

CS 445 – HW#2 Solutions

1. Text problem 3.1

- (a) General form: $s = T(r) = Ae^{-Kr^2}$. For the condition shown in the problem figure, $Ae^{-KL_0^2} = A/2$. Solving for K yields
- $$-KL_0^2 = \ln(0.5)$$

$$K = 0.693/L_0^2.$$

Then,

$$s = T(r) = Ae^{-\frac{0.693}{L_0^2}r^2}.$$

- (b) General form: $s = T(r) = B(1 - e^{-Kr^2})$. For the condition shown in the problem figure, $B(1 - e^{-KL_0^2}) = B/2$. The solution for K is the same as in (a) so

$$s = T(r) = B(1 - e^{-\frac{0.693}{L_0^2}r^2})$$

- (c) General form: $s = T(r) = (D - C)(1 - e^{-Kr^2}) + C$.

2. Text problem 3.2

(a) $s = T(r) = \frac{1}{1 + (m/r)^E}.$

- (b) See Fig. P3.2.

- (c) We want the value of s to be 0 for $r < m$, and s to be 1 for values of $r > m$. When $r = m$, $s = 1/2$. Then, because the values of r are integers, the behavior we want is

$$s = T(r) = \begin{cases} 0 & r \leq m-1 \\ 0.5 & r = m \\ 1 & r \geq m+1 \end{cases}$$

The question in the problem statement is to find the smallest value of E that will make the threshold behave as in the equation above. When $r = m$, we see from (a) that $s = 0.5$, regardless of the value of E . If C is the smallest positive number representable in the computer, and keeping in mind that s is positive, then any value of s less than $C/2$ will be called 0 by the computer. To find out the value of E for which this happens, simply solve the following equation for E , using the given value $m = 128$:

$$\frac{1}{1 + [m/(m-1)]^E} < C/2.$$

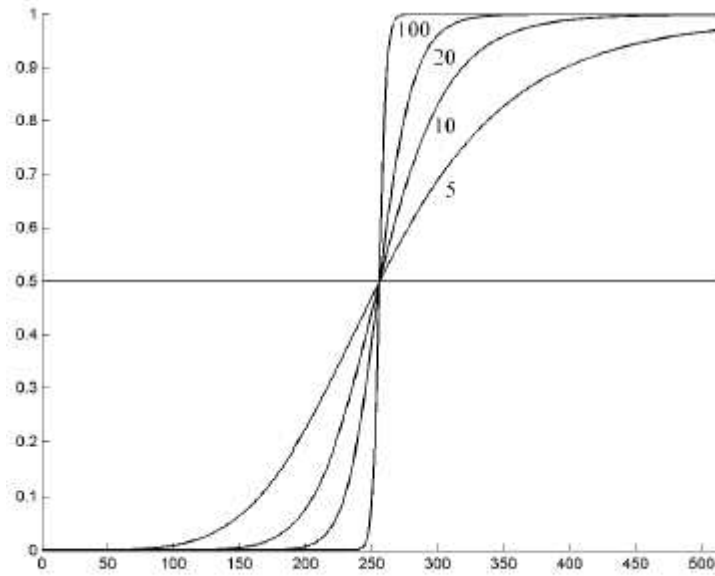


Figure P3.2

Because the function is symmetric about m , the resulting value of E will yield $s = 1$ for $r > m + 1$.

3. Text problem 3.4

- (a) The number of pixels having different gray level values would decrease, thus causing the number of components in the histogram to decrease. Since the number of pixels would not change, this would cause the height some of the remaining histogram peaks to increase in general. Typically, less variability in gray level values will reduce contrast.
- (b) The most visible effect would be significant darkening of the image. For example, dropping the highest bit would limit to 127 the brightest level in an 8bit image. Since the number of pixels would remain constant, the height of some of the histogram peaks would increase. The general shape of the histogram would now be taller and narrower, with no histogram components being located past 127.

4. Text problem 3.6

Let n be the total number of pixels and let n_{r_j} be the number of pixels in the input image with intensity value r_j . Then, the histogram equalization transformation is

$$s_k = T(r_k) = \sum_{j=0}^k n_{r_j} / n = \frac{1}{n} \sum_{j=0}^k n_{r_j}.$$

Since every pixel (and no others) with value r_k is mapped to value s_k , it follows that $n_{s_k} = n_{r_k}$. A second pass of histogram equalization would produce values v_k according to the transformation

$$v_k = T(s_k) = \frac{1}{n} \sum_{j=0}^k n_{s_j}.$$

But, $n_{s_j} = n_{r_j}$, so

$$v_k = T(s_k) = \frac{1}{n} \sum_{j=0}^k n_{r_j} = s_k$$

which shows that a second pass of histogram equalization would yield the same result as the first pass. We assumed negligible round-off errors.

