Answers to questions in Lab 2: Edge detection & Hough transform

Name: __Clément Gousseau___Program: __Master Degree in Computer Science_____

Instructions: Complete the lab according to the instructions in the notes and respond to the questions stated below. Keep the answers short and focus on what is essential. Illustrate with figures only when explicitly requested.

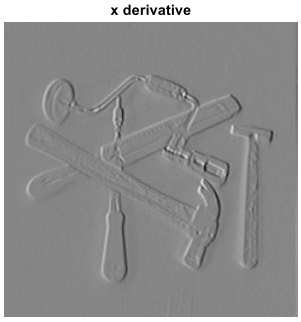
Good luck!

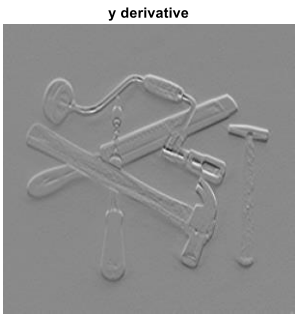
Question 1: What do you expect the results to look like and why? Compare the size of *dxtools* with the size of *tools*. Why are these sizes different?

Answers:

For dxtools, we expect the value to be large (ie white with showgrey) when there are changes on the horizontal direction. Therefore we should visualize the vertical edges.

For dytools, we expect the value to be large (ie white with showgrey) when there are changes on the vertical direction. Therefore we should visualize the horizontal edges.



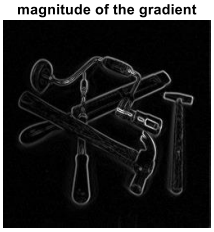


dxtools and dytools respectively have a size 256x254 and 254x256 because the derivative is not defined on the borders.

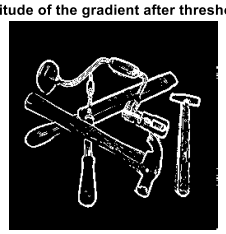
Question 2: Is it easy to find a threshold that results in thin edges? Explain why or why not!

Answers:

magnitude of the gradient



magnitude of the gradient after thresholding



magnitude of the gradient

gradient magnitude after thresholding

Finding a good threshold is difficult since it depends on the complexity of the image. 35 was a good threshold to see thin edges for the tools image whereas 1800 was a good threshold for the house image

---

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

Smoothing helps to find edges since it removes noisy patterns that may be interpreted as edges.

Answers:

original image

thresholded gradient magnitude without smoothing

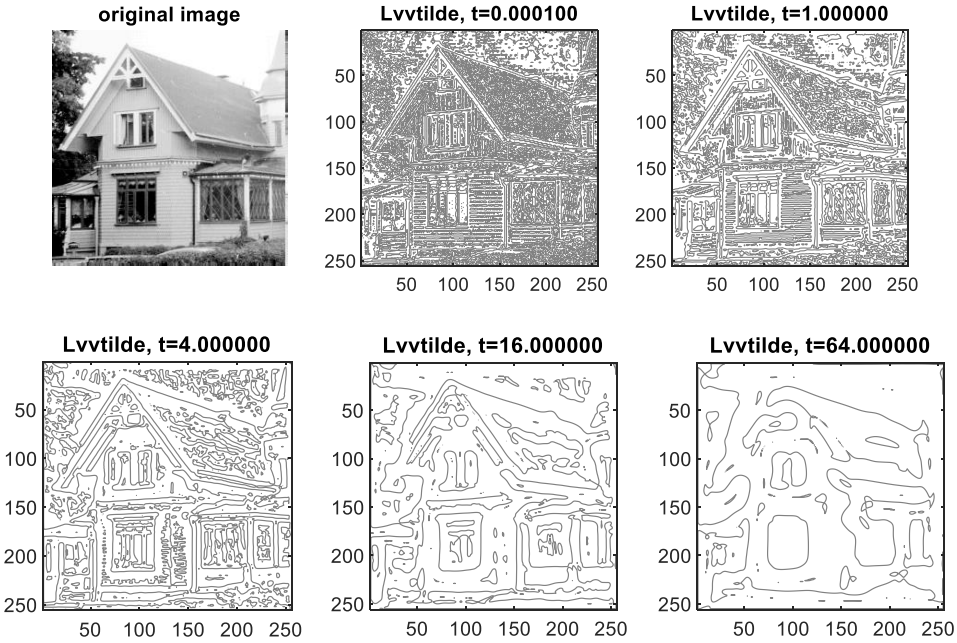
original image after smoothing

thresholded gradient magnitude with smoothing

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

Answers:

