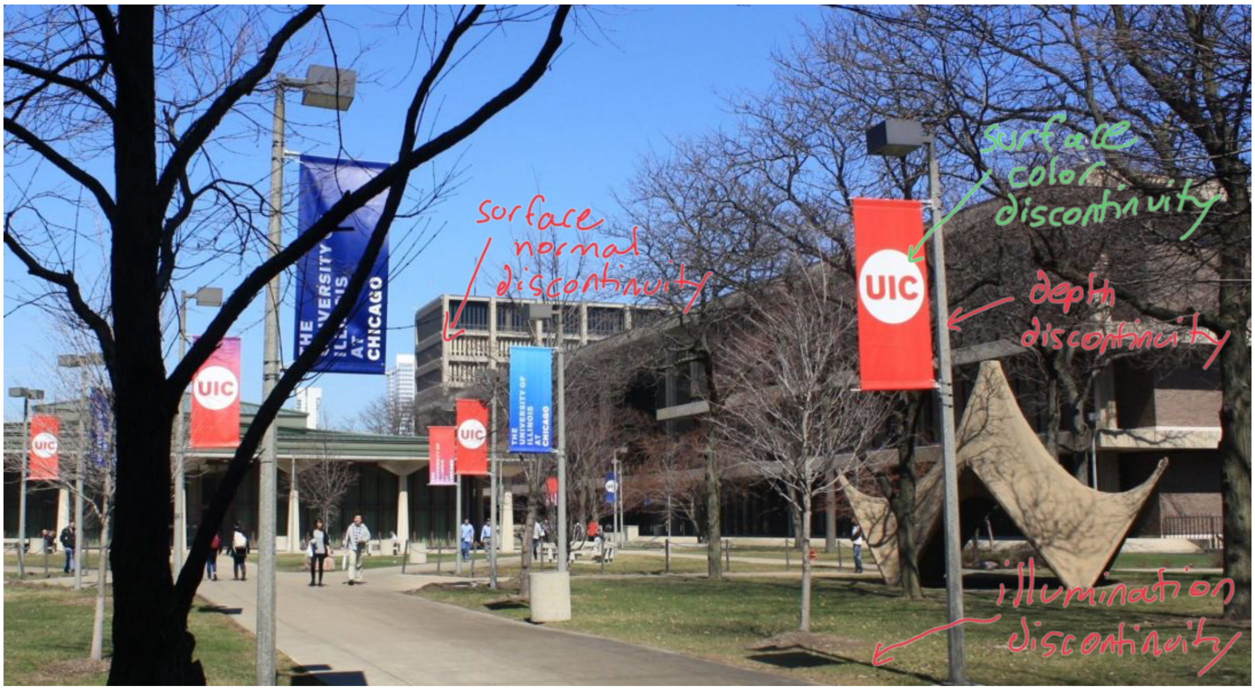
CS 415 Mini Project 2

1 Question Answering

Q1. (1) What is an edge in an image? (2) We learned four different factors that could cause edges in an image. For each of these factors, please find and label two segments of edges caused by it in the image below. (3) What is the relation between the direction of image gradients and that of edges. (20pt = 4pt + 12pt + 4pt)



Q2. Given a 5x5 image below, calculate the gradient of each pixel in the 3x3 central region (highlighted in blue) via finite difference. For the same region, calculate the gradient magnitude and gradient direction for each pixel. You can keep the square root and any trigonometric functions in the answer. (25 pt)

$$
\begin{array}{c|ccccc}
 & & & x & & \\
 & 1 & 1 & 1 & 1 & 1 \\
 & 1 & 1 & 0 & 2 & 1 \\
y & 1 & 2 & 2 & 1 & 1 \\
 & 1 & 2 & 1 & 0 & 1 \\
 & 1 & 1 & 1 & 1 & 1
\end{array}
$$

Q3. Why is the normal form of a line a better choice than the slope intercept form in Hough transform for line detection? (5 pt)

2. Programming

P1. Implement the hysteresis thresholding in Canny edge detector. You can use the partial code of Canny edge detector in our code tutorial as a base (available in Canvas) or build your own code from scratch. (15 pt)

- Use lena.png as a testing image. Select three different sets of low and high thresholds. For each set, calculate the three maps respectively corresponding to noise, weak edges, and strong edges.

P2 below is mandatory for graduate students and optional for undergraduate students. Undergraduate students will get the 10 points automatically. Undergraduate students completing P2 will get up to 3 extra points for this mini project.

P2. Implement the edge linking function in Canny edge detector. Weak edges that are connected to strong edges will be actual edges. Weak edges that are not connected to strong edges will be considered as noise and removed. (10 pt)

- Link edges in lena.png using the three sets of low and high thresholds selected in P1, respectively.

P3. In our code tutorial of Hough transform, the results include a lot of duplicate lines because it does not suppress detections that are not the local maxima. Please implement a function to address this issue. (15 pt)

- Test the effect of this function on paper.bmp and shape.bmp, respectively.

P4. Self-study the Canny edge detector and Hough transform implemented in OpenCV: cv2.Canny[1] and cv2.HoughLines[2]. (10pt)

- Apply cv2.Canny to lena.png with one set of parameters.
- Apply cv2.HoughLines to paper.bmp and shape.bmp with one set of parameters.

3 Submission

Please follow the instructions below for submission.

- You need to upload two files to Canvas: a PDF file and a .py file[3]. Do not compress them into a single ZIP file.
- The PDF file contains all your solutions to this homework. For Question Answering, you can either type answers or handwrite them and take a photo. For Programming, you need to include output of the program such as processed images.
- The .py file contains all your code for the programming problems.

---

[1] https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html
[2] https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html
[3] Using Jupyter Notebook and submitting a .ipynb file instead of a .py file are fine.

1) An edge is a rapid change in the intensity function of an image

2) see image above

3) image gradients are perpindicular to edges

Q2.

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

gradient

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

gradient magnitude

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y}, \frac{\partial f}{\partial x}\right)$$

gradient direction

$$\frac{\partial f}{\partial x}[x,y] \approx F[x+1,y] - F[x,y]$$

$$\frac{\partial f}{\partial y}[x,y] \approx F[x,y+1] - F[x,y]$$

$$\begin{bmatrix} -1 & 2 & -1 \\ 0 & -1 & 0 \\ -1 & -1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & -1 \\ 0 & -1 & -1 \\ -1 & 0 & 1 \end{bmatrix}$$

x gradient $\qquad$ y gradient

$$\begin{bmatrix} \sqrt{2} & 2\sqrt{2} & \sqrt{2} \\ 0 & \sqrt{2} & 1 \\ \sqrt{2} & 1 & \sqrt{2} \end{bmatrix} \begin{bmatrix} \tan^{-1}(-1) & \tan^{-1}(2) & \tan^{-1}(1) \\ 0 & \tan^{-1}(-1) & 0 \\ \tan^{-1}(-1) & \tan^{-1}(-1) & \tan^{-1}(1) \end{bmatrix} \begin{bmatrix} \tan^{-1}(1) & \tan^{-1}(2) & \tan^{-1}(-1) \\ 0 & \tan^{-1}(-1) & \tan^{-1}(-1) \\ \tan^{-1}(-1) & 0 & \tan^{-1}(1) \end{bmatrix}$$

gradient magnitude $\qquad$ xgradient direction $\qquad$ y gradient direction

Q3. Slope intercept form does not handle outliers properly. It can't handle vertical lines either. Normal form is better because you can easily find an edge by finding the intersection in parameter space