Angel: Interactive Computer Graphics; Fourth Edition

Chapter 10 Solutions

10.1 If the upper arm is longer than the lower arm then, as a first approximation, the robot can reach all points within a sphere centered at the joint on the base. However, this answer is only approximate because the lower arm cannot move below the level of the base, so a second approximation is a hemisphere above the base. This approximation has to have added to it points below the base that can be reached by the lower arm while the upper arm is parallel to the top of the base, a set of points that depends on the length of the upper arm and the radius of the base. If the lower arm is longer than the upper arm, then there will be an interior sphere radius equal to the difference in the arm lengths that cannot be reached.

10.3 For this simple example, there are three values of the joint angles and three coordinates in space, so there are a couple of simple approaches that can work. One is to usethe result in the previous problem to solve for the joint angles in terms of the given x, y, and z values, using a numerical method. We could then simply move the joint angles from their initial to final positions linearly.

A more interesting possibility is to have the tip of the robot arm trace out a given path, for example, a line segment between the starting and ending positions. If use the parametric form of this line, we can obtain a set of positions along it as the parameter varies from 0 to 1. For each of these intermediate positions, we can numerically solve for the joint angles and have the joint angles move linearly between successive points. This scheme would give us an approximately linear trajectory.

In general, the problem is far more difficult because a typical robot will have many more degrees of freedom and thus there will be many possible joint angle combinations for a given reachable position. We also have to worry about factors such as the maximum allowable rate at which joint angles can change.

10.11 The trick here is to store the average value of the four trees below at each node. Thus, if we have an $2^n \times 2^n$ image, the nodes at level n has the 2^{2n} pixel values, the nodes at level $n-1$ each store the average of the four pixels below them, and so on. The root node stores the average value of the image, which is the average of the values stored at its four children.

10.13 Yes. For instance, we can save the state at nodes so that changes in state cannot propagate.