数字图像处理及应用 第2次作业

组号：XX（两位数字）　小组成员：（列出所有小组成员，成员姓名间用1个空格间隔）

Part I Exercises

Ex.1

(**a**) Give a continuous function for implementing the contrast stretching transformation shown in **Fig. 3.2(a)**. In addition to k, your function must include a parameter, E, for controlling the slope of the function as it transitions from low to high intensity values. Your function should be normalized so that its minimum and maximum values are 0 and 1, respectively.
(**b**) Sketch a family of transformations as a function of parameter E, for a fixed value $m = L/4$ where L is the number of intensity levels in the image.
(**c**) What is the smallest value of E that will make your function effectively perform as the function in **Fig. 3.2(b)**? In other words, your function does not have to be identical to **Fig. 3.2(b)**. It just has to yield the same result of producing a binary image. Assume that you are working with 8-bit images, and let $k = 128$. Let C denote the smallest positive number representable in the computer you are using.
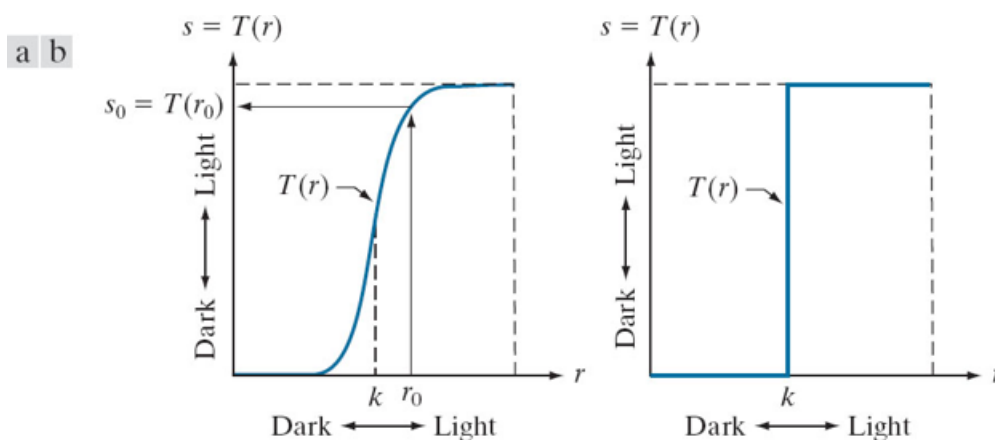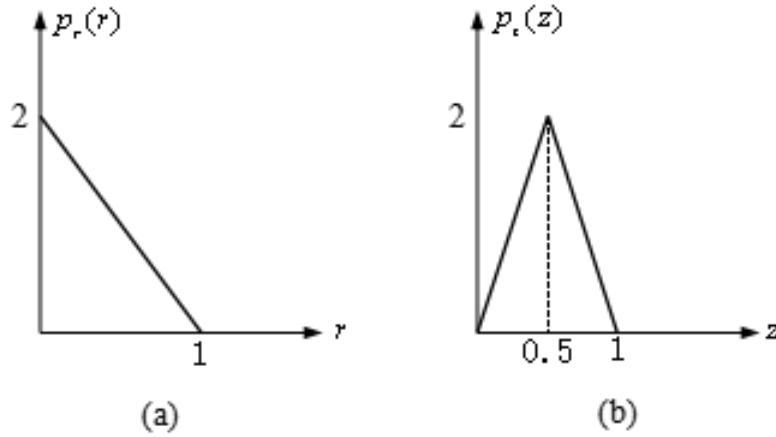

FIGURE 3.2 Intensity transformation functions

Answer:

Ex.2

An image with intensities in the range [0, 1] has the PDF $p_r(r)$ shown in the following diagram. It is desired to transform the intensity levels of this image so that they will have the specified $p_{z(z)}$ shown. Assume continuous quantities and find the transformation (in terms of r and z) that will accomplish this.

(a)   (b)

Answer:

Ex.3 The implementation of linear spatial filters requires moving the center of a mask throughout an image and, at each location, computing the sum of products of the mask coefficients with the corresponding pixels at that location (see **Section 3.4**). A lowpass filter can be implemented by setting all coefficients to 1, allowing use of a so-called box-filter or moving-average algorithm, which consists of updating only the part of the computation that changes from one location to the next.

(**a**) Formulate such an algorithm for an $n \times n$ filter, showing the nature of the computations involved and the scanning sequence used for moving the mask around the image.

