Digital Image Processing

Second Edition

Instructor's Manual

Rafael C. Gonzalez
Richard E. Woods

(d) Finally, assuming that $c \neq 0$, the components of $h_{\text{div}}(u_k)$ are the same as those of h_f, but their locations will be at $u_k = r_k/c$. Thus, the spacing between components of $h_{\text{div}}(u_k)$ will be compressed by an amount equal to $1/c$.

The preceding solutions are applicable if image $f(x, y)$ also is constant. In this case the four histograms just discussed would each have only one component. Their location would be affected as described (a) through (c).

Problem 3.13

Using 10 bits (with one bit being the sign bit) allows numbers in the range -511 to 511. The process of repeated subtractions can be expressed as

$$\begin{aligned} d_K(x,y) &= a(x,y) - \sum_{k=1}^{K} b(x,y) \\ &= a(x,y) - K \times b(x,y) \end{aligned}$$

where K is the largest value such that $d_K(x,y)$ does not exceed -511 at any coordinates (x,y), at which time the subtraction process stops. We know nothing about the images, only that both have values ranging from 0 to 255. Therefore, all we can determine are the maximum and minimum number of times that the subtraction can be carried out and the possible range of gray-level values in each of these two situations.

Because it is given that $g(x,y)$ has at least one pixel valued 255, the maximum value that K can have before the subtraction exceeds -511 is 3. This condition occurs when, at some pair of coordinates (s,t), $a(s,t) = b(s,t) = 255$. In this case, the possible range of values in the difference image is -510 to 255. The latter condition can occur if, at some pair of coordinates (i,j), $a(i,j) = 255$ and $b(i,j) = 0$.

The minimum value that K will have is 2, which occurs when, at some pair of coordinates, $a(s,t) = 0$ and $b(s,t) = 255$. In this case, the possible range of values in the difference image again is -510 to 255. The latter condition can occur if, at some pair of coordinates (i,j), $a(i,j) = 255$ and $b(i,j) = 0$.

Problem 3.14

Let $g(x,y)$ denote the golden image, and let $f(x,y)$ denote any input image acquired during routine operation of the system. Change detection via subtraction is based on computing the simple difference $d(x,y) = g(x,y) - f(x,y)$. The resulting image

Similarly, K applications of the filter would give

$$G_K(u,v) = e^{-KD^2(u,v)/2D_0^2}F(u,v).$$

The inverse DFT of $G_K(u,v)$ would give the image resulting from K passes of the Gaussian filter. If K is "large enough," the Gaussian LPF will become a notch pass filter, passing only $F(0,0)$. We know that this term is equal to the average value of the image. So, there is a value of K after which the result of repeated lowpass filtering will simply produce a constant image. The value of all pixels on this image will be equal to the average value of the original image. Note that the answer applies even as K approaches infinity. In this case the filter will approach an impulse at the origin, and this would still give us $F(0,0)$ as the result of filtering.

(b) To guarantee the result in (a), K has to be chosen large enough so that the filter becomes a notch pass filter (at the origin) for all values of $D(u,v)$. Keeping in mind that increments of frequencies are in unit values, this means

$$H_K(u,v) = e^{-KD^2(u,v)/2D_0^2} = \begin{cases} 1 & \text{if } (u,v) = (0,0) \\ 0 & \text{Otherwise.} \end{cases}$$

Because u and v are integers, the conditions on the second line in this equation are satisfied for all $u > 1$ and/or $v > 1$. When $u = v = 0$, $D(u,v) = 0$, and $H_K(u,v) = 1$, as desired.

We want all values of the filter to be zero for all values of the distance from the origin that are greater than 0 (i.e., for values of u and/or v greater than 0). However, the filter is a Gaussian function, so its value is always greater than 0 for all finite values of $D(u,v)$. But, we are dealing with digital numbers, which will be designated as zero whenever the value of the filter is less than $\frac{1}{2}$ the smallest positive number representable in the computer being used. Assume this number to be k_{min} (don't confuse the meaning of this k with K, which is the number of applications of the filter). So, values of K for which for which the filter function is greater than $0.5 \times k_{\text{min}}$ will suffice. That is, we want the minimum value of K for which

$$e^{-KD^2(u,v)/2D_0^2} < 0.5k_{\min}$$

or

$$\begin{aligned} K &> -\frac{\ln(0.5k_{\min})}{D^2(u,v)/2D_0^2} \\ &> -\frac{2D_0^2\,\ln(0.5k_{\min})}{D^2(u,v)}. \end{aligned}$$

As noted above, we want this equation for hold for all values of $D^2(u,v) > 0$. Since the exponential decreases as a function of increasing distance from the origin, we choose

