ECEN 642, Fall 2019

Texas A&M University

Electrical and Computer Engineering Department

Dr. Raffaella Righetti

Due: 09/27/2019 (before class)

Assignment #2

Gonzalez and Woods (4th edition), projects: 2.9, 3.2, 3.3, 3.4

Gonzalez and Woods (4th edition), problems: 3.7, 3.9, 3.13(a)

For 3.9, please show it by applying histogram equalization to "Fig0316(2)(2nd_from_top)".

For the assignments, you will need to use MATLAB. You can access it on campus through the open access lab (OAL) or remotely through the virtual open access lab (VOAL). The link below will guide you into configuring the Horizon client for VOAL remote connection step by step on your PC or mac:

https://tamu.service-now.com/tamu-selfservice/knowledge_detail.do?sysparm_document_key=kb_knowledge,6f7e0c6adbce5f84778ff5961d96199f#

After you connect to the server, you will see the MATLAB icon. If you don't, you can connect to your VOAL desktop (VOAL icon) and start MATLAB from there.

For projects from the 4th edition, photo copies of the project statements are provided. For projects from the 3rd edition, you can access them via the link: http://www.imageprocessingplace.com/DIP-3E/dip3e_student_projects.htm#02-04 (horizontal) directions. The output image should be of the same size as the input, and its background in the area vacated by the translated image is to be determined by **mode**. If **mode = 'black'**, the background should be black (this is the default if **mode** is not included in the function call). If **mode = 'white'**, the background should be white. Test your function by translating image **girl.tif** by half its height in the positive vertical direction and by one-fourth of its width in the positive horizontal direction. Test both values of **mode**.

(b)* Write a function **g = imageScaling4e(f,cx,cy)**, where **f** is a grayscale image and **cx** and **cy** are positive scaling factors along the x (vertical) and y (horizontal) directions, respectively (your function is not required to do reflections). As a test, scale the image **girl.tif** to one-half its height and twice its width, and display the result. (*Hint:* The code is simplified considerably by using an inverse transformation approach, as discussed in connection with Table 2.3.)

(c) Write a function **g = imageShear4e(f,sv,sh)**, where **sv** and **sh** are vertical and horizontal scalar shearing factors that can be any real numbers. Output **g** must be of the same size as **f**. The background of the sheared image should be black. (*Hint:* Combine the two shearing matrices into one composite shearing matrix.) Test your function by shearing image **girl.tif** using the following combination of values for **(sv, sh)**: (0.5,0), (0,−0.75), and (0.5,−0.75).

(d)* Write a function **g = imageRotate4e(f,theta,mode)** for performing image rotation (about the image center), where **theta** is the rotation angle in degrees (the default is zero degrees). A positive angle should produce counterclockwise rotation, as in Fig. 2.41. If **mode = 'crop'**, the rotated image should be cropped (about its center) to the same size as the input image. This should be the default. If **mode = 'full'**, the output image should be the smallest size capable of containing the full rotated input image for any angle. The background of the rotated image should be black. To test your function, rotate the image **girl.tif** by 45° using both the **'full'** and the **'crop'** modes. Display the results. (*Hint:* Keep in mind that the rotation equations in Table 2.3 are about the center of the coordinate system, not about the image center.)

2.9 Image histograms.

(a)* Write a function **h = imageHist4e(f,mode)** for computing the histogram of a 256-level grayscale image, **f**, whose intensities are nonnegative. If **mode = 'n'**, the histogram should be normalized (this is the default). Otherwise, if **mode = 'u'**, the histogram should be unnormalized.

(b) Generate and plot the histogram for the image **rose1024.tif** using **mode = 'n'**.

(c) What conclusions about image appearance can you draw from this histogram?

2.10 Statistical central moments.

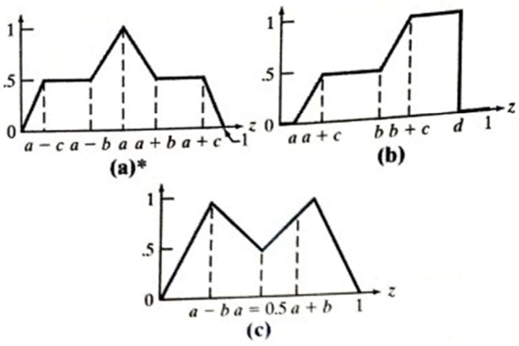
(a) Given a grayscale image, **f**, with intensities in the range [0, 1] or [0, 255], write a function **u = centralMoments4e(f,n)** that computes the mean, variance, and higher-order statistical central moments based on Eq. (2-96), with $p(z)$ being the histogram computed from the image. Output **u** is an n-dimensional vector. Instead of being a central moment, let **u(1)** represent be the mean value defined in Eq. (2-92). As usual, **u(2)** should be the variance, **u(3)** the third central moment, and so on. Because the number of samples in most practical images is relatively large, you may ignore the differences between population and sample estimates.