(**b**) The ratio of the number of computations performed by a brute-force implementation to the number of computations performed by the box-filter algorithm is called the *computational advantage*. Obtain the computational advantage in this case and plot it as a function of n for $n > 1$. The $1/n^2$ scaling factor is common to both approaches, so you need not consider it in obtaining the computational advantage. Assume that the image has an outer border of zeros that is wide enough to allow you to ignore border effects in your analysis.

Answer:

Ex.4
(**a**) Suppose that you filter an image, $f(x, y)$, with a spatial filter mask, $w(x, y)$, using convolution, as defined in **Eq. (3.4-2)**, where the mask is smaller than the image in both spatial directions. Show the important property that, if the coefficients of the mask sum to zero, then the sum of all the elements in the resulting convolution array (filtered image) will be zero also (you may ignore computational inaccuracies). Also, you may assume that the border of the image has been padded with the appropriate number of zeros.

$$w(x, y) \star f(x, y) = \sum_{s=-a}^{s=a} \sum_{t=-b}^{t=b} w(x, y) f(x - s, y - t) \qquad (3.4\text{-}2)$$

here, the symbol ★ stands for convolution operation.

(**b**) Would the result to (a) be the same if the filtering is implemented using correlation, as defined in **Eq. (3.4-1)**?

$$w(x, y) \otimes f(x, y) = \sum_{s=-a}^{s=a} \sum_{t=-b}^{t=b} w(x, y) f(x + s, y + t) \tag{3.4-1}$$

here, the symbol $\otimes$ stands for correlation operation.

Answer:

Ex.5 Discuss the limiting effect of repeatedly filtering an image with a 3×3 lowpass filter kernel. You may ignore border effects.

Answer:

Ex.6 You are given the following kernel and image:

$$w = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \qquad f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{1}$$

(**a**) By inspection, find two kernels, w_1 and w_2 so that $w = w_1 \star w_2$. (★ stands for convolution operation)

(**b**) Compute $w_1 \star f$ using the minimum zero padding. Show the details of your computation when the kernel is centered at point (2, 3) (2nd row, 3rd col) of f and then show the full convolution.

(**c**) Compute the convolution of w_2 with the result from (**b**). Show the details of your computation when the kernel is centered at
point (3, 3) of the result from (b), and then show the full convolution. Compare with the result of $w \star f$.

Answer:

Ex.7 In a character recognition application, text pages are reduced to binary using a thresholding transformation function of the form shown in following figure.

$$s = T(r)$$
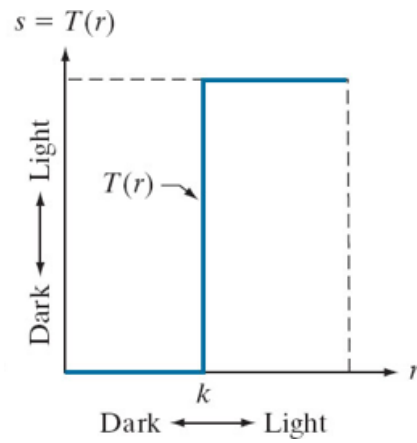
This is followed by a procedure that thins the characters until they become strings of binary 1's on a background of 0's. Due to noise, binarization and thinning result in broken strings of characters with gaps ranging from 1 to 3 pixels. One way to "repair" the gaps is to run a smoothing kernel over the binary image to blur it, and thus create bridges of nonzero pixels between gaps.

(**a**) Give the (odd) size of the smallest box kernel capable of performing this task.
(**b**) After bridging the gaps, the image is thresholded to convert it back to binary form. For your answer in (a), what is the minimum value of the threshold required to accomplish this, without causing the segments to break up again.

Answer:

Ex.8 In the original image used to generate the three blurred images shown, the vertical bars are 5 pixels wide, 100 pixels high, and their separation is 20 pixels. The image was blurred using square box kernels of sizes 23, 25, and 45 elements on the side, respectively. The vertical bars on the left, lower part of (**a**) and (**c**) are blurred, but a clear separation exists between them.
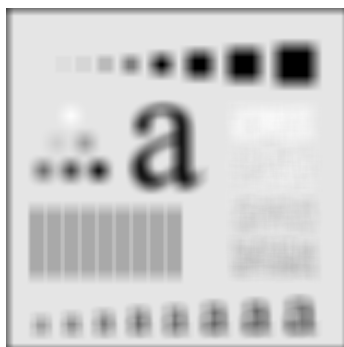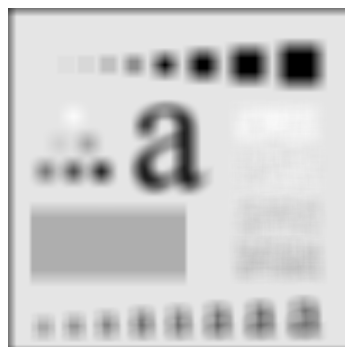
| (a) | (b) | (c) |

