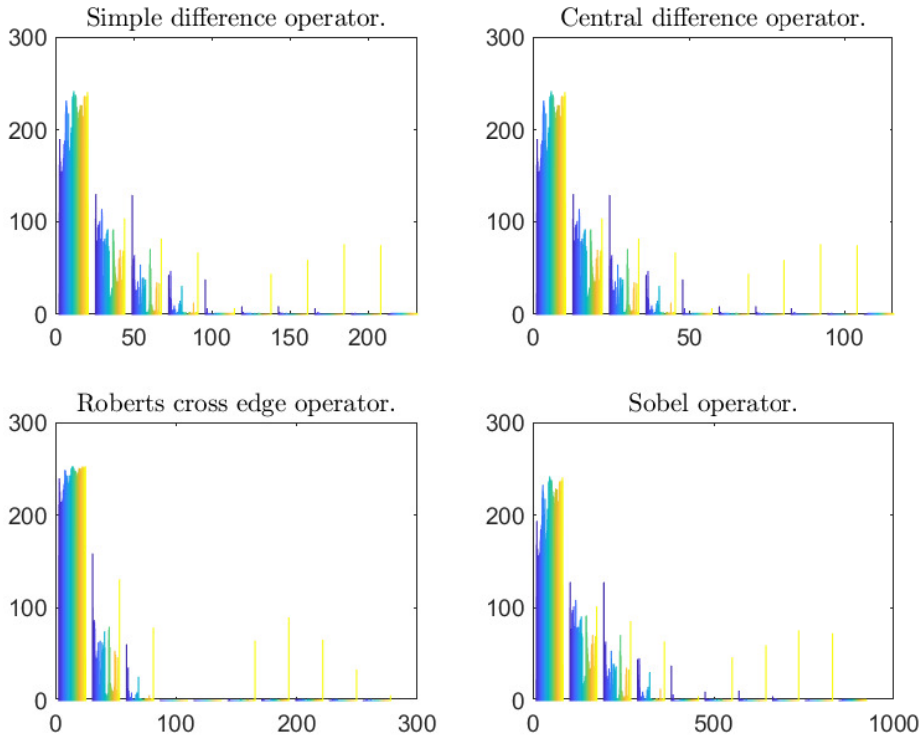

Figure 12: Histogram for smoothed gradient magnitude using Gaussian filter with $\sigma^2 = 2.25$ for `godthem256`.

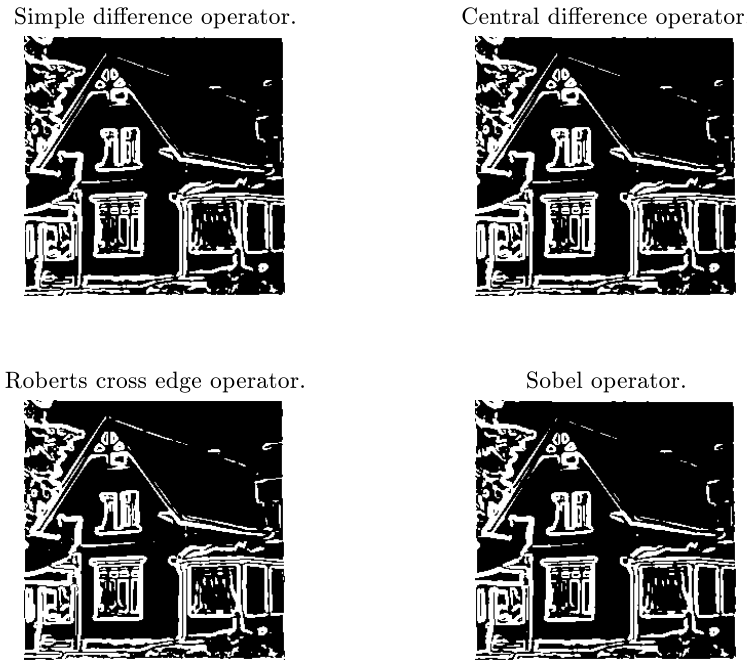

Figure 13: Smoothed gradient magnitude using Gaussian filter with $\sigma^2 = 2.25$ and different threshold for `godthem256`. The threshold in the case of simple difference operator is 23.4. 11.7 for central difference operator. 16 for Roberts cross edge operator. 93.4 for Sobel operator.

- **Question 3:** Does smoothing the image help to find edges?

