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

Name: Mikel Zhobro
Program: ___________________________

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:
As expected the partial derivatives deliver different results. While the partial derivative in x-direction captures better gradient differences in x-direction, the partial derivative in y suffers on detecting changes in x-direction. That can be seen best at the hammer stick on the second volume on the figure below.
Depending on the filter's size the output image decreases its size, since once the filter is on the border it cannot move anymore in that direction. A possible solution for this would be padding the contour of the input image.

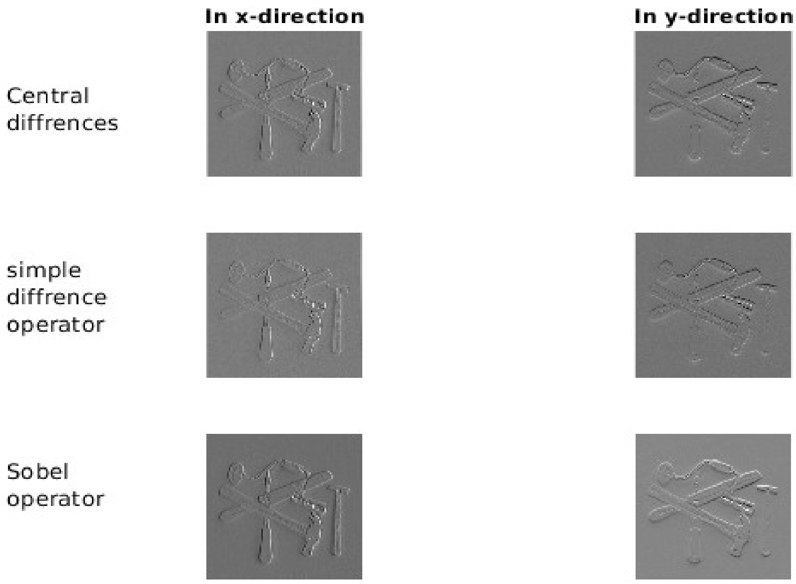

Figure 1.

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

Answers:
It is not easy to find a proper threshold, since the range of gradient magnitudes goes up to 255^2. In addition to that on the edges the gradient difference happens gradually so that even in the real image the edges are thicker than one pixel or two.

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

Answers:
Smoothing helps finding the edges by averaging out the noise which otherwise would have been seen as high gradients and so falsely assumed to be edges.

But we observe two main problems with edge detection based on thresholded gradient calculations. Noise very often is considered to be edges because of its high gradient. This is counterattacked by using smoothing techniques, which on the other side have the disadvantage of maybe losing desirable edges. That is why, at the end, a trade if between both methods has to be found. The figure below supports these arguments.

