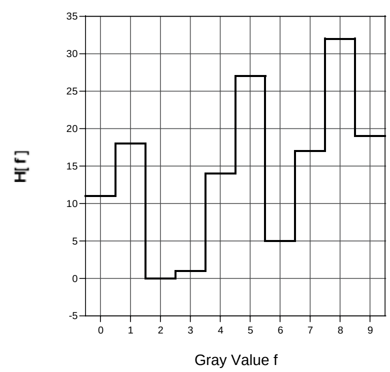
SIMG-712-90-20042 Solution Set #5

1. An image has been quantized to 10 gray levels per pixel and has the following histogram data. Generate the look-up table that will equalize the histogram.

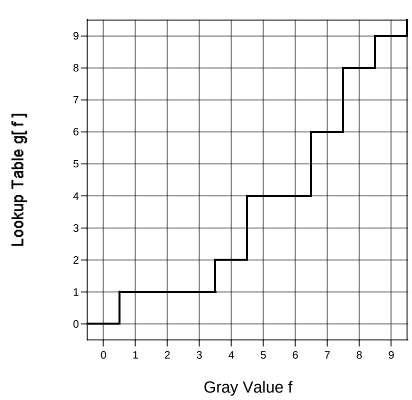
Gray Value	Population
0	11
1	18
2	0
3	1
4	14
5	27
6	5
7	17
8	32
9	19

(a) Plot the histogram:

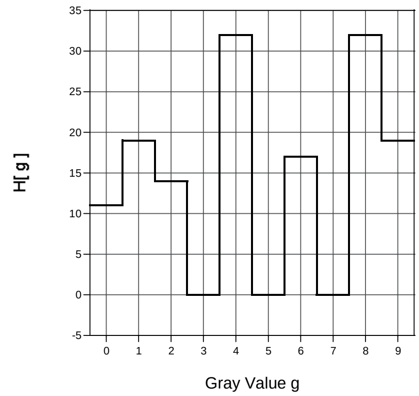
(b)



(c) Generate the look-up table that will equalize the histogram.



(d) Plot the equalized histogram:

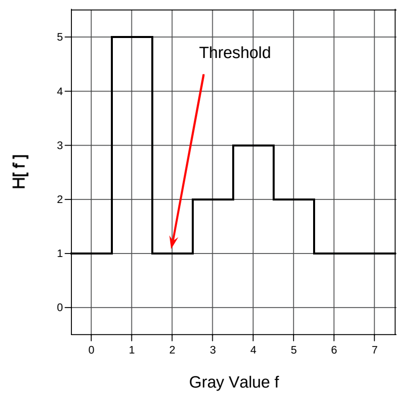


Note that the equalized histogram "spread out" the frequently occurring gray values and "crammed" some of the rarely occurring levels to the same level.

2. Consider the 4×4 quantized to 3 bits:

1	2	5	6
1	1	4	5
0	1	4	7
3	1	3	4

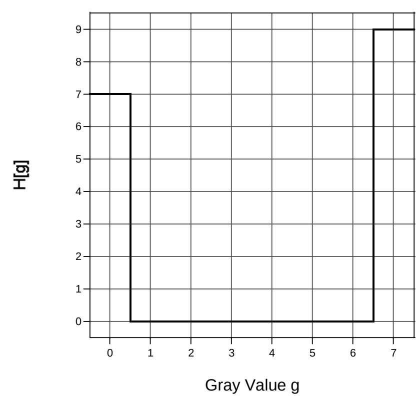
(a) Sketch the gray level image histogram.



(b) Characterize the histogram (one mode, bimodal, etc.). If bimodal, pick a threshold for binary thresholding transformation.

I'd say bimodal with threshold at 2.

(c) Sketch the histogram of the image after the thresholding transformation.



3. Design the impulse response of a filter that calculates an average along the radial direction specified by $\phi = +\frac{3\pi}{4}$ and calculates a second derivative along the orthogonal direction. What would this filter be useful for?