  Yes, smoothing the image helps to find edges. Smoothing would remove noise from the image, thus making the detected "edges" are more likely to be real edges.

  However, smoothing the image with Gaussian filter could blur the edges, thus making it difficult to detect the edges with difference operators and find a threshold. This can also be seen from the histograms of the images after smoothing which are more shewed when compared to the gradient magnitude with no Gaussian smoothing.

4 Computing differential geometry descriptors

The images after applying Gaussian filter to `godthem256` with different value of σ^2 is illustrated in Figure 14. The second order derivative of `godthem256` after Gaussian smoothing is shown in Figure 15 and the third order derivative is shown in Figure 16. Also, the third order derivative of `few256` after Gaussian smoothing is shown in Figure 17.

- **Question 4:** What can you observe? Provide explanation based on the generated images.

  From Figure 14, we can observe that, the larger the scale is (which means the larger the variance of the Gaussian white noise is), the more blurry and the less noisy the image becomes. This is also true when comes to the result of the edge detection.

  However, higher variance of Gaussian filter also blurs the edges and making it harder to determine the real position of the edges. Also, from Figure 15, 16, and 17, we can see that Gaussian filter with higher variance would make the edges seem to be thickened thus losing the accuracy.

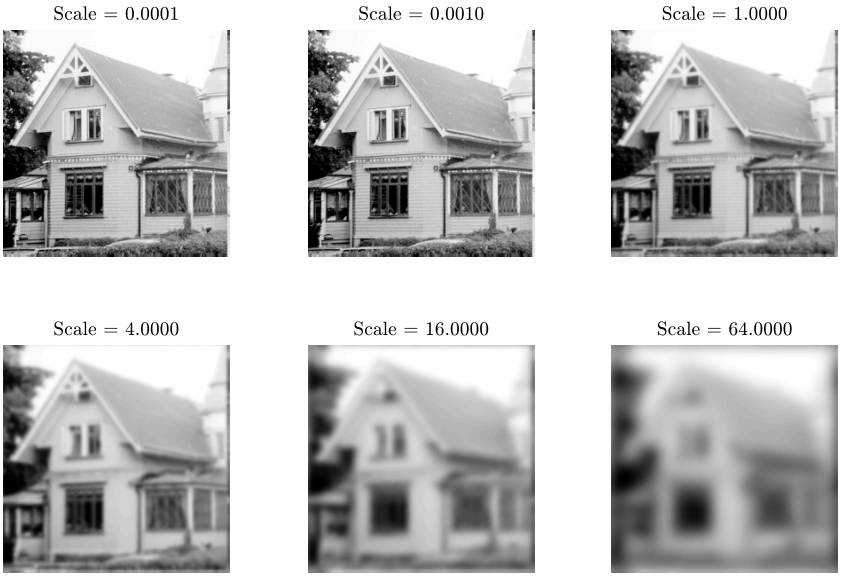

Figure 14: Gaussian smoothing for `godthem256` with different scale.

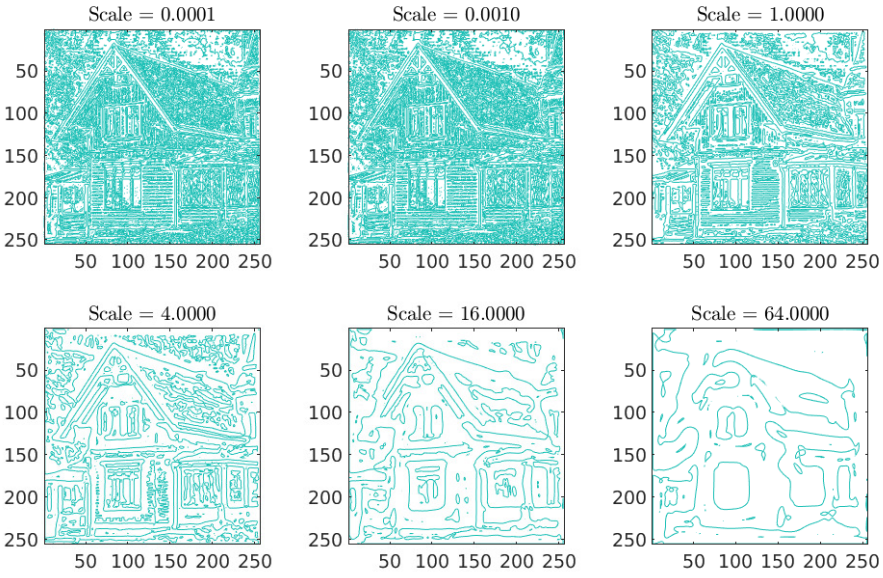

Figure 15: Second order derivative for `godthem256` after Gaussian smoothing.

