JHU 553.493/693, Spring 2020 – Homework #3

Corner detection, Laplacian of Gaussian, and Symmetries of signals

Due on Blackboard by the end of the day on Tuesday, March 24

Instructions: Write, on top of the first page of your assignment: Name (LAST, First), the HW# (in this case, "Homework #3"), and acknowledge others with whom you may have worked (just write "*Worked with ...*"). For the computational problems where you are asked to write computer code you may choose the programming language that you prefer, such as *Python* or *Matlab*. You do *not* have to type up your homework. If you handwrite it, please scan it, making sure it is readable. You should submit *one* PDF file on Blackboard (or at most two, if you choose to submit your code separately).

Problem 3.1 (Harris corner detector[1])**.** In class we introduced the Harris corner detector:

$$E_{(i,j)}(u,v) = \sum_{k=-\infty}^{\infty} \sum_{\ell=-\infty}^{\infty} w(k-i,\ell-j)\big[I(k,\ell) - I(k-u,\ell-v)\big]^2, \qquad (\star)$$

which measures the change of the image I around the location (i,j) due to a small displacement (u,v). Typically, w is a $(2K+1)\times(2K+1)$ filter of ones centered around the origin, but is could also be a rotationally invariant Gaussian mask. In class we showed[2] that we can approximate $(\star)$ with

$$E_{(i,j)}(u,v) = \begin{bmatrix} u & v \end{bmatrix} \cdot \overline{M}(i,j) \cdot \begin{bmatrix} u \\ v \end{bmatrix},$$

where

$$\overline{M}(i,j) = \sum_{k=-\infty}^{\infty} \sum_{\ell=-\infty}^{\infty} w(k-i,\ell-j) \begin{bmatrix} I_x^2(k,\ell) & I_x(k,\ell)I_y(k,\ell) \\ I_x(k,\ell)I_y(k,\ell) & I_y^2(k,\ell) \end{bmatrix}$$

is called the *local structure matrix*. We saw that we have evidence of the existence of a corner at (i,j) when both eigenvalues $\lambda_1(i,j)$ and $\lambda_2(i,j)$ of $\overline{M}(i,j)$ are large, and of comparable magnitude. This happens when the so-called *corner response function*

$$\boxed{Q_\alpha(i,j) = \det\big(\overline{M}(i,j)\big) - \alpha\big[\mathrm{Tr}\big(\overline{M}(i,j)\big)\big]^2}$$

is largest (typically the parameter α is chosen somewhere in the range between 0.04 and 0.15; this is discussed at the very end of the class notes).

(a) Choose $\alpha = 0.10$, and produce a contour plot of the corner response function with respect the eigenvalues (i.e., of $Q = \lambda_1\lambda_2 - \alpha(\lambda_1+\lambda_2)^2$). You should get a plot similar to Fig. 5 of Harris' paper[1]. (Feel free to use *Mathematica* , as it has one-line commands that allow you to achieve this.)

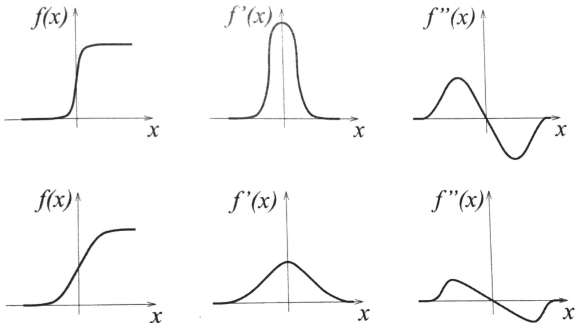
(b) Now download the images `image-polygons.gif`, `image-house.gif`, *and* other images of your own choice. Write code that computes the function $Q_\alpha(i,j)$, and plot, on top the image, markers (e.g. red crosses or bullets) that correspond to points within the image where Q_α is above a threshold t of your choice. *Remark:* There are several paramenters that you can play with. For my own code, I have used a 9×9 filter w of ones, $\alpha = 0.05$, and a threshold for Q_α of $t = 10^7$.