However, the bars have merged in image (**b**), despite the fact that the kernel used to generate this image is much smaller than the kernel that produced image (**c**). Explain the reason for this.

Answer:

Ex.9 Consider an application such as the one shown in **Fig. 3.34**, in which it is desired to eliminate objects smaller than those enclosed by a square of size $q \times q$ pixels. Suppose that we want to reduce the average intensity of those objects to one-tenth of their original average value. In this way, those objects will be closer to the intensity of the background and they can then be eliminated by thresholding. Give the (odd) size of the smallest averaging mask that will accomplish the desired reduction in average intensity in only one pass of the mask over the image.

FIGURE 3.34 Image from Hubble Space Telescope

Answer:

Ex.10 A CCD TV camera is used to perform a long-term study by observing the same area 24 hours a day, for 30 days. Digital images are captured and transmitted to a central location every 5 minutes. The illumination of the scene changes from natural daylight to artificial lighting. At no time is the scene without illumination, so it is always possible to obtain an image. Because the range of illumination is such that it is always in the linear operating range of the camera, it is decided not to employ any compensating mechanisms on the camera itself. Rather, it is decided to use image processing techniques to postprocess, and thus normalize, the images to the equivalent of constant illumination. Propose a method to do this. You are at liberty to use any method you wish, but state clearly all the assumptions you made in arriving at your design.

Answer:

Part II Programming

1. ***Generate two images with the size of*** 256×256 ***pixels*** which looks like the following images just containing two gray-levels. These two images are quite different, but their histograms are the same. Suppose that each image is blurred with a averaging mask.
(**a**) Would the histograms of the blurred images still be equal? Explain.
(**b**) If your answer is no, sketch the two histograms.
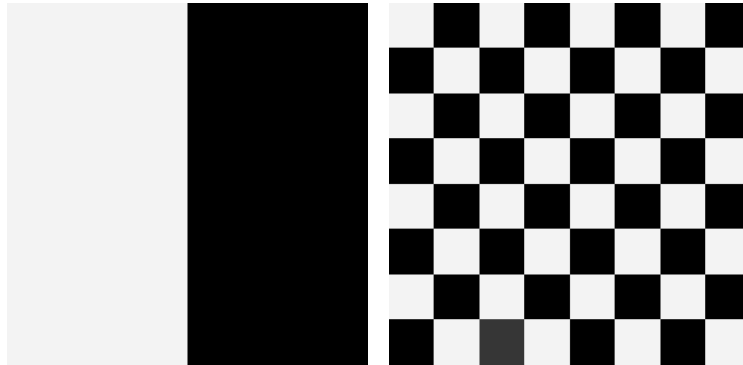
(*followed by **Matlab live Scripts** or **Jupyter Scripts** and running results*)

2. Implement the whole procedure described in **section 3.7** "Combining spatial enhancement methods."
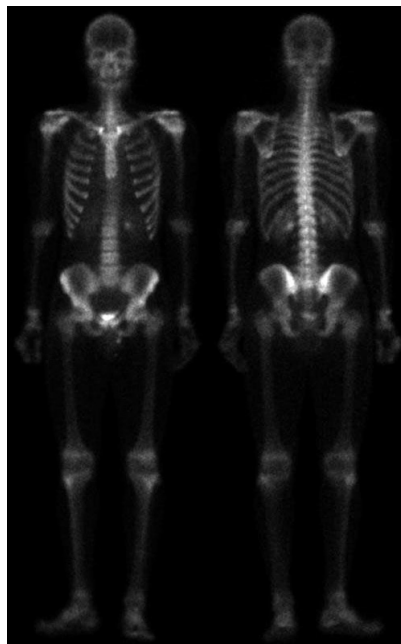

FIGURE 3.43(a) Image of whole body bone scan

(*followed by **Matlab live Scripts** or **Jupyter Scripts** and running results*)