The averaging kernel along the specified direction is:

$$h_1[n,m] = \begin{array}{|c|c|c|}\hline \frac{1}{3} & 0 & 0 \\ \hline 0 & \frac{1}{3} & 0 \\ \hline 0 & 0 & \frac{1}{3} \\ \hline \end{array}$$

The second derivative kernel along the orthogonal direction is:

$$h_2[n,m] = \begin{array}{|c|c|c|}\hline 0 & 0 & +1 \\ \hline 0 & -2 & 0 \\ \hline +1 & 0 & 0 \\ \hline \end{array}$$

One reasonable option is to convolve these two together to make a 5×5 *kernel:*

$$\begin{aligned} h[n,m] &= h_1[n,m] * h_2[n,m] = \begin{array}{|c|c|c|}\hline +\frac{1}{3} & 0 & 0 \\ \hline 0 & +\frac{1}{3} & 0 \\ \hline 0 & 0 & +\frac{1}{3} \\ \hline \end{array} * \begin{array}{|c|c|c|}\hline 0 & 0 & +1 \\ \hline 0 & -2 & 0 \\ \hline +1 & 0 & 0 \\ \hline \end{array} \\ &= \begin{array}{|c|c|c|c|c|}\hline 0 & 0 & +\frac{1}{3} & 0 & 0 \\ \hline 0 & -\frac{2}{3} & 0 & +\frac{1}{3} & 0 \\ \hline +\frac{1}{3} & 0 & -\frac{2}{3} & 0 & +\frac{1}{3} \\ \hline 0 & +\frac{1}{3} & 0 & -\frac{2}{3} & 0 \\ \hline 0 & 0 & +\frac{1}{3} & 0 & 0 \\ \hline \end{array} \end{aligned}$$

Another reasonable option is to "compress" the 5×5 *kernel into a* 3×3 *by translating the extrema:*

$$h[n,m] = \begin{array}{|c|c|c|c|c|}\hline 0 & 0 & +\frac{1}{3} & 0 & 0 \\ \hline 0 & -\frac{2}{3} & 0 & +\frac{1}{3} & 0 \\ \hline +\frac{1}{3} & 0 & -\frac{2}{3} & 0 & +\frac{1}{3} \\ \hline 0 & +\frac{1}{3} & 0 & -\frac{2}{3} & 0 \\ \hline 0 & 0 & +\frac{1}{3} & 0 & 0 \\ \hline \end{array} \cong \begin{array}{|c|c|c|}\hline -\frac{2}{3} & +\frac{1}{3} & +\frac{1}{3} \\ \hline +\frac{1}{3} & -\frac{2}{3} & +\frac{1}{3} \\ \hline +\frac{1}{3} & +\frac{1}{3} & -\frac{2}{3} \\ \hline \end{array}$$

In either case, this kernel would be useful for locating thin (one-pixel-wide) lines that are oriented along the radial direction $\phi = +\frac{3\pi}{4}$.

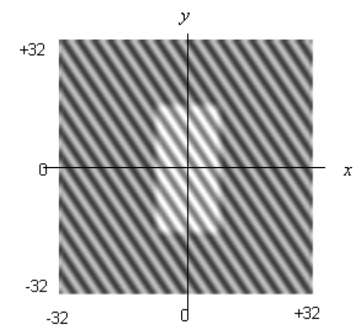
4. A rectangular object is to be imaged by a system that adds sinusoidal noise. The image to be recorded is:

$$g[x,y] = RECT\left[\frac{x}{16\,\text{mm}}, \frac{y}{32\,\text{mm}}\right] + \cos\left[2\pi\left(\frac{x}{4\,\text{mm}} + \frac{y}{6\,\text{mm}}\right)\right]$$

The image is to be sampled with $\Delta x = \Delta y = 1\,\text{mm}$.

(a) Design the transfer function of a filter that will "block" the sinusoidal noise while transmitting as much information about the rectangle as possible.

We clearly need to evaluate the Fourier spectrum of this function. and (as usual) it always is helpful to look at it first:



g[x,y]

(dimensions in mm*)*

The function is clearly real and even. Evaluating the spectrum may be facilitated by rewriting the cosine term via the trigonometric identity:

$$\cos[A+B] = \cos[A]\cos[B] - \sin[A]\sin[B]$$

$$\Longrightarrow \cos\left[2\pi\left(\frac{x}{4\,\text{mm}} + \frac{y}{6\,\text{mm}}\right)\right]$$

$$= \cos\left[2\pi\frac{x}{4\,\text{mm}}\right]\cdot\cos\left[2\pi\frac{y}{6\,\text{mm}}\right] - \sin\left[2\pi\frac{x}{4\,\text{mm}}\right]\cdot\sin\left[2\pi\frac{y}{6\,\text{mm}}\right]$$

Therefore we have the sum of three separable functions:

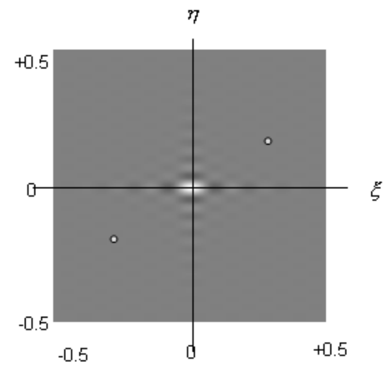
$$g[x,y] = RECT\left[\frac{x}{16\,\text{mm}}\right]\cdot RECT\left[\frac{y}{32\,\text{mm}}\right] + \cos\left[2\pi\frac{x}{4\,\text{mm}}\right]\cdot\cos\left[2\pi\frac{y}{6\,\text{mm}}\right]$$
$$- \sin\left[2\pi\frac{x}{4\,\text{mm}}\right]\cdot\sin\left[2\pi\frac{y}{6\,\text{mm}}\right]$$

$$G[\xi,\eta] = \left(16\cdot 32\,\text{mm}^2\right) SINC[16\,\text{mm}\cdot\xi]\cdot SINC[32\,\text{mm}\cdot\eta]$$
$$+\frac{1}{2}\left(\delta\left[\xi+\frac{1}{4\,\text{mm}}\right]+\delta\left[\xi-\frac{1}{4\,\text{mm}}\right]\right)\cdot\frac{1}{2}\left(\delta\left[\eta+\frac{1}{6\,\text{mm}}\right]+\delta\left[\eta-\frac{1}{6\,\text{mm}}\right]\right)$$
$$-\frac{1}{2i}\left(\delta\left[\xi+\frac{1}{4\,\text{mm}}\right]-\delta\left[\xi-\frac{1}{4\,\text{mm}}\right]\right)\cdot\frac{1}{2i}\left(\delta\left[\eta+\frac{1}{6\,\text{mm}}\right]-\delta\left[\eta-\frac{1}{6\,\text{mm}}\right]\right)$$

After expanding and collecting terms, we get the real-valued spectrum

$$G[\xi] = 512\,\text{mm}^2 SINC[16\,\text{mm}\cdot\xi, 32\,\text{mm}\cdot\eta] + \frac{1}{2}\delta\left[\xi+\frac{1}{4\,\text{mm}}, \eta+\frac{1}{6\,\text{mm}}\right]$$
$$+\frac{1}{2}\delta\left[\xi-\frac{1}{4\,\text{mm}}, \eta-\frac{1}{6\,\text{mm}}\right]$$

This is the sum of the "skinny" SINC function and an even pair of Dirac delta functions located at a radius of about 0.3 *cycles per* mm.



G[ξ,η]

(dimensions in cycles per mm*)*

Clearly we want to construct a transfer funtion that "blocks" the Dirac delta functions and passes the SINC function. You could use a circularly symmetric lowpass filter that passes frequencies out to 0.3 *cycles per* mm*:*

$$H_1(\rho) = CYL\left(\frac{\rho}{0.3\,\mathrm{mm}^{-1}}\right)$$

or a filter that passes frequencies near both axes and blocks those away from the axes (e.g.,

$$H_2[\xi,\eta] = RECT\left[\frac{\xi}{\frac{1}{2}\,\mathrm{mm}^{-1}-\varepsilon}\right]1[\eta] + 1[\xi]\,RECT\left[\frac{\eta}{\frac{1}{3}\,\mathrm{mm}^{-1}-\varepsilon}\right]$$

where ε *is some small positive number, or the transfer function could be a "notch" filter:*

$$H_3[\xi,\eta] = 1[\xi,\eta] - RECT\left[\frac{\xi}{\varepsilon},\frac{\eta}{\varepsilon}\right] * \left(\delta\left[\xi+\frac{1}{4\,\mathrm{mm}},\eta+\frac{1}{6\,\mathrm{mm}}\right]+\delta\left[\xi-\frac{1}{4\,\mathrm{mm}},\eta-\frac{1}{6\,\mathrm{mm}}\right]\right)$$

which will block just the delta functions.

(b) What changes do you expect to see in the box image after filtering?

In the lowpass filter case using $H_1(\rho)$*, the lowpass filter also will affect the spectrum of the box (the edges will become "fuzzy"). The effect of the filter on the box will be much less apparent in the second case since so little energy of the box appears away from the axes. Clearly the least effect will result in the third case; only the small amplitude of the box that occurs at the specific spatial frequency of the sinusoid will be blocked.*

5. Consider the result of a 7×7 uniform averaging filter to a digital image N times (ignore any effects at the edges of the image). Characterize the expected effect.

