Assignment 2: Image Representation

Harry Kim

Handed out: August 29, 2022
Due: 11:59pm, September 7, 2022
Handed in: 4:35pm, Month DD, 2022

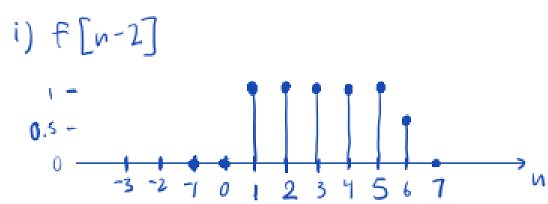
Important Notes:

- Feel free discuss the homework with the instructor or the TAs.
- Handwritten solutions will not be accepted.
- Turn in a PDF report and .m/.py files through Canvas as a compressed (.zip) file; turn in a hardcopy of PDF printout in class)

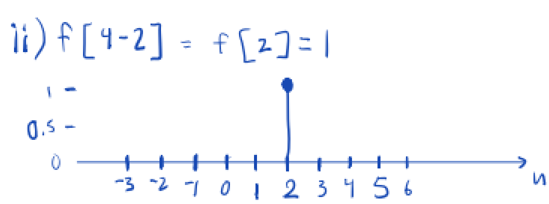
Question 1: 1D Convolution

(a) Sketch and label carefully each of the following signals.

(i) $f[n-2]$



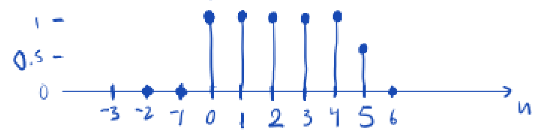
(ii) $f[4-2]$



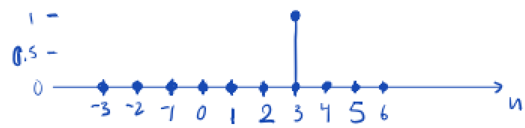
(iii) $f[n-1]\delta[n-3]$

iii) $f[n-1]\,\delta[n-3]$

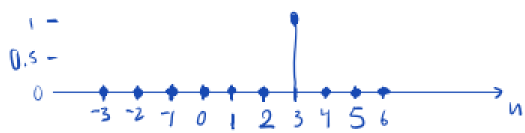
$f[n-1] = \{0, \dots, 0, \underset{\uparrow}{1}, 1, 1, 1, 1, 0.5, 0, \dots, 0\}$



$$\delta[n-3] = \begin{cases} 1, & n = 3 \\ 0, & n \neq 3 \end{cases}$$

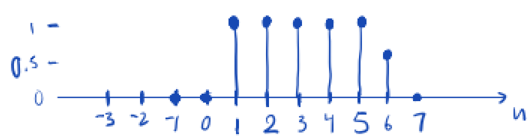


$f[n-1]\,\delta[n-3] = \{0, \dots, \underset{\uparrow}{0}, 0, 0, 1, 0, \dots, 0\}$

