Figure 14.5 Example of value iteration over state spaces in robot motion. Obstacles are shown in black. The value function is indicated by the gray shaded area. Greedy action selection with respect to the value function lead to optimal control, assuming that the robot's pose is observable. Also shown in the diagrams are example paths obtained by following the greedy policy.

for low-dimensional state and control spaces, due to the curse of dimensionality. In higher dimensional situations, it is common to introduce learning algorithms to represent the value function.

Second, we need a state! As noted above, it might be viable to replace the posterior by its mean

$$(14.19) \qquad \hat{x}_t \;\; = \;\; E[p(x_t \mid z_{1:t}, u_{1:t})]$$

In the context of robot localization, for example, such an approximation works well if we can guarantee that the robot is always approximately localized, and the residual uncertainty in the posterior is local. It ceases to work when the robot performs global localization, or if it is being kidnapped.

Figure 14.5 illustrates value iteration in the context of a robotic path planning problem. Shown there is a two-dimensional projection of a configuration space of a circular robot. The configuration space is the space of all $\langle x, y, \theta \rangle$ coordinates that the robot can physically attain. For circular robots, the configuration space is obtained by "growing" the obstacles in the map by the radius of the robot. These increased obstacles shown in black in Figure 14.5.

The value function is shown in gray, where the brighter a location, the higher its value. The path obtained by following the optimal policy leads to