(b) Compute the mean, variance, third, and fourth moments of the image **rose1024.tif**. Display the image and the moments.

(c) Repeat for the image **angiography-live-image.tif**.

(d) Compute and plot the histograms of both images.

(e) Discuss how the values obtained in (b) and (c) relate to the shape of the histograms in (d).



3.61 Because the term z_0 is a constant in both μ_{red} and μ_3, show that Eq. (3-88) can be written as $Q_3(v) = \min\{\mu_{red}(z_0), \mu_{mat}(v)\}$.

3.62* What would be the effect of increasing the neighborhood size in the fuzzy filtering approach discussed at the end of Section 3.9? Explain. (You may use an example to support your answer).

3.63 You are employed to design a fuzzy, rule-based filtering system for reducing the effects of impulse noise on a noisy image with intensity values in the interval $[0, L-1]$. As in the filtering approach discussed at the end of Section 3.9, use only the differences d_2, d_4, d_6, and d_8 in a 3×3 neighborhood in order to simplify the problem. Let z_5 denote the intensity at the center of the neighborhood. The corresponding output intensity values should be $z_5' = z_5 + v$, where v is the output of your fuzzy system. That is, the output of your fuzzy system is a correction factor used to reduce the effect of a noise spike that may be present at the center of the 3×3 neighborhood. Assume that the noise spikes occur sufficiently apart so that you need not be concerned with multiple noise spikes being present in the same neighborhood. The spikes can be dark or light. Use triangular membership functions throughout.

(a)* Give a fuzzy system for this problem.

(b)* Specify the IF-THEN and ELSE rules.

(c) Specify the membership functions graphically, as in Fig. 3.77.

(d) Show a graphical representation of the rule set, as in Fig. 3.78.

(e) Give a summary diagram of your fuzzy system similar to the one in Fig. 3.72.

Projects

MATLAB solutions to the projects marked with an asterisk (*) are in the DIP4E Student Support Package (consult the book website: www.ImageProcessingPlace.com).

3.1* Write a function `g = imPad4e(f,r,c,padtype,loc)` for padding image `f` with `r` rows above and below the image, and `c` columns to the left and right. If `padtype = 'zeros'`, or is omitted from the argument, the function should implement zero padding. If `padtype = 'replicate'`, replicate padding, as defined in Section 3.5, should be used. If `loc` is specified as `loc = 'post'`, the function should behave as above, except that `r` rows are placed only below the image and `c` columns are placed only to the right of it.

3.2 Intensity transformation of grayscale images.

(a) Write a function `[g,map] = intXform4e(f,mode,param)` for transforming the intensities of an input 8-bit grayscale image `f`. The intensities of `f` (and output image `g`) are assumed to be in the range [0, 1] (use function `intScaling4e` from Chapter 2 in the body of `intXform4e` to make the conversion to [0, 1] automatic). The type of transformation performed is specified in parameter `mode`, a character string with values: `'negative'`, `'log'`, `'gamma'`, or `'external'`. The first two specifications implement Eqs. (3-3) and (3-4), (with $c = 1.0$). The third specification implements Eq. (3-5), in which case `param` is a scalar equal to the value of γ (use $c = 1.0$). Specifying `mode` as `'external'` means that the user is specifying the transformation function (e.g., for histogram equalization), whose values must be in the range [0, 1] and be provided as a 1-D array in `param`. On the output, `map` is the transformation function computed by `intXform4e` (or provided by the user if `'external'` was specified for `mode`).

(b) Read and display the image `spillway-dark.tif`. Apply a log transformation function to it. Display the result.

(c) See if you can improve on the result in (b) by using a gamma transformation. Display your best result.

(d) If an improvement is possible (or not possible), explain why.

3.3 Histogram equalization.

(a) Write a function **g = histEqual4e(f)** for performing histogram equalization on 8-bit input image **f**. (*Hint:* Use functions **imageHist4e** from Chapter 2, and **intXform4e** from Project 3.2 as part of function **histEqual4e**).

(b) Histogram equalize the image **spillway-dark.tif** and compare the result with the result in Project 3.2(c).

(c) Histogram equalize the image **hidden-horse.tif** and compare your result with Fig. 3.25(c).

3.4 Local histogram equalization.

(a)* Use functions **imageHist4e** from Chapter 2 and **intXform4e** from Projects 3.2 as the basis for writing a function **g = localHistEqual4e(f,m,n)** for performing local histogram equalization of 8-bit grayscale image **f** based on a neighborhood of size $m \times n$. Use replicate padding to pad the image border.

