CS6640 – Project 4 Canny Filter: Step- and Line Edges
Assigned Nov. 3, 2010
Due Wed Nov. 17 (Just before midnight)
Instructor: Guido Gerig
TA: Miaomiao Zhang

Goals

The purpose of this assignment is to implement the image filtering strategies developed by John Canny (Canny filter) for step edges and line edges (detection of line- and curve structures). This combines several techniques discussed so far into a system, including filtering, derivatives, peak detection, thresholding of edge maps, histogram analysis and connectivity to find connected parts of contours.

Materials: Slides and additional documents on the Canny filter, including Canny's paper, uploaded on the web under *slides*. Please also see the notes I and notes II (handwritten) that summarize Canny's approach to extract thin contour fragments.

1 Canny Filtering

The Canny filter is built upon a series of steps extract edge and contour fragments from images.

1.1 Image Filtering

The first stage is Gaussian filtering of an image. Please refer to project 2 where you implemented Gaussian filtering, e.g. via an efficient separable filtering in x and y. The only input parameter is the Gaussian width σ which decides about blurring and noise reduction.

1.2 Gradient: First Derivatives

Build the gradient of the Gaussian smoothed image $\nabla(G(\bar{x},\sigma) \otimes I(\bar{x}))$ which is the vector composed of the first derivatives in x and y, $\frac{\partial}{\partial x}$, $\frac{\partial}{\partial y}$, applied to the Gaussian smoothed image.

Calculate the edge magnitude image as required in project 2, i.e. vector norm $\sqrt{IG_x^2 + IG_y^2}$, where IG is the Gaussian filtered image. This results in the edge map image.

1.3 Non-maximum Suppression

As discussed in the course, and in the course material (paper, slides, handouts), Canny proposes a non-maximum suppression of the edge map to get thin pixel lines at the top of the smooth edge map, i.e. only to keep the ridge pixels and eliminating all the others.

Non-maximum suppression is achieved by checking each pixel if it is a peak in normal direction, i.e. if its immediate neighbors in normal direction have lower values than the center. A simple procedure is the calculation of the normal direction, and looking at the closest raster neighbors (vertical, horizontal and diagonal), and then testing if the center is higher than the neighbors. A more sophisticated and accurate approach (see slides) is the use of the normal to the edge, interpolation of edge map values at two sides of the normal, and then again applying the simple peak test of center versus normals.

The nonmaximum suppression simplifies the edge map into an image where only the peaks are kept. At those positions, keep the edge magnitude, whereas at all other locatioins, replace them by 0.

Implement a nonmaximum suppression scheme to be applied to the edge magnitude map.

1.4 Thresholding

The edge map can finally be thresholded to keep only the edge pixels with values above a user-defined threshold. This can be done by a user choosing heuritic thresholds to test if the result is acceptable. **Implement a threshold with user-selected value.**

A more sophisticated approach is automatic selection of a suitable threshold. Canny's paper describes a framework where you can build a histogram of the edge magnitude image, then build the cumulative distribution function of the histogram, and then select a high percentile (e.g. 80% or 90%, e.g.) for thresholding. Such a procedure assumes that the first 80% or higher is just response to noise, where the upper 20% or less percent pixels can be true edge pixels. **Implement such a histogram-based analysis for automatic threshold selection**.

1.5 Hysteresis Thresholding

Canny developed a new system, called hysteresis thresholding (see notes II and Canny paper) that works with double thresholding. He defines two threshold, a lower and higher threshold I_l and T_h. All pixels above the lower threshold are edge candidates, the others are set to 0. All pixels above the higher, restrictive threshold are so called seed regions for good edge contours. The idea is that all pixels above the lower threshold but connected to such a seed pixel are kept, whereas pixels above the threshold but not connected to such a seed are assumed to be noise and deleted. The notes explain why such a scheme can avoid broken contours and efficiently remove small edge response due to noise.

