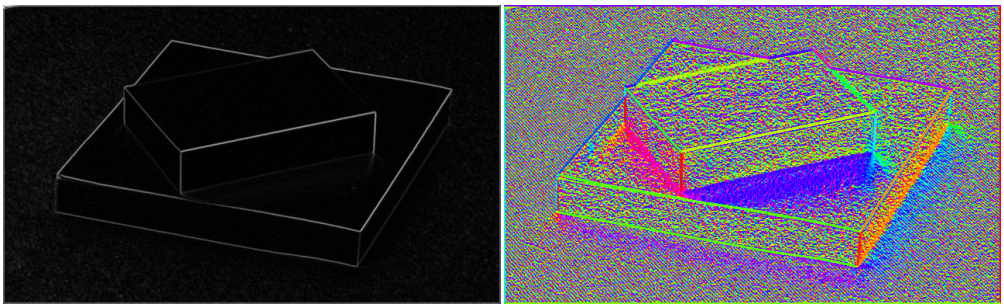
Lecture Machine Vision 2019/20
Institut für Mess- und Regelungstechnik, Karlsruher Institut für Technologie
Dr. Martin Lauer, Christian Kinzig, M.Sc.

Example Solutions for Practical Exercises: Edge Detection and Hough Transform

Since Matlab offers many functions for low-level image processing the exercises can be solved with only little own programing. Throughout the solution sheet we assume that the image is stored in a variable named `postit2g`.

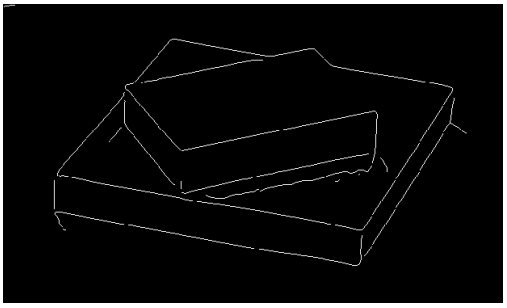
- **Calculating the gray level gradient**
  We need to perform several steps:

  1. calculate the horizontal partial derivatives of the image by:
     `gu = imfilter(im2double(postit2g), double(fspecial('sobel'))');`
  2. calculate the vertical partial derivatives of the image by:
     `gv = imfilter(im2double(postit2g), double(fspecial('sobel')));`
  3. calculate the gradient length by:
     `glen = sqrt(gu.^2+gv.^2);`
  4. calculate the gradient angle by:
     `gphi = atan2(gv,gu);`
  5. finally, display `glen` and `gphi` by:
     `figure; imshow(glen,[]);`
     `figure; imshow((gphi/(2 * pi)+0.5), 'Colormap', hsv);`
     Note, that the additional attribute *'Colormap'* in the last call of *imshow* can be used to specify a colormap *hsv* which displays the numbers between 0 and 255 in different colors instead of different gray levels.

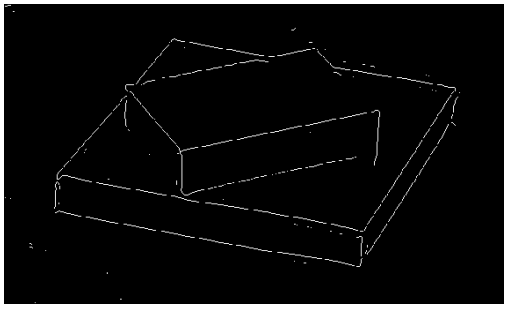


- **Edge image**
  After trying many different values for the upper and lower threshold and the σ-value of the Gaussian filter a reasonable result could be achived by:
  `Icanny = edge (postit2g, 'canny', [0.05 0.15], 2);`

However, it was not possible to detect all object boundaries without finding artificial edges due to shadow.
Trying the same with the LoG-approach returned the best results with:
`Ilog = edge (postit2g, 'log', 0.0008, 3);`
Again, finding the right parameters was not easy. Some contours are not perfectly connected, however, the result is not noise-free.

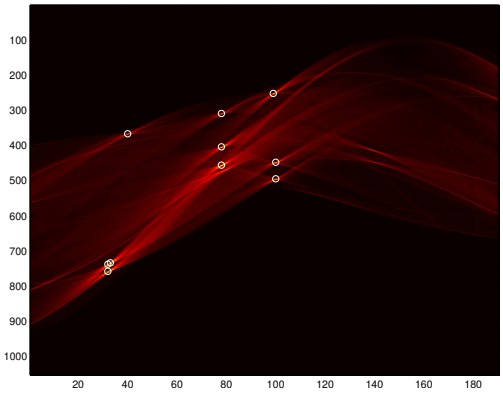


result with Canny-operator



result with LoG-operator

- **Hough transform**
  The hough transform can be applied onto the edge image by:
  `hs=robust_hough(Icanny);`
  The 10 most dominant line segments are extracted calling:
  `ln=robust_hough_lines(hs, 10, Icanny);`
  The result can be visualized by:
  `robust_hough_plot_lines(postit2g, ln);`



Hough-space with local maxima (white circles)

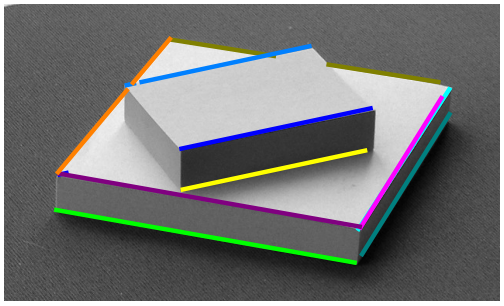


image with line segments found by the Hough-transform