ECE/OPTI 532, Spring 2023

Homework 2 Assignment

Due Mon. Feb. 13

Write a computer program (e.g., MATLAB, Python, or C/C++) to implement the fictitious rms edge detector described below. (There are better methods for edge detection, but this is a good example for edge detection programming.)

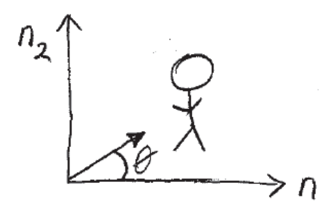
Let $f(r,c)$ be the input image. Replication padding or symmetric (mirror-image) padding is recommended, but not required. If you do not use array padding, then you must do bounds checking to ensure that you exclude out-of-bounds pixels from the calculations.

For each pixel, estimate the derivatives in the n_1 and n_2 directions as follows:

$$\begin{aligned} f_1(r,c) &= \text{right\_median} - \text{left\_median} \\ &= \text{median}\{f(r-1,c+1), f(r,c+1), f(r+1,c+1)\} \\ &\quad - \text{median}\{f(r-1,c-1), f(r,c-1), f(r+1,c-1)\} \end{aligned}$$

$$\begin{aligned} f_2(r,c) &= \text{top\_median} - \text{bottom\_median} \\ &= \text{median}\{f(r-1,c-1), f(r-1,c1), f(r-1,c+1)\} \\ &\quad - \text{median}\{f(r+1,c-1), f(r+1,c),\ \ f(r+1,c+1)\} \end{aligned}$$

where $\text{median}\{\dots\}$ is calculated by sorting the arguments in numerical order and selecting the middle value. The n_1 and n_2 directions are shown in the diagram:



Think about the minimum and maximum values that f_1 and f_2 can have, and then choose the correct data type for the f_1 and f_2 arrays.

Do <u>not</u> implement the optional gradient scaling factor (which is $1/2$).

Next, calculate the gradient magnitude and gradient direction:

$$\left|\vec{\nabla} f(r,c)\right| = \sqrt{f_1^2(r,c) + f_2^2(r,c)}\ ; \qquad \theta = \measuredangle \vec{\nabla} f(r,c) = \tan^{-1}\left(\frac{f_2(r,c)}{f_1(r,c)}\right)$$

Calculate the gradient direction so that all four quadrants are used. E.g., use the atan2( ) function.

Threshold the gradient magnitude to ensure that $\left|\vec{\nabla} f(r,c)\right| \geq$ thresh at the edge points.

Perform non-maximum suppression (NMS) so that pixels are excluded from the edge map if the gradient magnitude is not a local maximum in the gradient direction. Use the interpolation method for NMS. You may copy the source code provided in the lecture.

Run your program on the provided cman image, using gradient threshold = 35.

Submit the following:

1. Your source code in a file format that allows it to be compiled and executed.
2. An output file including the following:

   (a) Numerical values for f_1 for the following region:

   - If your array origin is at $(r, c) = (0,0)$, show the numerical output pixel values for
     $359 \leq r \leq 363,\ 149 \leq c \leq 153$ in a square array arrangement to facilitate grading.

   - If you are using MATLAB with array origin at $(r, c) = (1,1)$, show the numerical output pixel values for $360 \leq r \leq 364,\ 150 \leq c \leq 154$ in a square array arrangement to facilitate grading.

   (b) Repeat step (a) for f_2, $|\vec{\nabla} f|$, $\measuredangle \vec{\nabla} f(x, y)$ in <u>degrees</u>, and the edge map values.

   (c) The final edge map for the whole image, where the edge pixels are black and the non-edge pixels are white.