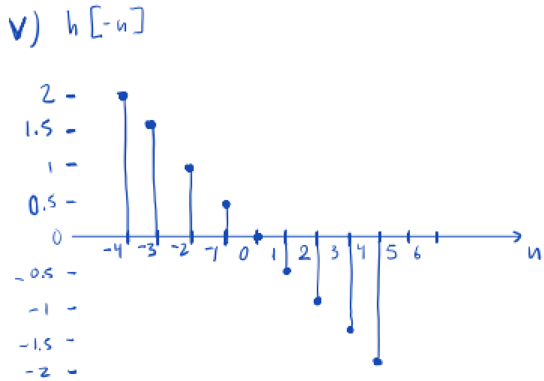


(iv) $f[n-2]$

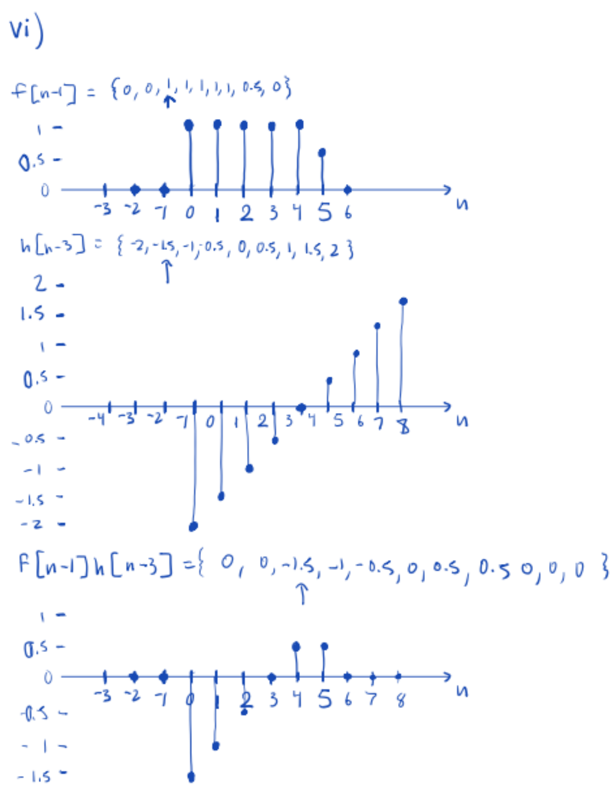
iv) $f[n-2]$



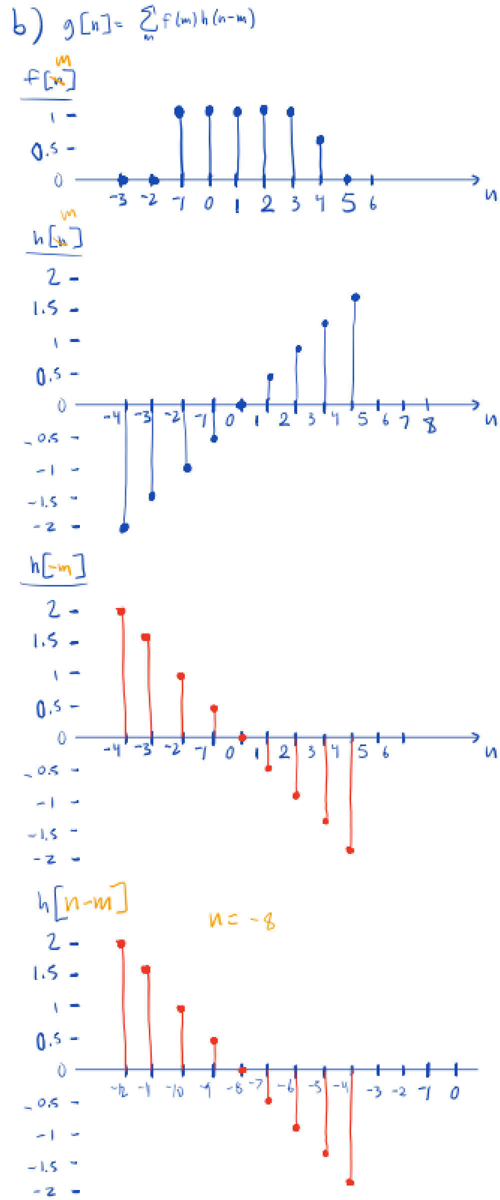
(v) $h[-n]$

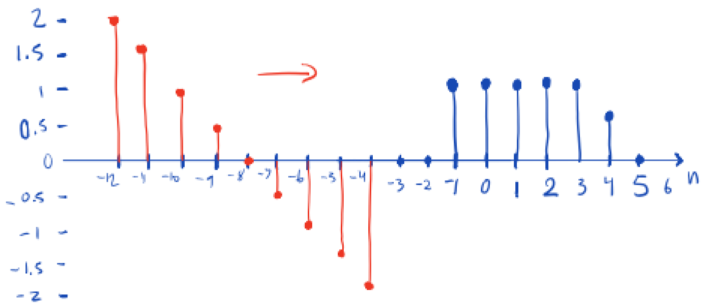


(vi) $f[n-1]h[n-3]$



(b) Sketch and label carefully the steps to convolve $f[n]$ and $h[n]$.





