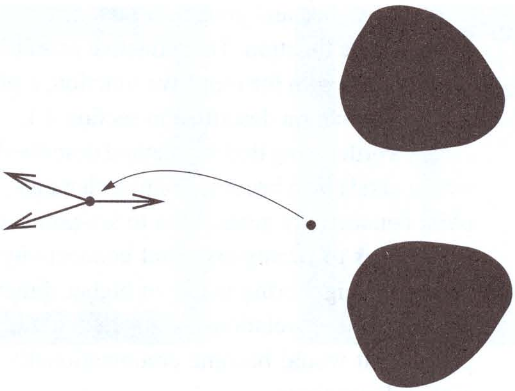

Figure 4.11 Local minimum without concave obstacles. The robot moves away from the two convex obstacles until it reaches a point where the gradient vanishes; at this point, the sum of the attractive gradient and the repulsive gradient is zero.

the robot. At this point, the effect of the top and bottom arms keeps the robot halfway between them and the goal continues to attract the robot. The robot reaches a point where the effect of the obstacle's base counteracts the attraction of the goal. In other words, the robot has reached a q^* where $\nabla U(q^*) = 0$ and q^* is *not* the goal. Note, this problem is not limited to concave obstacles as can be seen in figure 4.11. Local minima present a significant drawback to the attractive/repulsive approach, and thus the attractive/repulsive technique is not complete.

Barraquand and Latombe [37] developed search techniques other than gradient descent to overcome the problem of local minima present when planning with potential functions. Their planner, the Randomized Path Planner (RPP) [37], used a variety of potential functions some of which were simplified expressions of the potentials presented in this chapter. RPP followed the negative gradient of the specified potential function and when stuck at a local minimum, it initiated a series of random walks. Often the random walks allowed RPP to escape the local minimum and in that case, the negative gradient to the goal was followed again.

4.5 Wave-Front Planner

The wave-front planner [38, 208] affords the simplest solution to the local minima problem, but can only be implemented in spaces that are represented as grids. For the sake of discussion, consider a two-dimensional space. Initially, the planner starts with the standard binary grid of zeros corresponding to free space and ones to obstacles. The planner also knows the pixel locations of the start and goal. The goal pixel is labeled with a two. In the first step, all zero-valued pixels neighboring the goal are labeled with a three. Next, all zero-valued pixels adjacent to threes are labeled with four. This procedure essentially grows a wave front from the goal where at each iteration, all pixels on the wave front have the same path length, measured with respect to the grid, to the goal. This procedure terminates when the wave front reaches the pixel that contains the robot start location.

The planner then determines a path via gradient descent on the grid starting from the start. Essentially, the planner determines the path one pixel at a time. Assume that the value of the start pixel is 33. The next pixel in the path is any neighboring pixel whose value is 32. There could be multiple choices; simply pick any one of the choices. The next pixel is then one whose value is 31. Boundedness of the free space (and hence the discretization) and continuity of the distance function ensure that construction of the wave front guarantees that there will always be a neighboring pixel whose value is one less than that of the current pixel and that this procedure forms a path in the grid to the goal, i.e., to the pixel whose value is two.

Figure 4.12 contains six panels that demonstrate various stages of the wave-front propagation using four-point connectivity. Note that all points on the wave front have the same Manhattan distance to the goal. In the lower-left panel, note how the wave-front seemingly collides on itself. We will see later that the point of initial collision corresponds to a saddle point of the function that measures distance to the goal. This point then propagates away from the start as well. The trace of this propagation corresponds to a set of points that have two choices for shortest paths back to the goal, either going around the top of the triangle or below it.

The wave-front planner essentially forms a potential function on the grid which has one local minimum and thus is resolution complete. The planner also determines the shortest path, but at the cost of coming dangerously close to obstacles. The major drawback of this method is that the planner has to search the entire space for a path.

