Solutions to Optional Homework (Lecture 13)

Problem 10.32

We know from Problem 10.32 that if the initial threshold value is selected between the minimum and maximum intensity values, the algorithm is guaranteed to converge. However, nothing is stated in the proof about the final value being independent of the initial value. It stands to reason to conclude that the threshold value at convergence is influenced by the shape of the image histogram. Consequently, we may expect that in general the final value will be influenced by the starting value. We can show that this is true using a simple example. Let $f = [0\,1\,2; 3\,4\,5; 6\,7\,8]/8$. If we choose an initial value of 0.1, the algorithm will converge to $T = 0.4688$. If we start with $T = 0.8$, the algorithm converges to $T = 0.5313$.

Problem 10.33

(a) For a uniform histogram, we can view the intensity levels as points of unit mass along the intensity axis of the histogram. Any values $m_1(k)$ and $m_2(k)$ are the means of the two groups of intensity values G_1 and G_2. Because the histogram is uniform, these are the centers of mass of G_1 and G_2. We know from the solution of Problem 10.30 that if T starts moving to the right, it will always move in that direction, or stop. The same holds true for movement to the left. Now, assume that $T(k)$ has arrived at the center of mass (average intensity). Because all points have equal "weight" (remember the histogram is uniform), if $T(k+1)$ moves to the right G_2 will pick up, say, Q new points. But G_1 will lose the same number of points, so the sum $m_1 + m_2$ will be the same and the algorithm will stop.

(b) The proof is similar to (a) because the modes are identical. When the algorithm arrives at the point between the two means, further motion of the threshold toward one or the other mean will cause one group to pick up and the other to lose the same "mass". Thus, their sum will be the same and the algorithm will stop.

Problem 10.41

The means are at 60 and 170, and the standard deviation of the noise is 10 intensity levels. A range of $\pm 3\sigma$ about 60 gives the range [30, 90] and a similar range about 170 gives [140 200] so significant separation exists between the two intensity populations. So choosing 170 as the seed value from which to grow the objects is quite adequate. One approach is to grow regions by appending to a seed any pixel that is 8-connected to any pixel previously appended to that seed, and whose intensity is $170 \pm 3\sigma$

Problem 10.42

The region splitting is shown in Fig. P10.42(a). The corresponding quad tree is shown in Fig. P10.42(b).

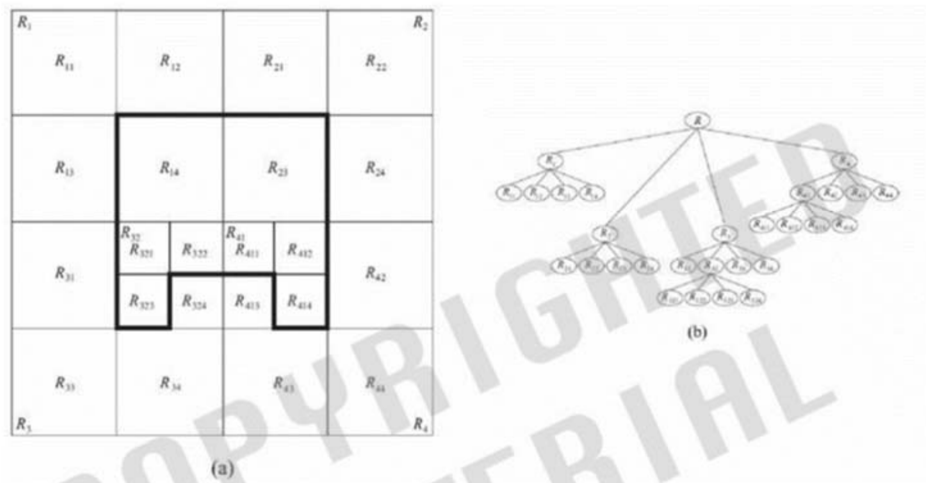

Figure P10.42

Problem 10.49

The first step in the application of the watershed segmentation algorithm is to build a dam of height $\max+1$ to prevent the rising water from running off the ends of the function, as shown in Fig. P10.49(b). For an image function we would build a box of height $\max+1$ around its border. The algorithm is initialized by setting $C[1]=T[1]$. In this case, $T[1]=\{g(2)\}$, as shown in Fig. P10.49(c) (note the water level). There is only one connected component in this case: $Q[1]=\{q_1\}=\{g(2)\}$.

Next, we let $n=2$ and, as shown in Fig. P10.49(d), $T[2]=\{g(2),g(14)\}$ and $Q[2]=\{q_1;q_2\}$, where, for clarity, different connected components are separated by semicolons. We start construction of $C[2]$ by considering each connected component in $Q[2]$. When $q=q_1$, the term $q\cap C[1]$ is equal to $\{g(2)\}$, so condition 2 is satisfied and, therefore, $C[2]=\{g(2)\}$. When $q=q_2$, $q\cap C[1]=\varnothing$ (the empty set) so condition 1 is satisfied and we incorporate q in $C[2]$, which then becomes $C[2]=\{g(2);g(14)\}$ where, as above, different connected components are separated by semicolons.