If the 7×7 uniform averaging kernel is $h[n,m]$, then the output is:

$$g[n,m] = (((f[n,m] * h_1[n,m]) * h_2[n,m]) * h[n,m] * \cdots * h_{N-1}[n,m]) * h_N[n,m]$$

where the subscripts only keep track of the order in the sequence of operations. The associativity of convolution allows us to write:

$$\begin{aligned} g[n,m] &= f[n,m] * ((((h_1[n,m] * h_2[n,m]) * h[n,m]) * \cdots * h_{N-1}[n,m]) * h_N[n,m]) \\ &= f[n,m] * (h_1[n,m] * h_2[n,m] * h[n,m] * \cdots * h_{N-1}[n,m] * h_N[n,m]) \\ &= f[n,m] * h'[n,m] \end{aligned}$$

Thus we can equivalently convolve the input image with the N-fold "autoconvolution" of the uniform averaging kernel. The convolution of two continuous rectangles is a triangle, the convolution of three rectangles is a quadratic function, of four rectangles is a cubic function, etc. The central-limit theorem tells us that the more rectangles convolved, the closer the result approaches a Gaussian. In the linear-math notes, we showed that:

$$(RECT[x])_1 * (RECT[x])_2 * \cdots * (RECT[x])_N \cong \frac{1}{\sqrt{\frac{N\pi}{6}}} GAUS\left[\frac{x}{\sqrt{\frac{N\pi}{6}}}\right]$$

Thus the result acts like a nonuniform averager that more heavily weigh amplitudes close to the pixel in question than those far from the pixel.

6. Consider this 8×8 three-bit image that contains an edge and some isolated noise points:

0	0	1	2	3	1	1	1
0	0	1	2	3	1	1	1
0	0	1	2	3	1	6	1
0	0	1	2	3	1	1	1
0	0	1	2	7	1	1	1
0	0	1	2	3	1	4	1
0	0	1	2	3	1	1	1
0	0	1	2	3	1	1	1

(a) Convolve this image with each of the following 3×3 convolution kernels.

(b) Deal with the edges in a manner you think appropriate, but specify what you are doing.

(c) Comment on the character of the output in each case.

1. *Original image data:*

0.0	0.0	1.0	2.0	3.0	1.0	1.0	1.0
0.0	0.0	1.0	2.0	3.0	1.0	1.0	1.0
0.0	0.0	1.0	2.0	3.0	1.0	6.0	1.0
0.0	0.0	1.0	2.0	3.0	1.0	1.0	1.0
0.0	0.0	1.0	2.0	7.0	1.0	1.0	1.0
0.0	0.0	1.0	2.0	3.0	1.0	4.0	1.0
0.0	0.0	1.0	2.0	3.0	1.0	1.0	1.0
0.0	0.0	1.0	2.0	3.0	1.0	1.0	1.0

1. $h_a[n, m] =$

-1	-1	-1
+1	+2	-1
+1	+1	-1

Convolution, assuming zeros off the edges:

0.0	1.0	4.0	4.0	1.0	-6.0	0.0	-1.0
0.0	1.0	4.0	4.0	1.0	-11.0	-5.0	-6.0
0.0	1.0	4.0	4.0	1.0	-1.0	10.0	-6.0
0.0	1.0	4.0	0.0	-3.0	-5.0	5.0	-6.0
0.0	1.0	4.0	8.0	9.0	-13.0	-3.0	-4.0
0.0	1.0	4.0	8.0	5.0	-7.0	6.0	-4.0
0.0	1.0	4.0	4.0	1.0	-3.0	3.0	-4.0
0.0	2.0	7.0	10.0	7.0	-1.0	3.0	1.0

2. $h_b[n,m] =$

-1	-1	-1
+1	+2	-1
+1	+1	-1

Rotate kernel by π *radians:*

0.0	-3.0	-4.0	-4.0	5.0	2.0	0.0	3.0
0.0	-3.0	-4.0	-4.0	5.0	-3.0	5.0	8.0
0.0	-3.0	-4.0	-4.0	5.0	-3.0	10.0	8.0
0.0	-3.0	-4.0	-8.0	9.0	1.0	-5.0	-2.0
0.0	-3.0	-4.0	-8.0	13.0	3.0	3.0	6.0
0.0	-3.0	-4.0	-8.0	1.0	-5.0	6.0	6.0
0.0	-3.0	-4.0	-4.0	5.0	-1.0	-3.0	0.0
0.0	-2.0	-3.0	-4.0	1.0	-1.0	-1.0	1.0