Finally, just like the brushfire method, the wave-front planner generalizes into higher dimensions as well. Consider the three-dimensional case first. Just as a pixel has four edges, a voxel (a three-dimensional pixel) has six faces. Therefore, the analogy to four-point connectivity with pixels is six-point connectivity with voxels. For a voxel with value i, if we assume six-point connectivity, then we assign $i + 1$ to the surrounding six voxels that share a face with the current voxel. Likewise, if we assume twenty-six-point connectivity (analogous to eight-point connectivity), then we assign $i + 1$ to all surrounding voxels that share a face, edge or vertex. It should be noted, however, implementation of the wavefront planner in higher dimensions becomes computationally intractable.

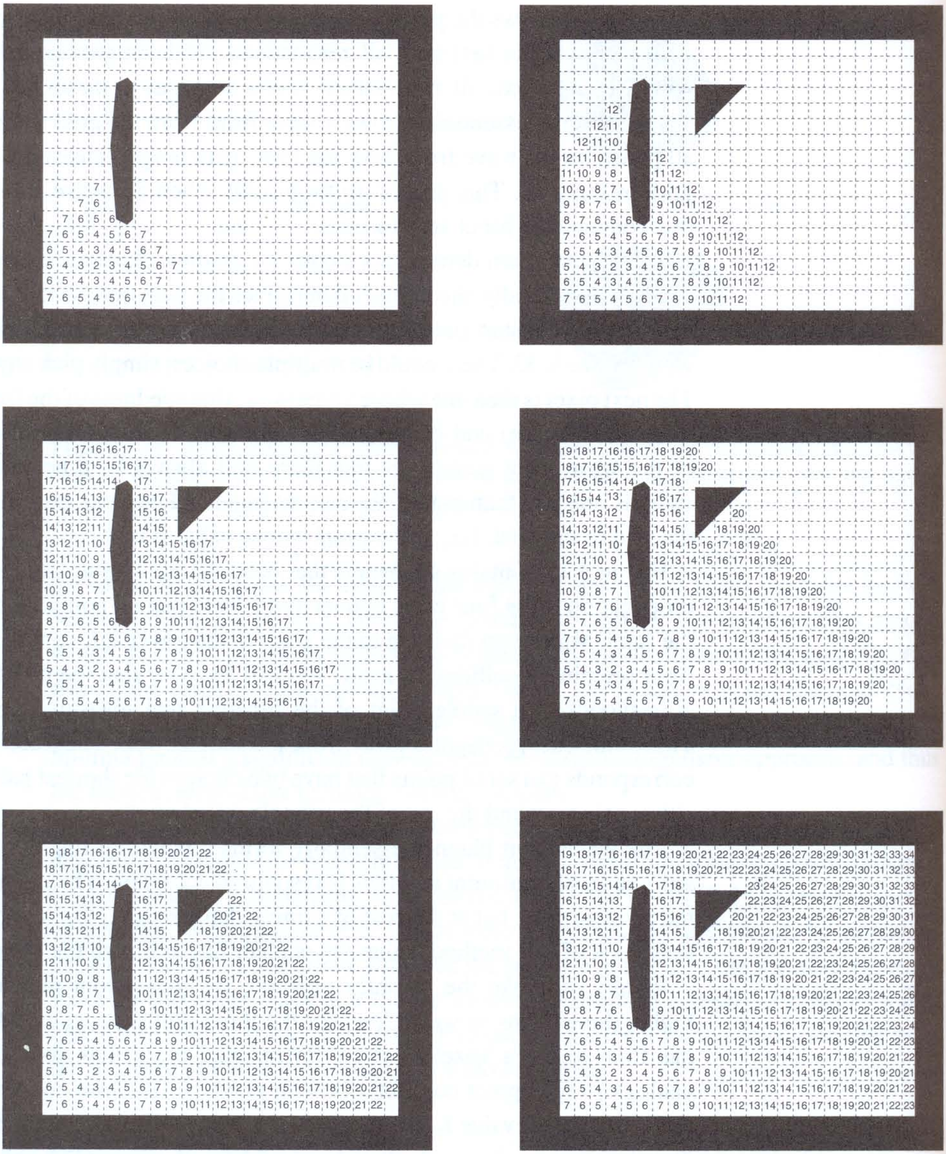

Figure 4.12 Propagation of the wave front using four-point connectivity (assume the start is in the upper-right corner and the goal is the origin of the wave front).