$$g[-8] = \sum_m f(m)\,h(-8-m) = 0 = 0$$

$$g[-7] = \sum_m f(m)\,h(-7-m) = (-2\cdot 0) = 0$$

$$g[-6] = \sum_m f(m)\,h(-6-m) = (-1.5\cdot 0) + (-2\cdot 0) = 0$$

$$g[-5] = \sum_m f(m)\,h(-5-m) = (-2\cdot 1) + (-1.5\cdot 0) + (-1\cdot 0) = -2$$

$$g[-4] = \sum_m f(m)\,h(-4-m) = (-2\cdot 1) + (-1.5\cdot 1) + (-1\cdot 0) + (0.5\cdot 0) = -3.5$$

$$g[-3] = \sum_m f(m)\,h(-3-m) = (-2\cdot 1) + (-1.5\cdot 1) + (1\cdot 1) + (-0.5\cdot 0) = -4.5$$

$$g[-2] = \sum_m f(m)\,h(-2-m) = (-2\cdot 1) + (-1.5\cdot 1) + (-1\cdot 1) + (-0.5\cdot 1) + (0\cdot 0) + (0.5\cdot 0)\ \ -5$$

$$g[-1] = \sum_m f(m)\,h(-1-m) = (-2\cdot 1) + (-1.5\cdot 1) + (-1\cdot 1) + (-0.5\cdot 1) + (0\cdot 1) + (0.5\cdot 0) + (1\cdot 0) = -5$$

$$g[0] = \sum_m f(m)\,h(0-m) = \underbrace{(-2\cdot 0.5) + (-1.5\cdot 1) + (-1\cdot 1) + (-0.5\cdot 1)} + \underbrace{(0\cdot 1) + (0.5\cdot 1) + (\cancel{1}\cdot 0)} + (\cancel{1.5}\cdot 0) =$$

$$-5 \qquad +1.5 = -3.5$$

$$g[1] = \sum_m f(m)h(1-m) = \underbrace{(-2\cdot 0) + (-1.5\cdot 0.5) + (-1\cdot 1) + (-0.5\cdot 1)}_{-2.25} + \underbrace{\cancel{(0\cdot 1)} + (0.5\cdot 1) + (1\cdot 1) + \cancel{(1.5\cdot 0)} + \cancel{(2\cdot 0)}}_{+\ 1.5} = -0.75$$

$$g[2] = \sum_m f(m)h(2-m) = \cancel{(-1.5\cdot 0)}^{0} + \underbrace{\underbrace{(-1\cdot 0.5)}_{-0.5} + \underbrace{(-0.5\cdot 1)}_{-0.5}}_{-1} + \cancel{(0\cdot 1)} + \underbrace{\underbrace{(0.5\cdot 1) + (1\cdot 1)}_{1.5} + (1.5\cdot 1) + \cancel{(2\cdot 0)}}_{3} = 2$$

$$g[3] = \sum f(m)h(3-m) = \cancel{(-1\cdot 0)} + \underbrace{(-0.5\cdot 0.5)}_{-0.25} + \cancel{(0\cdot 1)} + \underbrace{(0.5\cdot 1) + (1\cdot 1) + (1.5\cdot 1) + (2\cdot 1)}_{5} = 4.75$$

$$g[4] = \sum f(m)h(4-m) = \cancel{(-0.5\cdot 0)}^{0} + \cancel{(0\cdot 0.5)}^{0} + (0.5\cdot 1) + (1\cdot 1) + (1.5\cdot 1) + (2\cdot 1) = 5$$

$$g[5] = \sum f(m)h(5-m) = \cancel{(0\cdot 0)} + \underbrace{(0.5\cdot 0.5)}_{0.25} + \underbrace{(1\cdot 1)}_{1}\ \underbrace{(1.5\cdot 1)}_{1.5} + \underbrace{(2\cdot 1)}_{2} = 4.75$$