The procedure requires the definition of a connectivity, i.e. check for every pixel above the lower threshold if connected via the d_m metric to a seed. You need to develop a strategy to start from the seed regions and walk through the connected pixels (candidates above the lower threshold but smaller than the higher threshold). As long as they are connected, keep them as edges.

This way, you should get a cleaner edge map where lots of noise is removed and contours are less fragmented. The lower and higher thresholds can be chosen manually or estimated via the histogram-based approach with 2 different percentile choices.

Implement a scheme for hysteresis thresholding

1.6 Tests on images

The procedure can be tested on a synthetic checkerboard image and/or the synthetic edges-lines images. Try to add different level of noise to make edge detection very hard.

Choose your favorite real world image which contains many step edge structures, and apply your Canny edge framwork for extraction of edges.

2 Instructions, Requirements, and Restrictions

1. Please use your name **"NAME"** in all reportts and submitted materials to uniquely identify your projects.
2. Write your project code in a single directory, called `project1-NAME`.
3. For Matlab each individual function (including functions you define) should be a ".m" file, and your functions should call one another as necessary.
4. We **do not allow to use Matlab toolbox functions** (e.g. Imaging Toolbox) or other existing image processing libraries in order to give all students the same conditions for code development [1].
5. You should have in your report a short description of each algorithm you used and documentation on how your code is organized. Failure to do this will result in a loss of points. Please remmember to **add your name** to the report title.

[1] The core MATLAB package comes with several rudimentary functions that can be used to load, save, and perform custom functions on images. Taken from wikibooks

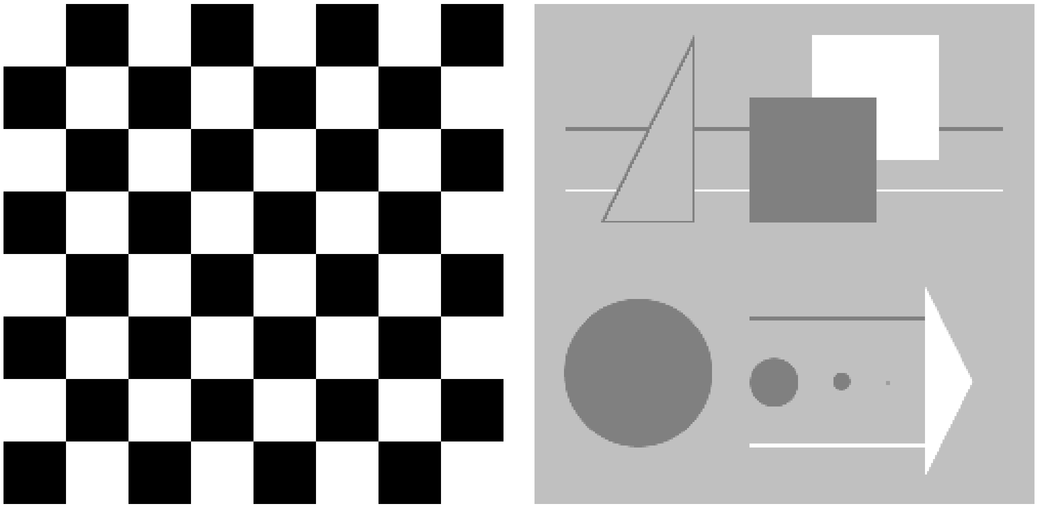

Figure 1: Test images for Canny edge detection

6. **You can use submit your report as html or pdf document** (created from Word, LaTeX or other word processing system of your choice.
   - Using html: Your project report will be in the form of an html file called `index.html`, contained in that directory. All links from that file must be relative and all other files necessary to read your report must be in that directory (or subdirectories).
7. You should use examples of images in your report. They should be viewable in the browser when we open your html file.
8. You will submit a single tar file created from from your project directory with the unix command *tar -czf project1-NAME.tgz./project1-NAME.*
9. Please remember or look-up the honor code and requirement to provide your own solution as discussed in the syllabus.
10. **Please look up the late policy as defined in the syllabus**