- **Question 5:** Assemble the results of the experiment above into an illustrative collage with the `subplot` command. Which are your observations and conclusions?

  As stated at the beginning of this section, the images are shown in Figure 14, 15, 16, and 17, where the conclusion could be extracted: the bigger the variance of the Gaussian filter applied to the image is, the less noise will appear in the high order derivative while the less accurate the edges are found.

- **Question 6:** How can you use the response from $\widetilde{L}_{vv}$ to detect edges, and how can you improve the result by using $\widetilde{L}_{vvv}$?

  The local maximum and local minimum of the gradient magnitude would be reached

Figure 16: Third order derivative for `godthem256` after Gaussian smoothing.

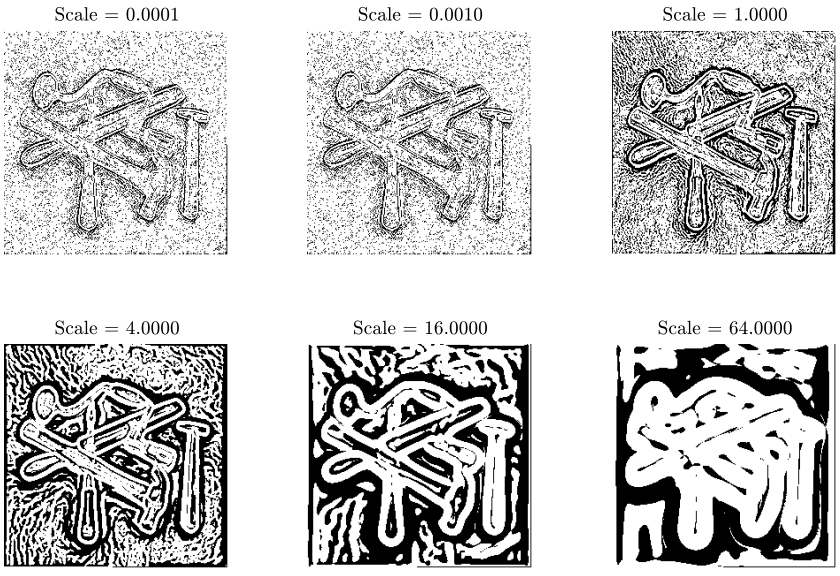

Figure 17: Third order derivative for `few256` after Gaussian smoothing.

when $\widetilde{L}_{vv} = 0$, however, if we want to find edges more accurately, we only want the local maximum be reserved. So when $(\widetilde{L}_{vv} = 0 \cap \widetilde{L}_{vvv} < 0)$, the local maximum would be reached. Based on such idea, we can improve the response from $\widetilde{L}_{vv}$ by combining the result of $\widetilde{L}_{vv}$ and $\widetilde{L}_{vvv}$ above. The combining result is shown in Figure 18 and 19.

5 Extraction of edge segments

- **Question 7:** Present your best results obtained with `extractedge` for `house` and `tools`.