(b) Apply your function to the image **hidden-symbols.tif** using a 3×3 neighborhood, display the result, and compare it with Fig. 3.32(c). (The project image is of lower resolution than the image in the book, so your details will not be as sharp. However, the general characteristics of the two images should be the same.)

(c) Repeat (b) using a 7×7 neighborhood and explain any significant visual differences between the two results.

3.5 Two-dimensional convolution.

(a)* Write or obtain a function **g = twodConv4e(f,w)** for performing 2-D convolution of image **f** and kernel **w** in the language you are using for your projects. This function should use replicate padding by default. If you are using MATLAB, use function **conv2** as a starting point. This function uses zero-padding by default but you can get around this by pre-padding **f** with replicate padding and then stripping out the excess rows and columns due to the extra replicate padding before outputting **g**. Function **conv2** requires floating point inputs. Your function should by default scale the input to the range $[0, 1]$ using the default mode in project function **intScaling4e**. However, the function also has to have the capability of disabling this automatic scaling.

(b) Create an image of size 512×512 pixels that consists of a unit impulse at location $(256, 256)$ and zeros elsewhere. Use this image and a kernel of your choice to confirm that your function is indeed performing convolution. Display your results and explain what you did and why.

3.6 Lowpass filter kernel.

(a) Write a function **w = gaussKernel4e(m,sig,K)** that uses Eq. (3-55) to generate a normalized Gaussian lowpass kernel of size $m \times m$. If **K** is not included in the function call it should default to 1.

(b)* Filter the image from Project 3.5 with a Gaussian lowpass kernel large enough so that the maximum value of the filtered image is approximately 0.005 of the maximum value of the original image. Display the filtered image. (*Hint:* The image will appear black unless you scale it. Use the **'full'** mode in project function **intScaling4e** to scale the intensity values.)

3.7 Lowpass filtering.

(a) Read the image **testpattern1024.tif** and lowpass filter it using a Gaussian kernel large enough to blur the image so that the large letter "a" is barely readable, and the other letters are not.

(b)* Read the image **testpattern1024.tif**. Lowpass filter it using a Gaussian kernel of your specification so that, when thresholded, the filtered image contains only part of the large square on the top, right. (*Hint:* It is more intuitive to work with the negative of the original image.)

(c) Read the image **checkerboard1024-shaded.tif** and reproduce the results in Example 3.18, keeping in mind that the above image is of size 1024×1024 pixels, so the checkerboard squares are 64×64 pixels. (*Hint:* to obtain square, flat area, located 0.5 m away. The camera is equipped with a 35-mm lens. How many line pairs per mm will this camera be able to resolve? (*Hint:* Model the imaging process as in Fig. 2.3, with the focal length of the camera lens substituting for the focal length of the eye.)

2.9* Suppose that a given automated imaging application requires a minimum resolution of 5 line pairs per mm to be able to detect features of interest in objects viewed by the camera. The distance between the focal center of the camera lens and the area to be imaged is 1 m. The area being imaged is 0.5×0.5 m. You have available a 200 mm lens, and your job is to pick an appropriate CCD imaging chip. What is the minimum number of sensing elements and square size, $d \times d$, of the CCD chip that will meet the requirements of this application? (*Hint:* Model the imaging process as in Fig. 2.3, and assume for simplicity that the imaged area is square.)

2.10 An automobile manufacturer is automating the placement of certain components on the bumpers of a limited-edition line of sports cars. The components are color-coordinated, so the assembly robots need to know the color of each car in order to select the appropriate bumper component. Models come in only four colors: blue, green, red, and white. You are hired to propose a solution based on imaging. How would you solve the problem of determining the color of each car, keeping in mind that cost is the most important consideration in your choice of components?

2.11 A common measure of transmission for digital data is the *baud rate*, defined as symbols (bits in our case) per second. As a minimum, transmission is accomplished in packets consisting of a start bit, a byte (8 bits) of information, and a stop bit. Using these facts, answer the following:

(a)* How many seconds would it take to transmit a sequence of 500 images of size 1024×1024 pixels with 256 intensity levels using a 3 M-baud (10^6 bits/sec) baud modem? (This is a representative medium speed for a DSL (Digital Subscriber Line) residential line.

(b) What would the time be using a 30 G-baud (10^9 bits/sec) modem? (This is a representative medium speed for a commercial line.)

2.12* High-definition television (HDTV) generates images with 1125 horizontal TV lines interlaced (i.e., where every other line is "painted" on the screen in each of two fields, each field being 1/60th of a second in duration). The width-to-height aspect ratio of the images is 16:9. The fact that the number of horizontal lines is fixed determines the vertical resolution of the images. A company has designed a system that extracts digital images from HDTV video. The resolution of each horizontal line in their system is proportional to vertical resolution of HDTV, with the proportion being the width-to-height ratio of the images. Each pixel in the color image has 24 bits of intensity, 8 bits each for a red, a green, and a blue component image. These three "primary" images form a color image. How many bits would it take to store the images extracted from a two-hour HDTV movie?