Note: Here is a silly problem that I encountered when solving the problem. In *Matlab*, the convention is that *for images* the x-axis runs vertically from top to bottom, while the y-axis runs horizontally from left to right. But, when plotting *points*, the x-axis runs horizontally from left to right, while y-axis runs vertically from bottom to top. So, for part **(b)**, if you want to plot both the *image* and the *points* that highlight the corners *on the same graph*, you will have to switch the coordinates of one of the two!

[1] C. Harris and M. Stephens: A Combined Corner and Edge Detector. In *Proceedings of the Fourth Alvey Vision Conference*, pages 147–151. University of Manchester, August 31—September 2, 1988.

[2] The fact that (u,v) is small allowed us to use a first-order Taylor expansion for the term in square brackets in $(\star)$.

Problem 3.2 (zero crossigns, Laplacian of Gaussian)**.**
In a previous homework assignment you detected edges in an image by thresholting the magnitude of the gradient of an image, or the positive part of its Laplacian. A more precise way to proceed is finding the *zero crossings* of the Laplacian (in one dimension, this corresponds to finding the inflection points of the graph of a finction—see the graphs on the right). The problem with this approach is that computing the second derivative can tremendously amplify additive noise that is already present in the image.



Marr and Hildreth[3] proposed an approach where one pre-filters the image I with a Gaussian mask g, then applies a Laplacian convolution with kernel h_L, and finally finds the zero crossings. The first two steps produce an image $J = h_L * (g * I)$, which by associativity of convolution can be written as $J = (h_L * g) * I$. The convolution kernel $h_{\text{LoG}} = h_L * g$ is called *Laplacian of Gaussian* (LoG).

(a) Consider the basic (3×3) Gaussian mask: $g = \dfrac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ and the Laplacian kernel $h_L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$. Compute the LoG kernel $h_{\text{LoG}} = h_L * g$. (Feel free to use `conv2` in *Matlab*.)

3B	2B	1B
4B		4A
1A	2A	3A

(b) In general, one fixes the standard deviation σ of the Gaussian mask (in the manner that we explored in Problem 2.5 of Homework#2) and then takes the convolution with a Laplacian kernel. Download the image `image-zebra.png`, then filter it with a *Laplacian of Gaussian* kernel, and find the resulting zero crossings (display them in an image). You should try out different values of the standard deviation (e.g. $\sigma = 2$, $\sigma = 5$, $\sigma = 10$, or others). In order to identify the zero crossing, you should also implement the *thresholding* algorightm that we discussed in class. Each pixel is surrounded by four pairs of pixels (shown above), and we determine that the central pixel is a zero crossing if the image J has opposite signs in *at least two* of the pairs, and the absolute of the difference of the values of J in such pairs is above a certain threshold t, that you choose (e.g. you can try $t = 1$ or similar values).

(c) We saw in class that the continuous analog of the LoG is the following function of 2 variables:

$$G_\sigma(x,y) = c_\sigma \ell_\sigma(x)\ell_\sigma(y), \text{ with } \ell_\sigma(r) = e^{-\frac{1}{2}\frac{r^2}{\sigma^2}} \implies \text{LoG}(x,y) = \nabla^2 G(x,y) = \frac{x^2+y^2-2\sigma^2}{\sigma^4} G_\sigma(x,y).$$

In the above expressions, $c_\sigma = \frac{1}{2\pi\sigma^2}$. An alternative to the Laplagian of Gaussian (LoG) is the so-called *Difference of Gaussians* (DoG), where $\nabla^2 G_\sigma(x,y)$ is approximated by the following difference,

$$d(x,y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{1}{2}\frac{x^2+y^2}{\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{1}{2}\frac{x^2+y^2}{\sigma_2^2}} \tag{1}$$

which depends on σ_1 and σ_2. To get an idea of its shape, plot its cross section $d(x,0)$ for $\sigma_1 = 2$ and $\sigma_2 = 1$. Show that $\nabla^2 G_\sigma(x,y)$ and (1) and have the same zero crossings when

$$\sigma^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 - \sigma_2^2} \ln\left(\frac{\sigma_1^2}{\sigma_2^2}\right) \tag{2}$$