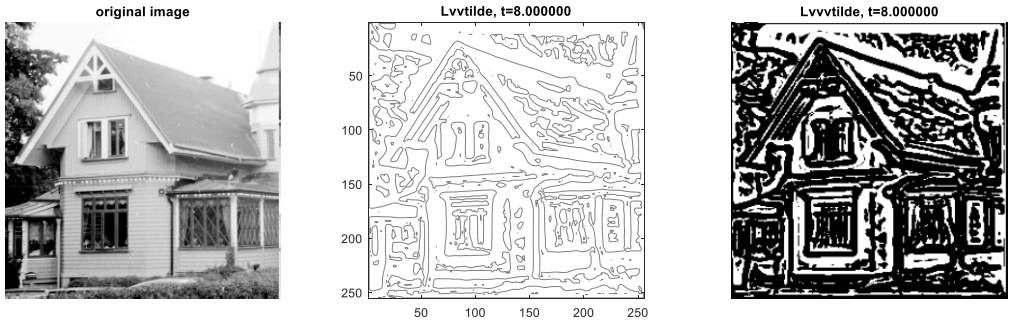
When t is too low (t=0.0001 and t=1), the smoothed image is very similar to the original image, Lvvtilde is too noisy to extract edges or any relevant information. With t=4 and t=16 the results are much better since some noise is removed and so we can see some edges. With t=64, the image is too much smoothed, both noise and edges are destroyed.



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

Answers:

On the plots below we can see that both Lvvtilde and Lvvvtilde contain information about the edges.



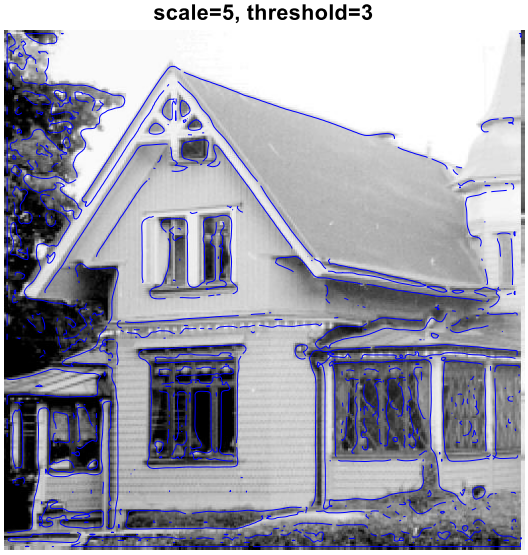
Question 6: How can you use the response from *Lvv* to detect edges, and how can you improve the result by using *Lvvv*?

Answers:

We could combine the information from Lvv and Lvvv, that is to say considering a point is part of an edge if both Lvv=0 and Lvvv<0

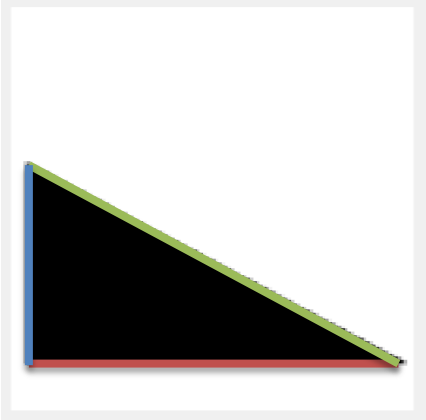
Question 7: Present your best results obtained with *extractedge* for *house* and *tools*.