5. Text problem 3.10

First, we obtain the histogram equalization transformation:

$$s = T(r) = \int_0^r p_r(w) dw = \int_0^r (-2w + 2) dw = -r^2 + 2r.$$

Next we find

$$v = G(z) = \int_0^z p_z(w) dw = \int_0^z 2w dw = z^2.$$

Finally,

$$z = G^{-1}(v) = \pm\sqrt{v}.$$

But only positive gray levels are allowed so $z = \sqrt{v}$. Then, we replace v with s , which in turn is $-r^2 + 2r$, and we have

$$z = \sqrt{-r^2 + 2r}.$$

6. Text problem 3.16

With reference to Section 3.4.2, when $i = 1$ (no averaging), we have

$$\bar{g}(1) = g_1 \text{ and } \sigma_{\bar{g}(1)}^2 = \sigma_\eta^2.$$

When $i = K$,

$$\bar{g}(K) = \frac{1}{K} \sum_{i=1}^K g_i \text{ and } \sigma_{\bar{g}(K)}^2 = \frac{1}{K} \sigma_\eta^2.$$

We want the ratio of $\sigma_{\bar{g}(K)}^2$ to $\sigma_{\bar{g}(1)}^2$ to be 1/10, 10

$$\frac{\sigma_{\bar{g}(K)}^2}{\sigma_{\bar{g}(1)}^2} = \frac{1}{10} = \frac{\frac{1}{K} \sigma_\eta^2}{\sigma_\eta^2}$$

from which we get $K = 10$. Since the images are generated at 30 frames/s, the stationary time required is 1/3 s.

7. Text problem 3.17

(a) Consider a 3×3 mask first. Since all the coefficients are 1 (we are ignoring the $1/9$ scale factor), the net effect of the lowpass filter operation is to add all the gray levels of pixels under the mask. Initially, it takes 8 additions to produce the response of the mask. However, when the mask moves one pixel location to the right, it picks up only one new column. The new response can be computed as

$$R_{new} = R_{old} - C_1 + C_3$$

Where C_1 is the sum of pixels under the first column of the mask before it was moved, and C_3 is the similar sum in the column it picked up after it moved. This is the basic box-filter or moving-average equation. For a

3 x 3 mask it takes 2 additions to get C_3 (C_1 was already computed). To this we add one subtraction and one addition to get R_{new} . Thus, a total of 4 arithmetic operations are needed to update the response after one more. This is a recursive procedure for moving from left to right along one row of the image. When we get to the end of a row, we move down one pixel (the nature of the computation is the same) and continue the scan in the opposite direction.

For a mask of size $n \times n$, $(n - 1)$ additions are needed to obtain C_3 , plus the single subtraction and addition needed to obtain R_{new} which gives a total of $(n + 1)$ arithmetic operations after each move. A brute-force implementation would require $n^2 - 1$ additions after each move.

(b) The computational advantage is

$$A = \frac{n^2 - 1}{n + 1} = \frac{(n + 1)(n - 1)}{(n + 1)} = n - 1.$$

The plot of A as a function of n is a simple linear function starting at $A = 1$ for $n = 2$.

8. Text problem 3.20

- (a) Numerically sort the n^2 values. The median is
 $\zeta = \lceil (n^2 + 1) / 2 \rceil$ -th largest value.
- (b) Once the values have been sorted one time, we simply delete the values in the trailing edge of the neighborhood and insert the values in the leading edge in the appropriate locations in the sorted array.

9. Text problem 3.22

From Fig. 3.35, the vertical bars are 5 pixels wide, 100 pixels high, and their separation is 20 pixels. The phenomenon in question is related to the horizontal separation between bars, so we can simplify the problem by considering a single scan line through the bars in the image. The key to answering this question lies in the fact that the distance (in pixels) between the onset of one bar and the onset of the next one (say, to its right) is 25 pixels. Consider the scan line shown in Fig. P3.22. Also shown is a cross section of a 25 x 25 mask. The response of the mask is the average of the pixels that it encompasses. We note that when the mask moves one pixel to the right, it loses one value of the vertical bar on the left, but it picks up an identical one on the right, so the response doesn't change. In fact, the number of pixels belonging to the vertical bars and contained within the mask does not change, regardless of where the mask is located (as long as it is contained within the bars, and not near the edges of the set of bars). The fact that the number of bar pixels under the mask does not change is due to the peculiar separation between bars and the width of the lines in relation to the 25-pixel width of the mask. This constant response is the reason no white gaps are seen in the image shown in the problem statement. Note that this constant response does not happen with the 23 x 23 or the 45 x 45 masks because they are not "synchronized" with the width of the bars and their separation.

10.

a)	Y_k	0	1/6	2/6	3/6	4/6	5/6	1
	$\frac{n_k}{n \times n}$	0.0237	0.0723	0.1350	0.3022	0.1865	0.1040	0.1763

b)	Y_k	0	1/6	2/6	3/6	4/6	5/6	1
	S_k	0.0237 \updownarrow 0	0.0960 \updownarrow 1/6	0.231 \updownarrow 1/6	0.5333 \updownarrow 3/6	0.7198 \updownarrow 4/6	0.8238 \updownarrow 5/6	1 \updownarrow 1

c) Equalized Histogram

T_k	0	1/6	3/6	4/6	5/6	1
Prob	0.0237	0.2073	0.3022	0.1865	0.1040	0.1763

11.