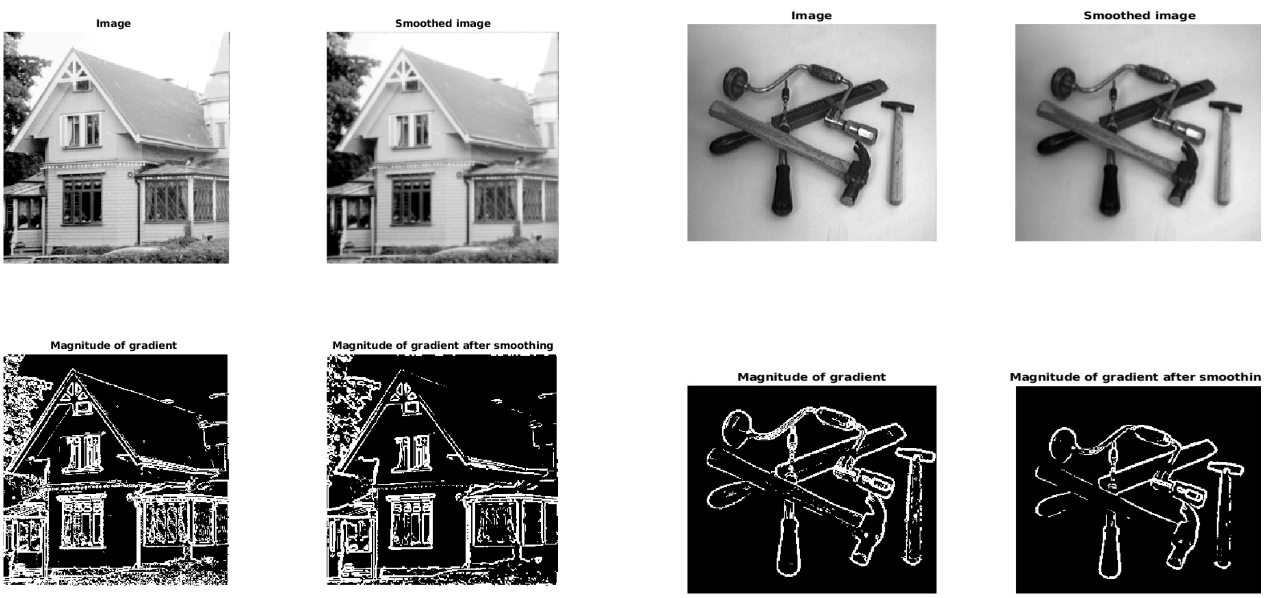

Figure 2.

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

Answers:
In the figure below it is noticable that edge detector based on second order derivatives are highly noise sensitive, only after Gaussian smoothing with variance 64 is it able to deliver good results. In addition to that we notice that edges are thin.

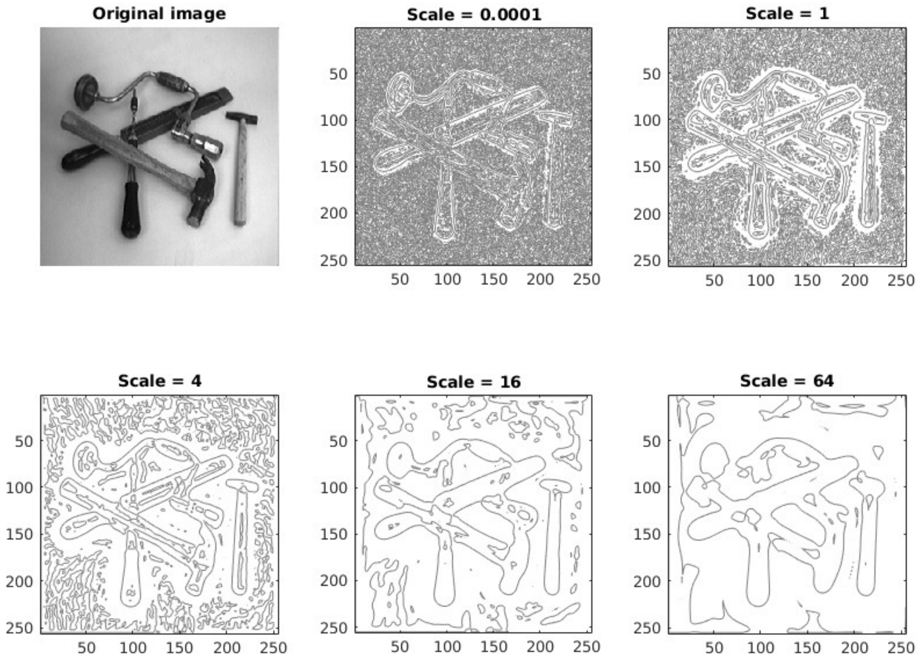

Figure 3.

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

Answers:
We observe that using third order derivative is very noise sensitive. Compared to the two other lower order derivatives it is rather unclear where the edges are.