2.13 When discussing linear indexing in Section 2.4, we arrived at the linear index in Eq. (2-14) by inspection. The same argument used there can be extended to a 3-D array with coordinates x, y, and z, and corresponding dimensions M, N, and P. The linear index for any (x, y, z) is

$$s = x + M(y + Nz)$$

Start with this expression and

(a)* Derive Eq. (2-15).

(b) Derive Eq. (2-16).

2.14* Suppose that a flat area with center at (x_0, y_0) is illuminated by a light source with intensity distribution

$$i(x,y) = Ke^{-[(x-x_0)^2 + (y-y_0)^2]}$$

Assume for simplicity that the reflectance of the area is constant and equal to 1.0, and let $K = 255$. If the intensity of the resulting image is quantized using k bits, and the eye can detect an abrupt change of eight intensity levels between adjacent pixels, what is the highest value of k that will cause visible false contouring?

2.15 Sketch the image in Problem 2.14 for $k = 2$.

2.16 Consider the two image subsets, S_1 and S_2 in the following figure. With reference to Section 2.5, and assuming that $V = \{1\}$, determine whether these two subsets are:

3.5 Do the following:

(a) Propose a method for extracting the bit planes of an image based on converting the value of its pixels to binary.

(b) Find all the bit planes of the following 4-bit image:

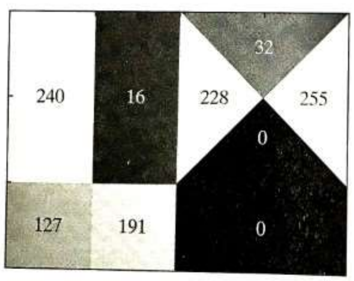
0	1	8	6
2	2	1	1
1	15	14	12
3	6	9	10

3.6 In general:

(a)* What effect would setting to zero the lower-order bit planes have on the histogram of an image?

(b) What would be the effect on the histogram if we set to zero the higher-order bit planes instead?

3.7 Obtain the unnormalized *and* the normalized histograms of the following 8-bit, $M \times N$ image. Give your histogram either in a table or a graph, labeling clearly the value and location of each histogram component in terms of M and N. Double-check your answer by making sure that the histogram components add to the correct value.



3.8 Explain why the discrete histogram equalization technique does not yield a flat histogram in general.

3.9 Suppose that a digital image is subjected to histogram equalization. Show that a second pass of histogram equalization (on the histogram-equalized image) will produce exactly the same result as the first pass.

3.10 Assuming continuous values, show by an example that it is possible to have a case in which the transformation function given in Eq. (3-11) satisfies conditions (a) and (b) discussed in Section 3.3, but its inverse may fail condition (a′).

3.11 Do the following:

(a) Show that the discrete transformation function given in Eq. (3-15) for histogram equalization satisfies conditions (a) and (b) stated at the beginning of Section 3.3.

(b)* Show that the inverse discrete transformation in Eq. (3-16) satisfies conditions (a′) and (b) in Section 3.3 *only if* none of the intensity levels r_k, $k = 0,1,2,\ldots,L-1$, are missing in the original image.

3.12 Two images, $f(x,y)$ and $g(x,y)$ have unnormalized histograms h_f and h_g. Give the conditions (on the values of the pixels in f and g) under which you can determine the histograms of images formed as follows:

(a)* $f(x,y) + g(x,y)$

(b) $f(x,y) - g(x,y)$

(c) $f(x,y) \times g(x,y)$

(d) $f(x,y) \div g(x,y)$

Show how the histograms would be formed in each case. The arithmetic operations are elementwise operations, as defined in Section 2.6.

3.13 Assume continuous intensity values, and suppose that the intensity values of an image have the PDF $p_r(r) = 2r/(L-1)^2$ for $0 \le r \le L-1$, and $p_r(r) = 0$ for other values of r.

(a)* Find the transformation function that will map the input intensity values, r, into values, s, of a histogram-equalized image.

(b)* Find the transformation function that (when applied to the histogram-equalized intensities, s) will produce an image whose intensity PDF is $p_z(z) = 3z^2/(L-1)^3$ for $0 \le z \le L-1$ and $p_z(z) = 0$ for other values of z.

(c) Express the transformation function from (b) directly in terms of r, the intensities of the input image.

3.14 An image with intensities in the range [0,1] has the PDF, $p_r(r)$, shown in the following figure. It is desired to transform the intensity levels of this image so that they will have the specified $p_z(z)$ shown in the figure. Assume continuous quantities