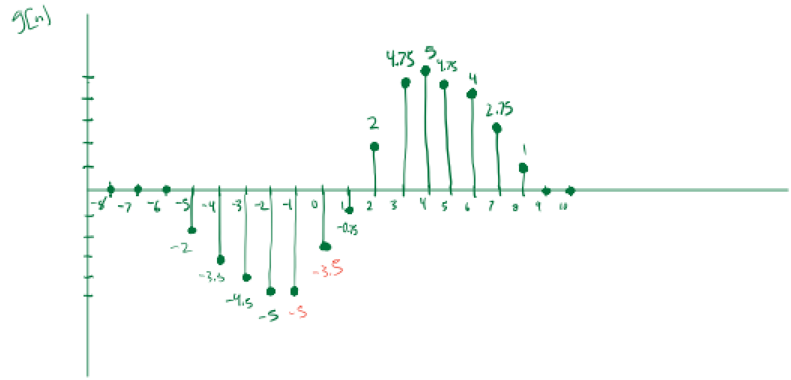
$$g[6] = \sum f(m)h(6-m) = \cancel{(0.5\cdot 0)} + \underbrace{(1\cdot 0.5)}_{0.5} + (1.5\cdot 1) + (2\cdot 1) = 4$$

$$g[7] = \sum f(m)h(7-m) = \cancel{(1\cdot 0)} + \underbrace{(1.5\cdot 0.5)}_{0.75} + \underbrace{(2\cdot 1)}_{+\ 2} = 2.75$$

$$g[8] = \sum f(m)h(8-m) = \cancel{(1.5\cdot 0)} + (2\cdot 0.5) = 1$$

$g[9] = \sum f(m)h(9-m) = (2 \cdot 0) = 0$

$g[10] = \sum f(m)h(10-m) = 0 = 0$



(c) Write a function that implements the 1D convolution operator given $f[n]$ and $h[n]$. Compare the output of your function with your answer in (b) and with the built-in convolution function in Matlab or Python.

My answer:
g: [0,0,-2,-3.5,-4.5,-5,-5,-3.5,-0.75,2,4.75,5,4.75,4,2.75,1,0]

Matlab answer:
g: [0,0,-2,-3.5,-4.5,-5,-5,-3.5,-0.75,2,4.75,5,4.75,4,2.75,1,0]

(d) Use the following pairs of $f[n]$ and $h[n]$sketch the steps to convolve them and comment on what you observe in the output when compared with input function $f[n]$.

d) $f[n] = [0,0,2,2,2,4,4,4,0,0]$ & $h[n] = [-1,1]$

$h[-n] = [1,-1]$

$g[n] = [0,0,-2,0,0,-2,0,0,4,0,0]$

$f[n] = [0,0,2,2,2,4,4,4,0,0]$ & $h[n] = [-1,2,1]$

$h[-n] = [1,2,-1]$

$g[n] = [0,0,-2,2,4,2,6,8,12,4,0,0]$

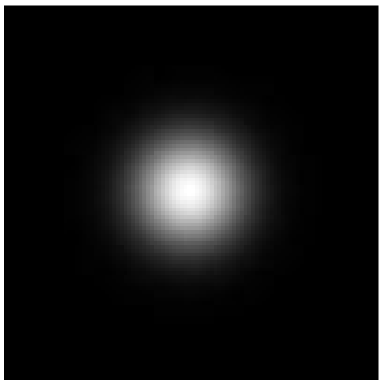
bell curve

more positive numbers

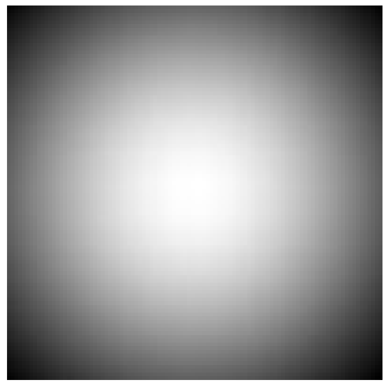
There are more positive numbers and there seems to be a Gaussian distribution happening.