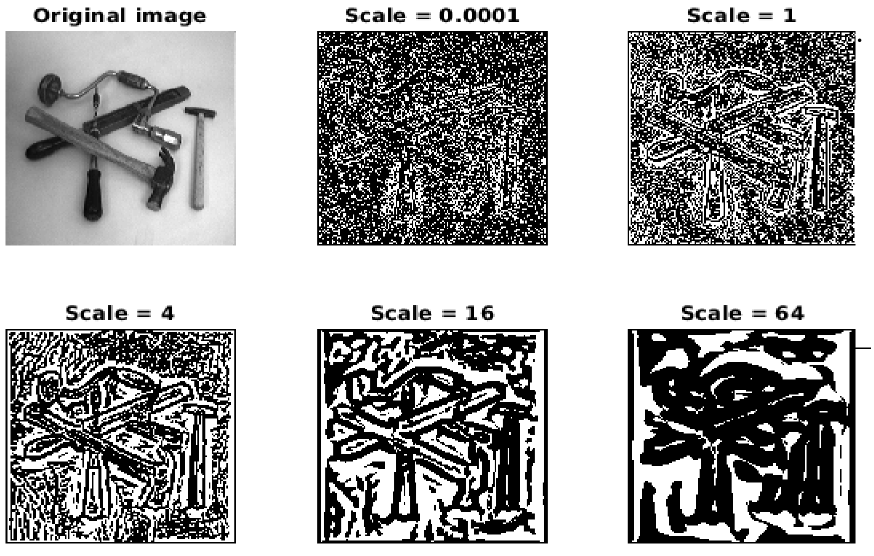

Figure 4.

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

Answers:
Edges correspond to places where first order derivative is maximal which on the other hand can be seen as points where the Laplacian(second order derivative) is zero. A possible way to improve the solution is to fuse the both filters together by finding the points where Laplacian is zero and the third order derivative is smaller than 0 at the same time and hope this improves the quality.

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

Answers:

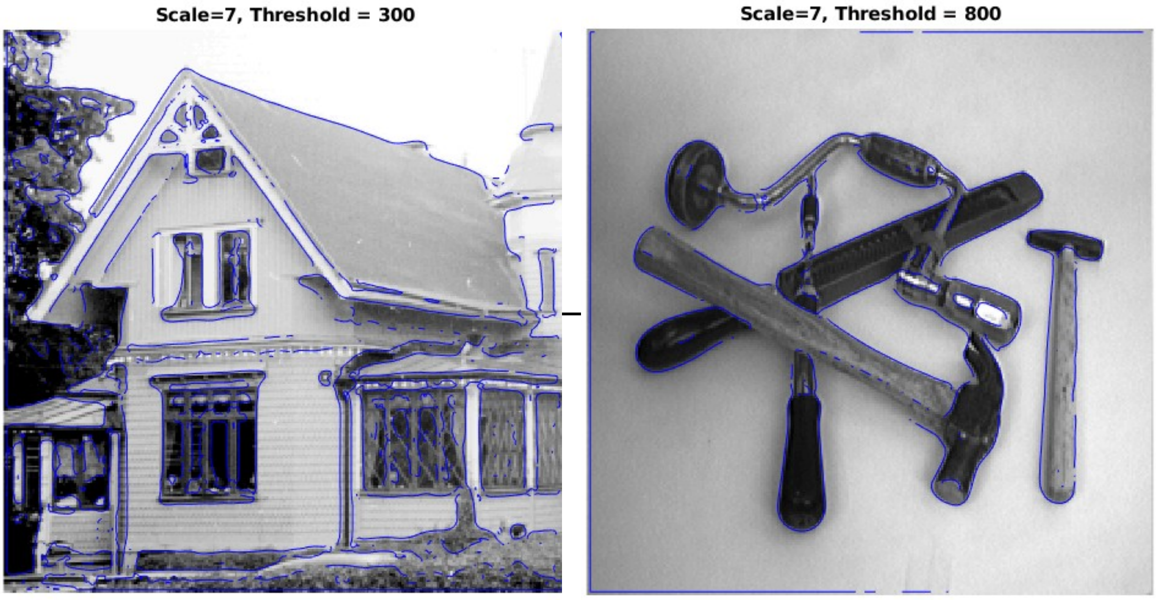

Figure 5. Combination of all methods for edge detection

Question 8: Identify the correspondences between the strongest peaks in the accu-mulator 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:
The θ and ρ in the Hough space correspond to the lines in the image space. The peaks in the middle have more than 1 pixel.