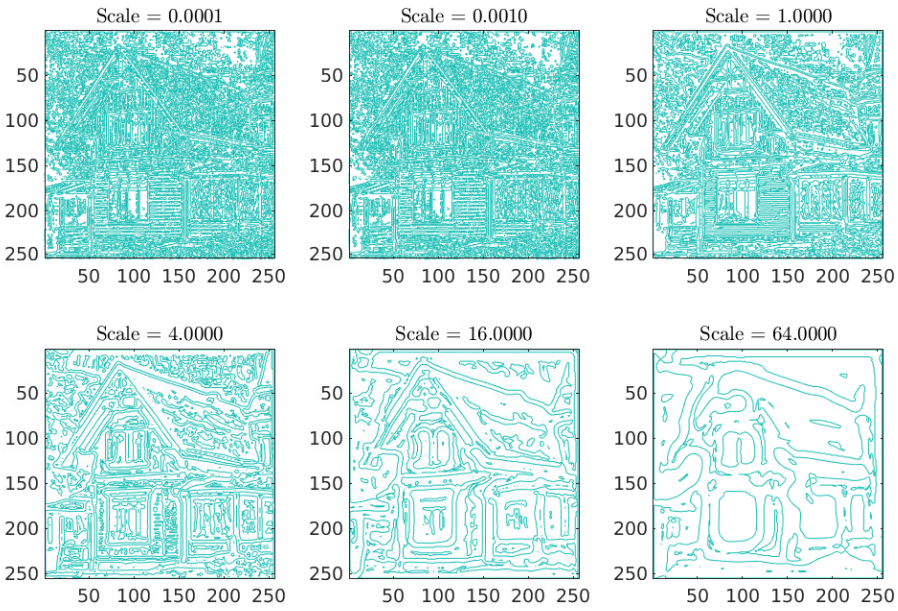

Figure 18: Combination of $\widetilde{L}_{vv} = 0$ and $\widetilde{L}_{vvv} < 0$ of `godthem256`.

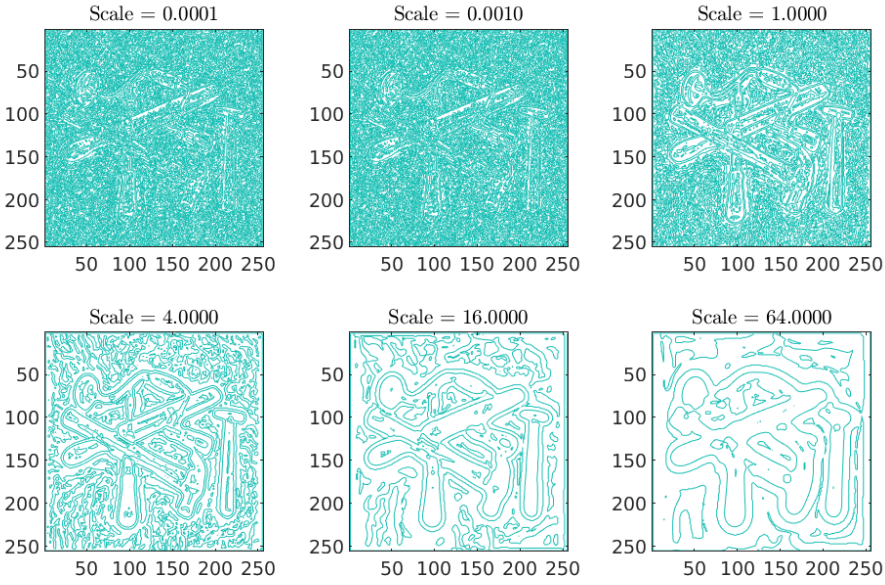

Figure 19: Combination of $\widetilde{L}_{vv} = 0$ and $\widetilde{L}_{vvv} < 0$ of `few256`.

The best results obtained for `house` and `tools` are shown in Figure 20 and 21 respectively. The `godthem256` is applied by Gaussian smoothing with variance $\sigma^2 = 4$ and threshold 4. The `few256` is applied by Gaussian smoothing with variance $\sigma^2 = 4$ and threshold 6.

Figure 20: Best result for `godthem256`.

Figure 21: Best result for `few256`.

6 Hough transform

6.1 Hints and practical advice

- **????? Question 8:** Identify the correspondences between the strongest peaks in the accumulator and line segments in the output image. Doing so convince yourself that the implementation is correct. Summarize the results in one or more figures.

  The strongest peaks in the accumulator should be the longest and the most obvious line segments in the edge plot.

- **Question 9:** How do the results and computational time depend on the number of cells in the accumulator?

  The accuracy of the results depends on the resolution of the accumulator since the larger of `ntheta` and `nrho`, the more accurate the points and curves would be presented in the accumulator thus the intersection points would be easier to find its real position in accumulator thus finding accurate line in spatial domain. However, high resolution could also lead to more local maximum, thus giving multiple lines for a single edge we would like to obtain from the image.

  When the resolution is higher, it would take much more computational time. The time complexity could be $O(n^2)$, where n is the resolution in x or y direction.

6.2 Choice of accumulator incrementation function

- **????? Question 10:** How do you propose to do this? Try out a function that you would suggest and see if it improves the results. Does it?

  I would like to try some monotonically increasing function like $\log$ or square root functions. The increasing rate of such functions becomes smaller when the magnitude becomes larger.