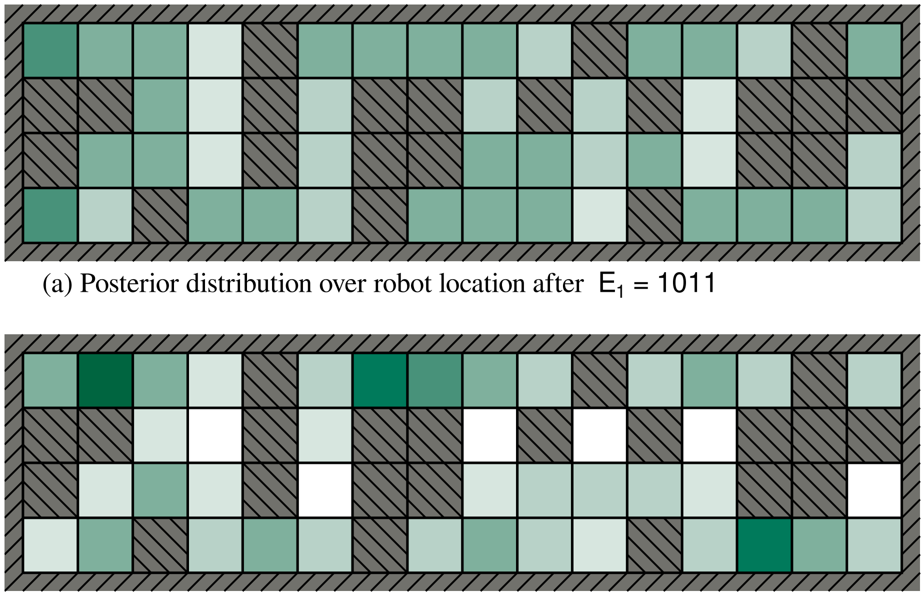

(a) Posterior distribution over robot location after $E_1 = 1011$

(b) Posterior distribution over robot location after $E_1 = 1011$, $E_2 = 1010$

Figure 14.7 Posterior distribution over robot location: (a) after one observation $E_1 = 1011$ (i.e., obstacles to the north, south, and west); (b) after a random move to an adjacent location and a second observation $E_2 = 1010$ (i.e., obstacles to the north and south). The darkness of each square corresponds to the probability that the robot is at that location; darker colors are more probable. The sensor error rate for each bit is $\epsilon = 0.2$.

bits that are different—between the true values for square i and the actual reading e_t, then the probability that a robot in square i would receive a sensor reading e_t is

$$P(E_t = e_t \mid X_t = i) = (\mathbf{O}_t)_{ii} = (1-\epsilon)^{4-d_{it}} \epsilon^{d_{it}} .$$

For example, the probability that a square with obstacles to the north and south would produce a sensor reading 1110 is $(1-\epsilon)^3 \epsilon^1$.

Given the matrices $\mathbf{T}$ and $\mathbf{O}_t$, the robot can use Equation (14.12) to compute the posterior distribution over locations—that is, to work out where it is. Figure 14.7 shows the distributions $\mathbf{P}(X_1 \mid E_1 = 1011)$ and $\mathbf{P}(X_2 \mid E_1 = 1011, E_2 = 1010)$. This is the same maze we saw before in Figure 4.18 (page 134), but there we used logical filtering to find the locations that were *possible*, assuming perfect sensing. Those same locations are still the most *likely* with noisy sensing, but now *every* location has some nonzero probability because any location could produce any sensor values.

In addition to filtering to estimate its current location, the robot can use smoothing (Equation (14.13)) to work out where it was at any given past time—for example, where it began at time 0—and it can use the Viterbi algorithm to work out the most likely path it has taken to get where it is now. Figure 14.8 shows the localization error and Viterbi path error for various values of the per-bit sensor error rate ϵ. Even when ϵ is 0.20—which means that the overall sensor reading is wrong 59% of the time—the robot is usually able to work out its location to within two squares after 20 observations. This is because of the algorithm's ability to integrate evidence over time and to take into account the probabilistic constraints imposed