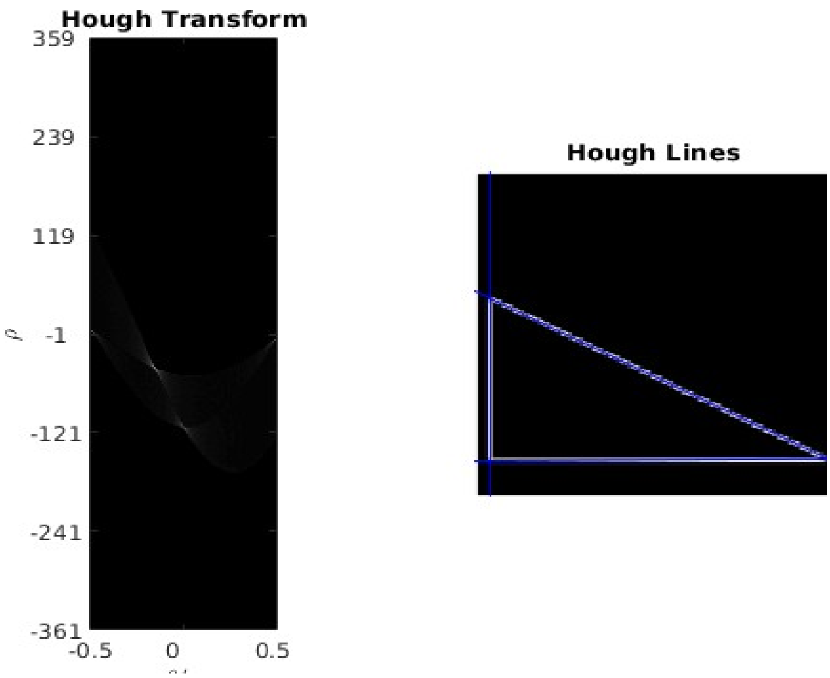

Figure 6. Hough Transformation with (#ρ=1400, #θ=400, #Lines=3)

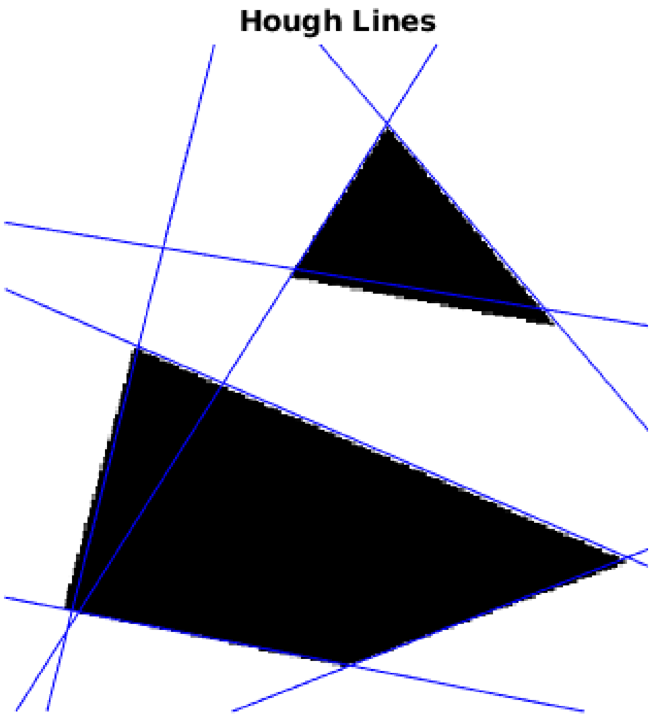

Figure 8.
Hough Transformation with (#ρ=4080, #θ=180, #Lines=7)

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

Answers:
Computational complexity is polynomial corresponding to number of angles. On the other hand the nr of rhos influences only the time for preallocation of the accumulator which is not significant. After trying with different number of parameters it was noticable that a high number of rhos helps with the quality of lines without adding too much computational time.