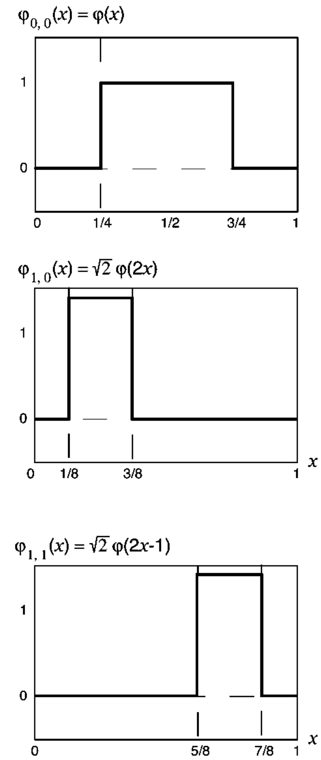

Figure P7.11

Problem 7.12

Substituting $j = 3$ into Eq. (7.2-13) we get

$$\begin{aligned} V_3 &= \overline{\underset{k}{Span}\{\varphi_{3,k}(x)\}} \\ &= \overline{\underset{k}{Span}\{2^{3/2}\varphi(2^3x-k)\}} \\ &= \overline{\underset{k}{Span}\{2\sqrt{2}\varphi(8x-k)\}}. \end{aligned}$$

Using the Haar scaling function in Eq. (7.2-14) we get the results shown in Fig. P7.12.

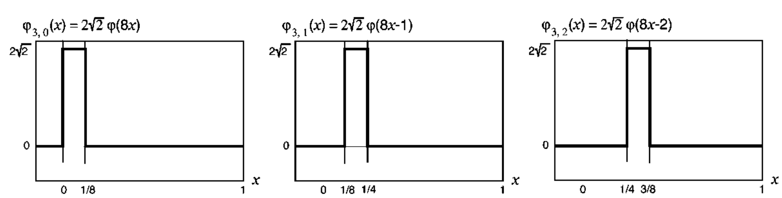

Figure P7.12

[1 1 1] would yield the following responses when centered at the pixels with values b, e, and h, respectively: $(a+b+c)$, $(d+e+f)$, and $(g+h+i)$. Next, we pass the mask

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

through these results. When this mask is centered at the pixel with value e, its response will be $[(a+b+c)+(d+e+f)+(g+h+i)]$, which is the same as the result produced by the 3×3 smoothing mask.

Returning now to problem at hand, when the G_x Sobel mask is centered at the pixel with value e, its response is $G_x = (g+2h+i)-(a+2b+c)$. If we pass the one-dimensional differencing mask

$$\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

through the image, its response when its center is at the pixels with values d, e, and f, respectively, would be: $(g-a)$, $(h-b)$, and $(i-c)$. Next we apply the smoothing mask [1 2 1] to these results. When the mask is centered at the pixel with value e, its response would be $[(g-a)+2(h-b)+(i-c)]$ which is $[(g+2h+i)-(a+2b+c)]$. This is the same as the response of the 3×3 Sobel mask for G_x. The process to show equivalence for G_y is basically the same. Note, however, that the directions of the one-dimensional masks would be reversed in the sense that the differencing mask would be a column mask and the smoothing mask would be a row mask.

1	1	1
1	1	1
1	1	1

Smoothing mask (scaled by 1/9).

a	b	c
d	e	f
g	h	i

Subimage area under the mask at any one time.

Figure P10.8

then Fig. P12.20 shows the only way that A can separate from C before separating from B. This, however, violates (2), which means that the condition $k_{ac} < \min[k_{ab}, k_{bc}]$ is violated (we can also see this in the figure by noting that $k_{ac} = k_{bc}$ which, since $k_{bc} < k_{ab}$, violates the condition). We use a similar argument to show that if (2) holds then (1) is violated. Thus, we conclude that it is impossible for the condition $k_{ac} < \min[k_{ab}, k_{bc}]$ to hold, thus proving that $k_{ac} \geq \min[k_{ab}, k_{bc}]$ or, equivalently, that $D(A, C) \leq \max[D(A, B), D(B, C)]$.

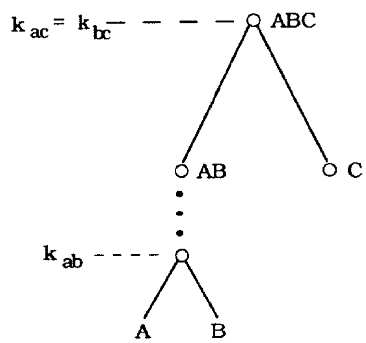

Figure P12.20

Problem 12.21

$Q = 0$ implies that $\max(|A|, |B|) = M$. Suppose that $|A| > |B|$. Then, it must follow that $|A| = M$ and, therefore, that $M > |B|$. But M is obtained by matching A and B, so it must be bounded by $M \leq \min(|A|, |B|)$. Since we have stipulated that $|A| > |B|$, the condition $M \leq \min(|A|, |B|)$ implies $M \leq |B|$. But this contradicts the above result, so the only way for $\max(|A|, |B|) = M$ to hold is if $|A| = |B|$. This, in turn, implies that A and B must be identical strings $(A \equiv B)$ because $|A| = |B| = M$ means that all symbols of A and B match. The converse result that if $A \equiv B$ then $Q = 0$ follows directly from the definition of Q.

Problem 12.22

(a) An automaton capable of accepting *only* strings of the form $ab^n a \geq 1$, shown in Fig. P12.22, is given by

$$A_f = (Q, \Sigma, \delta, q_0, F),$$