3. $h_c[n,m] = \frac{1}{9}$

+1	+1	+1
+1	+1	+1
+1	+1	+1

0.0	0.3	1.0	2.0	2.0	1.7	1.0	0.7
0.0	0.3	1.0	2.0	2.0	2.2	1.6	1.2
0.0	0.3	1.0	2.0	2.0	2.2	1.6	1.2
0.0	0.3	1.0	2.4	2.4	2.7	1.6	1.2
0.0	0.3	1.0	2.4	2.4	2.4	1.3	1.0
0.0	0.3	1.0	2.4	2.4	2.4	1.3	1.0
0.0	0.3	1.0	2.0	2.0	2.0	1.3	1.0
0.0	0.2	0.7	1.3	1.3	1.1	0.7	0.4

Before Quantizing

0.0	0.0	1.0	2.0	2.0	2.0	1.0	1.0
0.0	0.0	1.0	2.0	2.0	2.0	2.0	1.0
0.0	0.0	1.0	2.0	2.0	2.0	2.0	1.0
0.0	0.0	1.0	2.0	2.0	3.0	2.0	1.0
0.0	0.0	1.0	2.0	2.0	2.0	1.0	1.0
0.0	0.0	1.0	2.0	2.0	2.0	1.0	1.0
0.0	0.0	1.0	2.0	2.0	2.0	1.0	1.0
0.0	0.0	1.0	1.0	1.0	1.0	1.0	0.0

After quantizing by rounding to nearest integer

4. $h_d[n,m] =$

0	0	0
+1	-1	0
0	0	0

0.0	1.0	1.0	1.0	-2.0	0.0	0.0	-1.0
0.0	1.0	1.0	1.0	-2.0	0.0	0.0	-1.0
0.0	1.0	1.0	1.0	-2.0	5.0	-5.0	-1.0
0.0	1.0	1.0	1.0	-2.0	0.0	0.0	-1.0
0.0	1.0	1.0	5.0	-6.0	0.0	0.0	-1.0
0.0	1.0	1.0	1.0	-2.0	3.0	-3.0	-1.0
0.0	1.0	1.0	1.0	-2.0	0.0	0.0	-1.0
0.0	1.0	1.0	1.0	-2.0	0.0	0.0	-1.0

5. $h_e[n,m] =$

+1	+1	+1
+1	-8	+1
+1	+1	+1

0.0	3.0	0.0	0.0	-9.0	6.0	0.0	-3.0
0.0	3.0	0.0	0.0	-9.0	11.0	5.0	2.0
0.0	3.0	0.0	0.0	-9.0	11.0	-40.0	2.0
0.0	3.0	0.0	4.0	-5.0	15.0	5.0	2.0
0.0	3.0	0.0	4.0	-41.0	13.0	3.0	0.0
0.0	3.0	0.0	4.0	-5.0	13.0	-24.0	0.0
0.0	3.0	0.0	0.0	-9.0	9.0	3.0	0.0
0.0	2.0	-3.0	-6.0	-15.0	1.0	-3.0	-5.0

(d) Apply a 3×3 median filter to the digital image. Characterize the output relative to those in (c)

0.0	0.0	1.0	2.0	2.0	1.0	1.0	1.0
0.0	0.0	1.0	2.0	2.0	1.0	1.0	1.0
0.0	0.0	1.0	2.0	2.0	1.0	1.0	1.0
0.0	0.0	1.0	2.0	2.0	1.0	1.0	1.0
0.0	0.0	1.0	2.0	2.0	1.0	1.0	1.0
0.0	0.0	1.0	2.0	2.0	1.0	1.0	1.0
0.0	0.0	1.0	2.0	2.0	1.0	1.0	1.0
0.0	0.0	0.0	1.0	1.0	1.0	1.0	0.0

Compare to original image data:

0.0	0.0	1.0	2.0	3.0	1.0	1.0	1.0
0.0	0.0	1.0	2.0	3.0	1.0	1.0	1.0
0.0	0.0	1.0	2.0	3.0	1.0	6.0	1.0
0.0	0.0	1.0	2.0	3.0	1.0	1.0	1.0
0.0	0.0	1.0	2.0	7.0	1.0	1.0	1.0
0.0	0.0	1.0	2.0	3.0	1.0	4.0	1.0
0.0	0.0	1.0	2.0	3.0	1.0	1.0	1.0
0.0	0.0	1.0	2.0	3.0	1.0	1.0	1.0