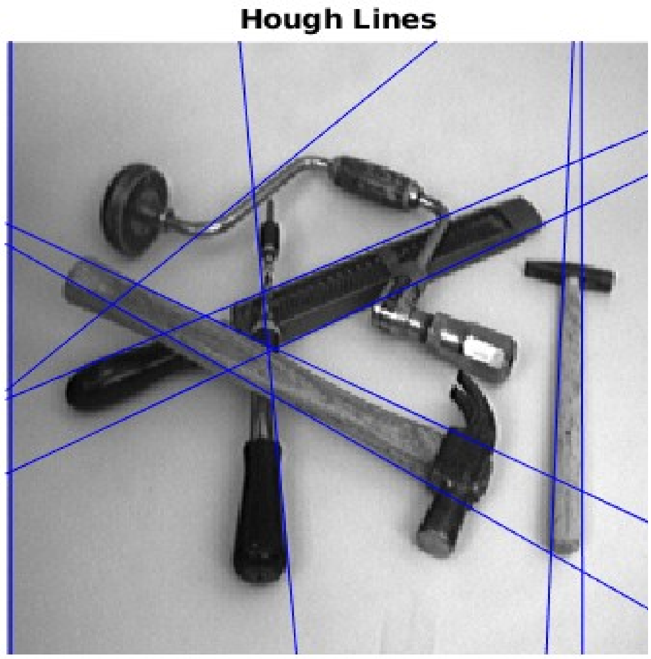

Figure 8. Hough Transformation with (#ρ=4500, #θ=150, #Lines=10)

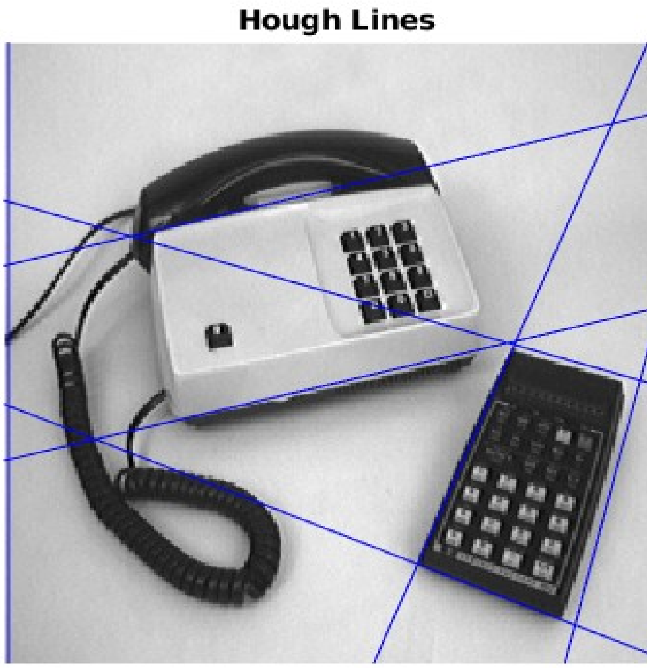

Figure 9. Hough Transformation with (#ρ=6500, #θ=130, #Lines=10)