Question 2: PSF & 2D convolution

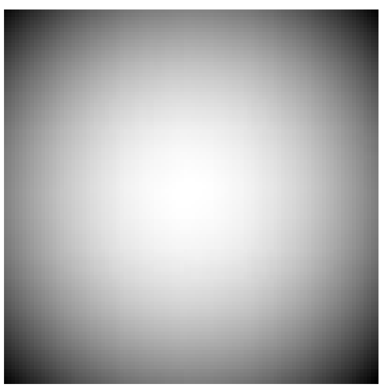
(a) Write a function that constructs a square image/matrix (i.e., number of rows = number of columns) with a Gaussian function (given below) in its center given the matrix's side length (i.e., number of rows or columns) and the standard deviation δ of the Gaussian. Display the constructed matrix as an image using varying values of the standard deviation. What happens when the standard deviation exceeds the third of the matrix's side length?



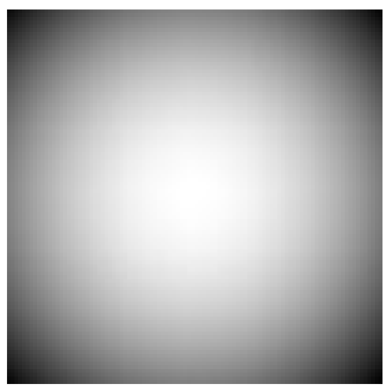
(a) Side Length: 100 Standard deviation: 10



(b) Side Length: 100 Standard deviation: 50



(c) Side Length: 100 Standard deviation: 300



(d) Side Length: 100 Standard deviation: 600

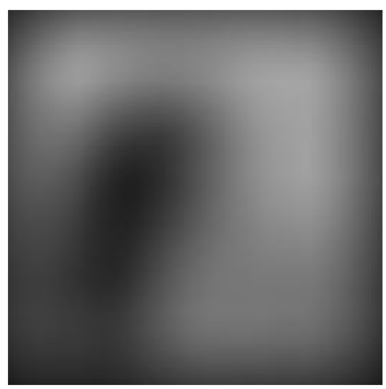
When the standard deviation exceeds the third of the matrix's side length, the "fade" of the image seems to begin to reach the sides of the image.

(b) Write a function that computes the convolution of a given 2D point spread function (PSF) with an object/scene function (a given image for the purpose of this question). You can use built-in convolution functions in Matlab or Python. For PSF, you can use a Gaussian function constructed by the function you wrote in (a). Experiment with different values of the PSF's standard deviation. Display the input image, the PSF, and the output image. Comment on the impact of different PSFs on the output image. Which PSF represents a better imaging system? Justify your answer.

(a) Side Length: 10
Standard deviation: 3

(b) Side Length: 10
Standard deviation: 50

(c) Side Length: 50
Standard deviation: 10



(d) Side Length: 100
Standard deviation: 40

The higher the side length used for the Gaussian image, the stronger the Gaussian effect seems to become. The standard deviation doesn't seem to significantly change the impact of the Gaussian effect as much as the side length does. The PSF with the lowest values represents a better imaging system because it smooths out the picture without distorting the image beyond recognition.

(c) Write a function that computes the PSF convolution with the object/scene function (a given image as in (b)) using a random percentage of the points in the object function. For PSF, you can use a Gaussian function constructed by the function you wrote in (a). Experiment with different random percentages by randomly selecting a percentage of the input image pixels and set them to zero. What happens as the percentage of "known" points in the input domain increases and decreases? Comment on the results. Display the input and output of each experiment.