Answers:

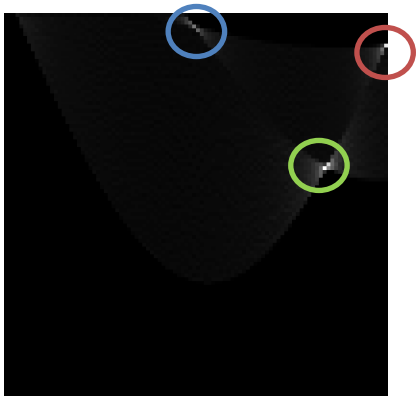


Question 8: Identify the correspondences between the strongest peaks in the accumulator and line segments in the output image. Doing so convince yourself that the implementation is correct. Summarize the results of in one or more figures.

Answers:



Expected results:
Blue line: theta=0 and rho very small
Red line: theta=pi/2 and rho small
Green line: theta≈pi/3 and rho pretty large

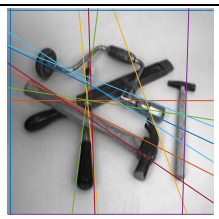
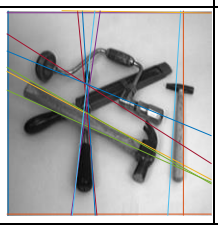
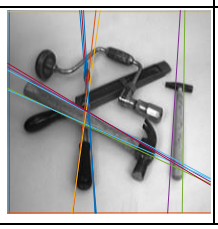
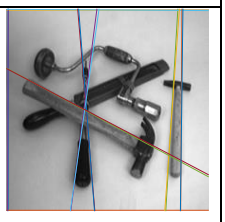


Actual results:
The results are as expected; the blue area corresponds to the blue line, the red area corresponds to the red line, the green area corresponds to the green line,

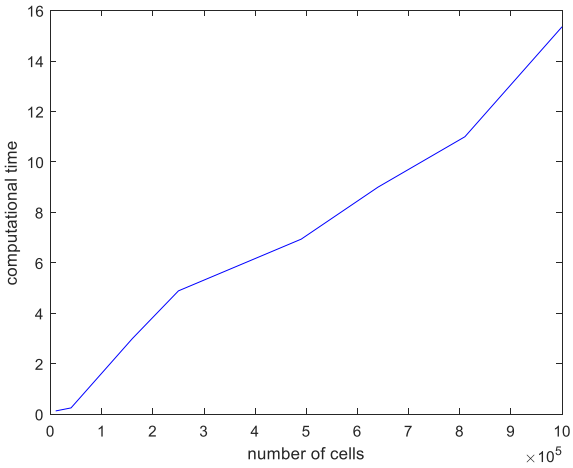
---

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

Answers:

Accumulator	100x100	200x200	500x500	1000x1000
Number of cells	10^4	4x10^4	2.5x10^5	10^6
Computational time (s)	0.13	0.25	4.89	15.36
Results				

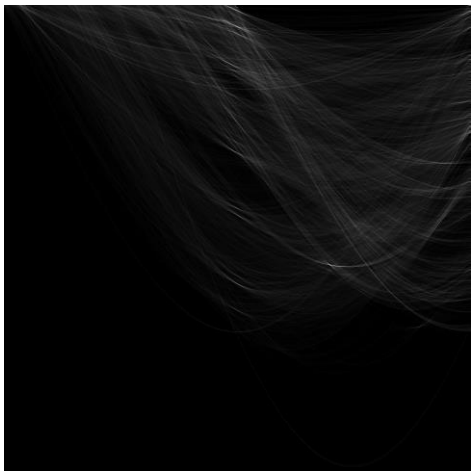
With a low number of cells, the results are pretty bad. For instance with a resolution 100x100 some lines does not correspond to actual lines. With a high number of cells the results are much better, the lines correspond to actual edges and the precision of these lines is pretty good. However the computational time can be pretty high. It grows linearly with the number of cells.



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?

Answers:

H(x) = log(1+x). This function increases the importance of edges with low gradient magnitude that may not be edges. Therefore some non-edges are considered as edges.




H(x) = exp(x). This function promotes edges with a large gradient. This gives better results, there are fewer misclassified edges.