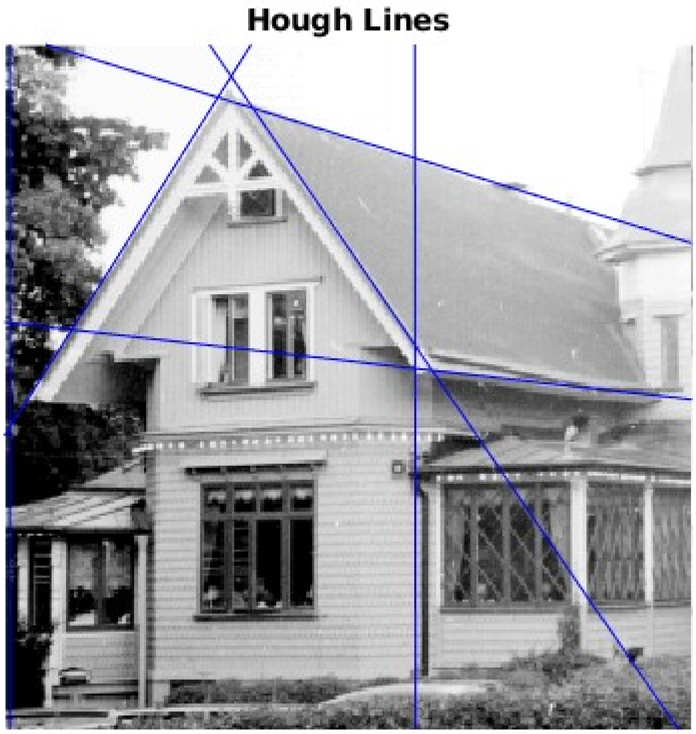

Figure 10. Hough Transformation with (#ρ=4000, #θ=170, #Lines=7)

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:
Realizing this concept can be done by introducing a type of weighting for votes given by a function which increases with the gradient at that point. Some possibilities include log functions or different polynomial functions.
Event though it sounds logical to give higher weight to higher gradients, from the observations we can only say that the constant and the log increment deliver the best results.

