National Tsing Hua University
Department of Electrical Engineering

EE3662 Digital Signal Processing Laboratory, Fall 2018

Lab #7 Image Filtering and Corner Detection

Assigned on Oct 29, 2018

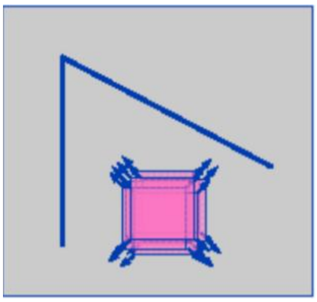
Due by Nov 5, 2018

Overview

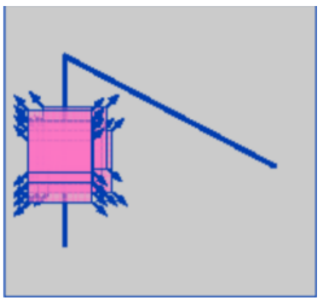
The goal of this homework is to use image filter to calculate image gradients. Then, we combine image gradients and apply Gaussian filter to compute second moment matrix at each pixel location. Next, we can compute the corner response function. Finally, we apply a threshold and non-maximum suppression to obtain distinctive corner locations.

(Left: original image. Right: image overlap with corners.)

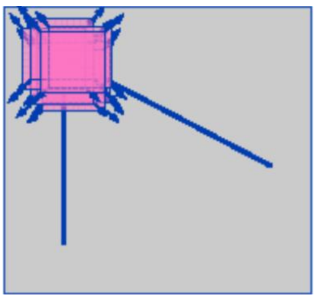
Ideas and Derivation



Flat region→no change in all direction



Edge→no change along the edge direction



Corner→significant change in all directions

The change of intensity for the shift (u,v) is given by

$$E(x,y) = \sum_{x,y} w(x,y)|I(x+u,y+v) - I(x,y)|^2,$$

where w(x,y) is window function. By the first order approximation of Taylor Series for 2D functions, we have

$$\mathrm{I(x+u,y+v)} = I(x,y) + uI_x(x,y) + vI_y(x,y),$$

then we can get the equation as follows.

$$E(x,y) = \sum_{x,y} w(x,y)\left(u^2 I_x^2 + 2uvI_xI_y + v^2 I_y^2\right).$$

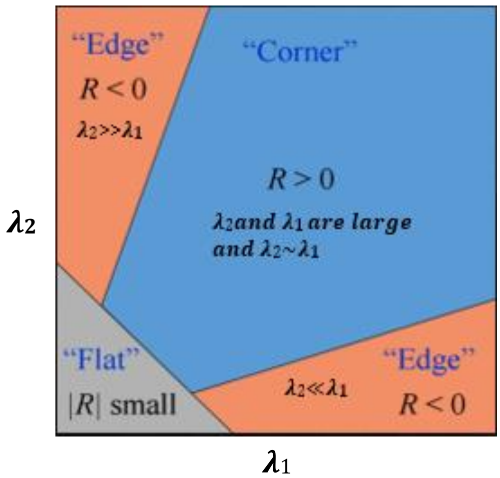
Rewrite it as matrix equation

$$E(x,y) = \sum_{x,y}(u \ \ v)w(x,y)\begin{pmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{pmatrix}\binom{u}{v} \text{ , where } A = \begin{pmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{pmatrix}.$$

Measurement of corner response is given by

$$\mathrm{R} = \det(\mathrm{A}) - \mathrm{k}(trace(A))^2 = \lambda_1\lambda_2 - \mathrm{k}(\lambda_1 + \lambda_2)^2.$$

We use eigenvalues of A to classify image points as below.



Details

I. Procedure

1. **Grey scale**
   Change the RGB to Grey scale by R*0.299+G*0.587+B*0.114.
2. **Get Image gradient**

Use horizontal and vertical gradient filter to get I_x, I_y, I_{xy}.

3. **Get Gaussian smoothed I_x, I_y, I_{xy}**

   Use Gaussian filter to get Gaussian smoothed I_x, I_y, I_{xy}.

4. **Calculate corner response R, and map R to 0~100**

   Calculate $\mathrm{R} = \det(\mathrm{A}) - \mathrm{k}(trace(A))^2$, where k is 0.04 empirically and trace(A) is the sum of diagonal component of A.

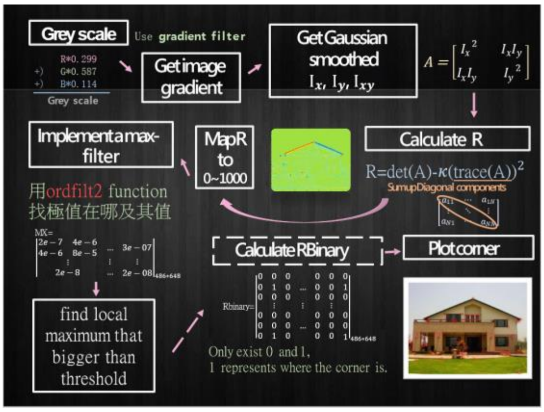
5. **Find local maximum larger than threshold**

   Use ordfilt2 function to find the local maximum.

6. **Calculate RBinary and plot corners**

   RBinary = $\begin{vmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & 0 & & 0 & 0 & 0 \\ & \vdots & & \vdots & & \vdots & \\ 0 & 0 & 0 & & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & & 0 & 0 & 1 \end{vmatrix}$, and 1 represents where the corner is.

Overall



II. In-class demo

1. **Implement MyHarrisCornerDetector.m.(50%)**
2. **Working corner image generation.(30%)**

III. Report

1. **Write up with several examples of images. (10%)**
2. **Why using Gaussian window to get Gaussian smoothed I_x, I_y, I_{xy}? (5%)**
3. **Why we need threshold in procedure5? (5%)**

IV. Deliverable and file organization

Directory	Filename	Description
LAB7/code/	MyHarrisCornerDectector.m	Matlab code
LAB7/results/	*.png	Your result
LAB7/report/	report.pdf	Your report

When you submit your file, please organize your files according to the above table and compress your files to LAB7_10xxxxxxx.zip in ZIP format.

P.S 10xxxxxxx is your student ID.

V. Reference

[1] http://www.cse.psu.edu/~rtc12/CSE486/lecture06.pdf