Again: in order to find (2), it suffices to consider the cross sections at $y = 0$. Assuming that (2) holds, find the constant M in terms of σ_1 and σ_2 such that the function $\text{LoG}(x,y)$ and *Difference of Gaussians* function defined as $\text{DoG}(x,y) = M \cdot d(x,y)$ have the *same value* at the origin $(x,y) = (0,0)$.

(d) Let $\sigma_1 = 1.6$ and $\sigma_2 = 1$. Plot $z = \text{LoG}(x,0)$ and $z = \text{DoG}(x,0)$ the on the same graph, in the plane.

[3] D. Marr and E. Hildreth: Theory of Edge Detection. *Proceedings of the Royal Society*, B 107: 187–217, 1980.

Problem 3.3. (symmetries and decompositions) Consider the signals $s : \mathbb{R} \to \mathbb{C}$ and $g : \mathbb{R} \to \mathbb{C}$.

(a) Show that the linear combination $h(t) = as(t) + bg(t)$, $t \in \mathbb{R}$ of two *even* signals s and g is even for any $a, b \in \mathbb{R}$, and that the the linear combination of two odd signals is odd.

(b) If $s(t)$, $t \in \mathbb{R}$, is simultaneously even and odd, what signal is it? (Justify your answer precisely.)

(c) Suppose that the signal s is generic (not necessarily even or odd). The functions $s_{\mathrm{e}}(t) = \frac{1}{2}\big[s(t)+s(-t)\big]$ and $s_{\mathrm{o}}(t) = \frac{1}{2}\big[s(t)-s(-t)\big]$, defined in $\mathbb{R}$, are called, respectively, the *even part* and the *odd part* of s. Show that: (i) s_{e} is an even signal; (ii) s_{o} is odd; (iii) $s(t) = s_{\mathrm{e}}(t) + s_{\mathrm{o}}(\mathrm{t})$, for all t.

(d) Show that the decomposition $s(t) = s_{\mathrm{e}}(t) + s_{\mathrm{o}}(t)$, $t \in \mathbb{R}$ from part (c) is the *unique* decomposition of s into the sum of an even signal and an odd signal. *Hint:* assume $s(t) = g_{\mathrm{e}}(t) + g_{\mathrm{o}}(t)$, where $g_{\mathrm{e}}(t)$ is even and $g_{\mathrm{o}}(t)$ is odd; prove that $g_{\mathrm{e}}(t) = s_{\mathrm{e}}(t)$, and $g_{\mathrm{o}}(t) = s_{\mathrm{o}}(t)$, using parts (a) and (b).

(e) Now suppose that the signal $s : \mathbb{R} \to \mathbb{C}$ is differentiable. Show that (i) if s is even, then s' is odd; and that (ii) if s is odd, then s' is even. *Hint:* use the definition of even (or odd) signal, and the chain rule.

Remark: One could repeat parts (a)-(d) for other types of symmetries as well: e.g., for *real/imaginary* symmetries (by defining $s_{\mathrm{r}}(t) = \mathrm{Re}[s](t) = \frac{1}{2}\big[s(t) + s^*(t)\big]$ and $s_{\mathrm{i}}(t) = i\mathrm{Im}[s](t) = \frac{1}{2}\big[s(t) - s^*(t)\big]$), or *Hermitian/anti-Hermitian symmetries* (by defining $s_{\mathrm{h}}(t) = \frac{1}{2}\big[s(t)+s^*(-t)\big]$ and $s_{\mathrm{a}}(t) = \frac{1}{2}\big[s(t)-s^*(-t)\big]$).