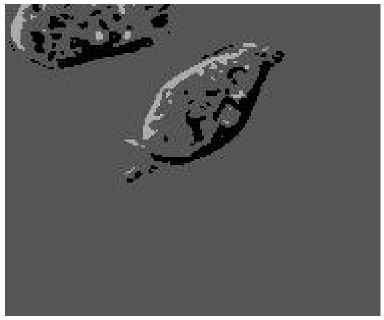
(a) Original image

(b) Known points: 50%

(c) Known points: 10%

As the percentage of known points decreases, the blurred picture becomes darker and vise-versa.

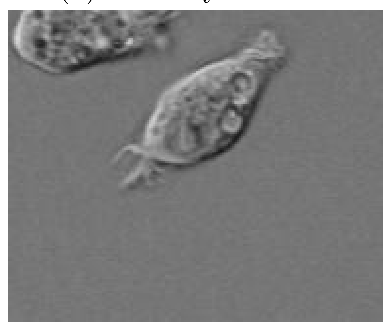
Question 3: Quantization



(a) Intensity levels: 4

(b) Intensity levels: 8

(c) Intensity levels: 64

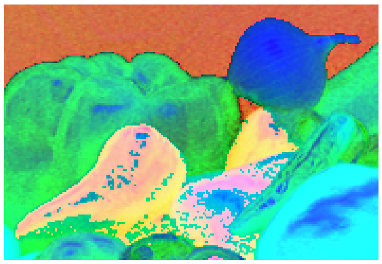


(d) Intensity levels: 128

(a) When you decrease the number of intensity levels, the intensities between the pixels becomes harsher and the image becomes more "blocky".

(a) Original image

(b) Intensity levels: 4



(c) Original image HSV

(d) Intensity levels: 4

(b) RGB is a reasonable color space for quantization because it quantifies each channel of color just like it would quantize a gray-scale image that only has one channel. When changing the image to an HSV color space and trying to quantize it, the process fails and the image returns as a black

Question 4: Extended Projects

(a) Convolve an image with the below 2D point spread functions (PSFs). Display the input and output of the convolution operator. What do these PSFs do to the images? What type of features do you observe in the output? Can you observe any variations in the output images, is one better than the other in some features?

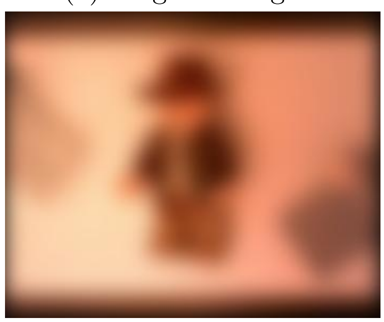
(a) Kernel from i (b) Kernel from ii (c) Kernel from iii

Using the kernel from i or ii gives the best results as it outlines the images vertically and horizontally respectively. Doing both, using kernel from iii, nearly zeros out the entire image making it really hard to see.

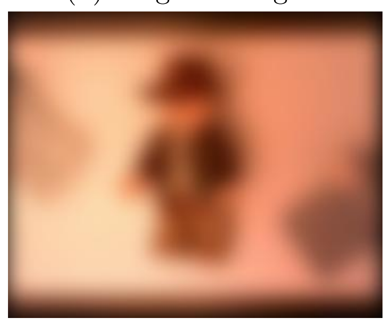
(b) Use a Gaussian PSF (e.g., use one from Question 2 (a)) and empirically show that the convolution operator is a linear operator by showing the results of convolving the Gaussian PSF with a linear combination of two images and comparing it (visually and numerically) with linearly combining the convolution results on two images. Comment on the results.

(a) Original image 1

(b) Original image 2



(c) Combined images then Gaussian



(d) Gaussian both images then combined

Visually they appear the same. Numerically, all values of the image that was combined then had the Gaussian function performed upon it is 0.0003 more than all the values of the image that had the Gaussian function performed upon each of the original images then combined. They are pretty much the same.