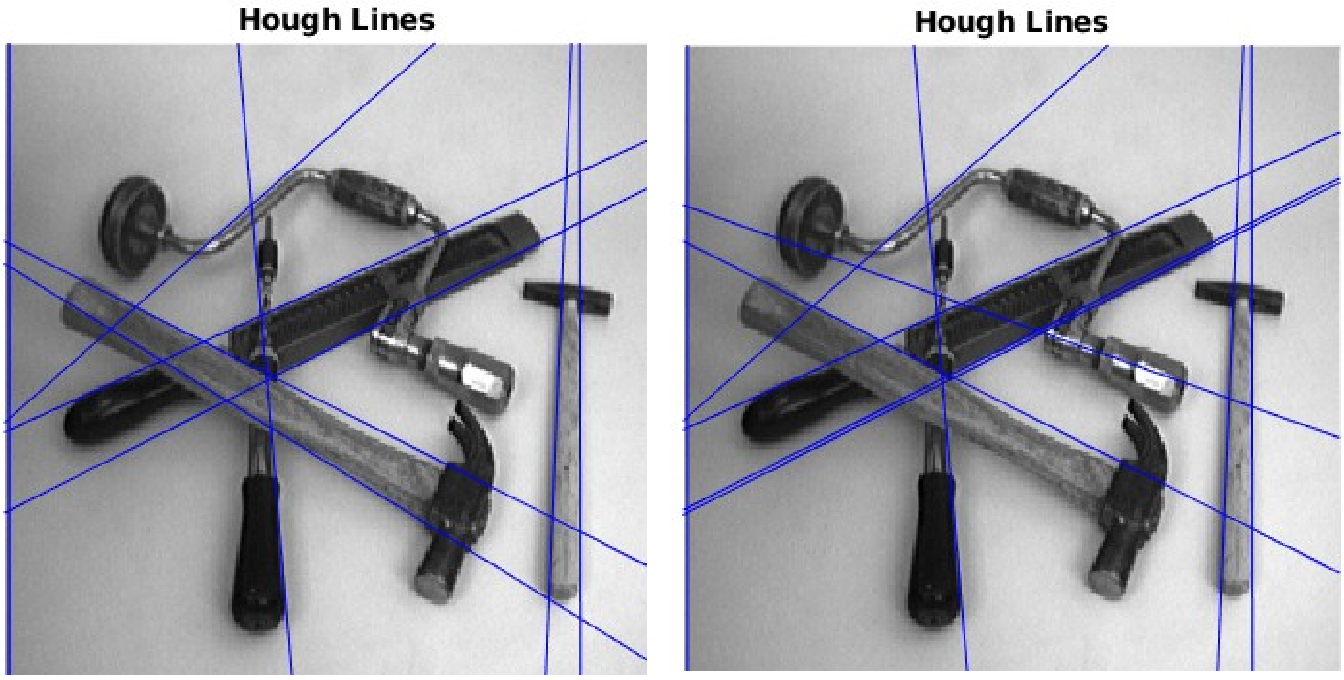

Figure 11. h=1 [left] and h=log(m) [right]

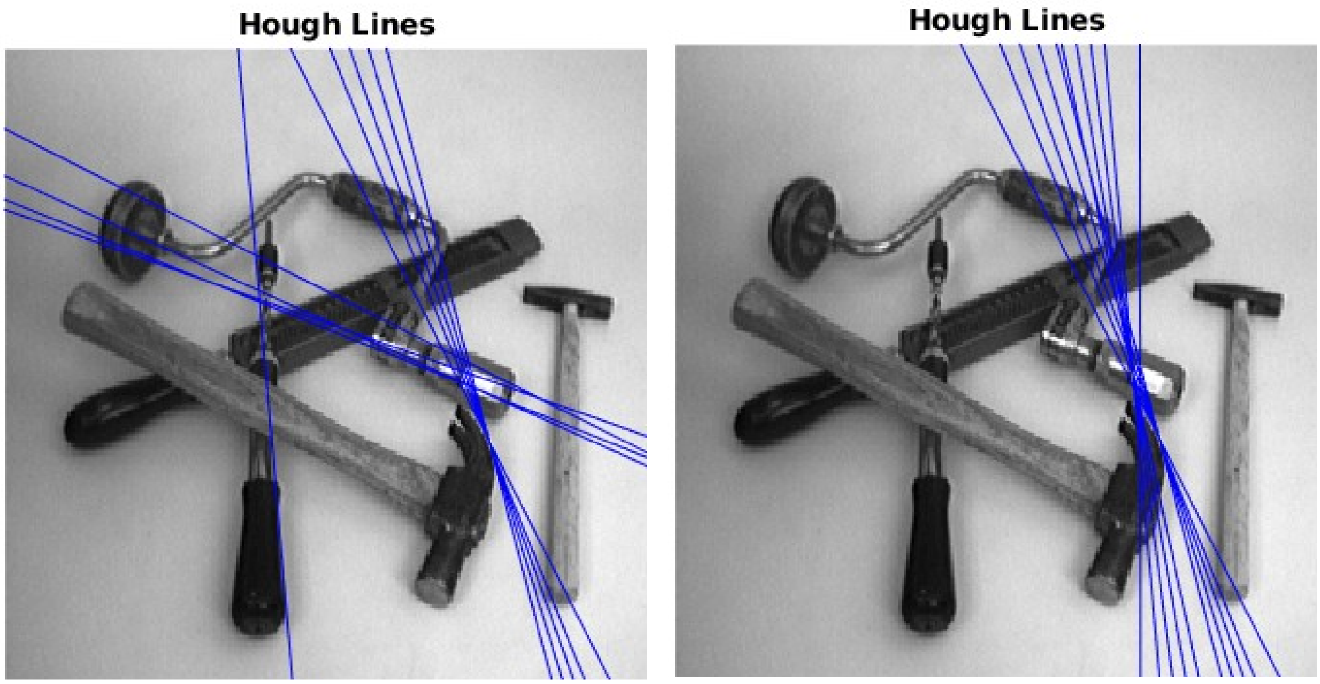

Figure 12. h= m^2 [left] and h=m^3 [right]