When $n=3$ [Fig. P10.49 (e)], $T[3]=\{2,3,10,11,13,14\}$ and $Q[3]=\{q_1;q_2;q_3\}=\{2,3;10,11;13,14\}$ where, in order to simplify the notation we let k denote $g(k)$. Proceeding as above, $q_1\cap C[2]=\{2\}$ satisfies condition 2, so q_1 is incorporated into the new set to yield $C[3]=\{2,3;14\}$. Similarly, $q_2\cap C[2]=\varnothing$ satisfies condition 1 and $C[3]=\{2,3;10,11;13,14\}$. Finally, q3 ∩ C[2] = {14} satisfies condition 2 and C[3] = {2, 3;10,11;13,14}. It is easily verified that C[4] = C[3] = {2, 3;10,11;13,14}.

When $n=5$ [Fig. P10.49(f)], we have, $T[5]=\{2,3,5,6,10,11,12,13,14\}$ and $Q[5]=\{q_1;q_2;q_3\}=\{2,3;5,6;10,11,12,13,14\}$ (note the merging of two previously distinct connected components). It is easily verified that $q1\cap C[4]$ satisfies condition 2 and that $q_2\cap C[4]$ satisfies condition 1. Proceeding with these two connected components exactly as above yields $C[5]=\{2,3;5,6;10,11;13,14\}$ up to this point. Things get more interesting when we consider q_3. Now, $q_3\cap C[4]=\{10,11;13,14\}$ which, because it contains two connected components of $C[4]$, satisfies condition 3. As mentioned previously, this is an indication that water from two different basins has merged and a dam must be built to prevent this condition. Dam building is nothing more than separating q_3 into the two original connected components. In this case, this is accomplished by the dam shown in Fig. P10.49(g), so that now $q_3=\{q_{31};q_{32}\}=\{10,11;13,14\}$.

Then, $q_{31} \cap C[4]$ and $q_{32} \cap C[4]$ each satisfy condition 2 and we have the final result for $n = 5$, $C[5] = \{2, 3; 5, 6; 10, 11; 13; 14\}$.

Continuing in the manner just explained yields the final segmentation result shown in Fig. P10.49(h), where the "edges" are visible (from the top) just above the water line. A final post-processing step would remove the outer dam walls to yield the inner edges of interest.

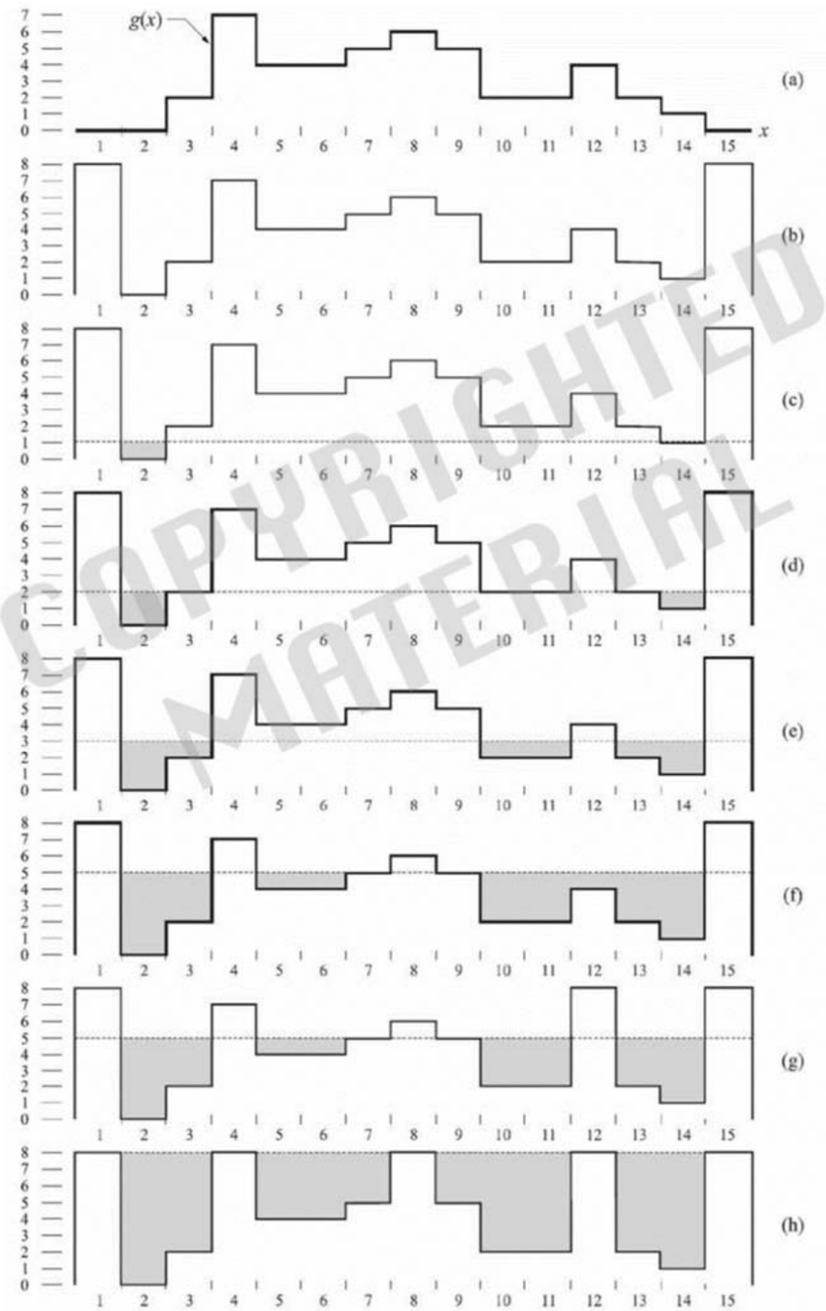

Figure P10.49