UNIVERSITY OF NORTH CAROLINA AT CHARLOTTE

Department of Electrical and Computer Engineering
ECGR 4161/5196 Introduction to Robotics

Experiment No. 5 – A* Path Planning

Overview: The purpose of this experiment is to introduce A* path planning and implement the algorithm with the robot to navigate a room.

A*,said as "A star", Path Planning is an algorithm that is used in a variety of applications like video game development and driving direction software. The goal of the algorithm is to analyze the surrounding area of the object at each point along the way to the desired destination in order to obtain the shortest path while avoiding obstacles. An example of this algorithm is shown below. [1]

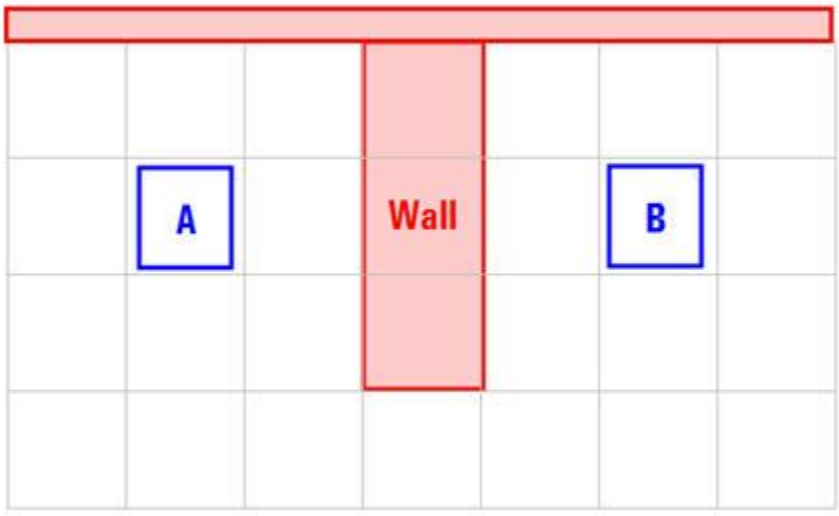

Figure 1. Starting point of example [1]

In Figure 1, the object is at point A and is moving towards point B. The algorithm divides the area of the room into squares with a value that will define it as either an empty space or a space that is occupied by an obstacle of some sort. Note that it is possible to change the squares into other shapes. The figure below shows the next step in the process for the algorithm. [1]

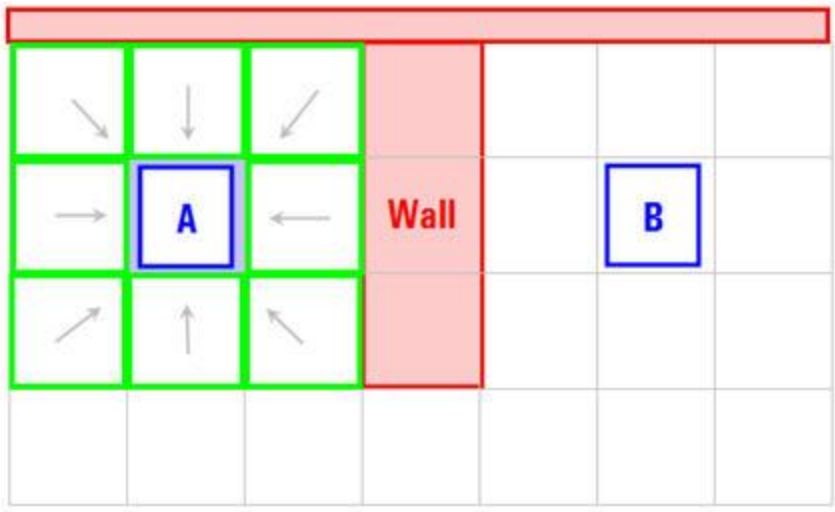

Figure 2. Searching surrounding area [1]

A search is then done at each square that is adjacent to the position of the object at point A. Each block is analyzed and the next best step is chosen in order to get to point B. Two lists are created and are called an open list and a closed list. The open list consists of nodes that are potential candidates for the next step. The closed list involves nodes that have already been examined and need not be examined again. Each time a search is done, the nodes that can be navigated are added to the open list and the nodes that are not able to be navigated are ignored. Point A is considered as the Parent Node and the gray arrows in Figure 2 point to this node. [1]

In order to determine which node would be the best one to walk to, an equation is used to determine the value of each node. The equation that is used to score the nodes is shown below. [1]

$$F = G + H$$

G = Movement cost to move from point A to the node on the grid.
H = Estimated movement cost to move from that node on the grid to the destination node, point B. This is usually referred to as heuristic.

The path is obtained by looking at each node and choosing the one that has the lowest F value. For this example, a value of 10 will be assigned to G when moving horizontally or vertically and a value of 14 for when moving diagonally. These numbers have been chosen because the distance to move diagonally is roughly the square root of 2 which equals 1.414 times the cost of moving horizontally or vertically. The value of H can be determined in many different ways. The method used for this example is the Manhattan method. This method involves counting the number of blocks it takes to move from the current node to the destination node while ignoring any obstacles in the way and any diagonal movement. Once we know the number of nodes, we multiply by 10, which is just the cost of moving horizontally and vertically. The results of using this equation are shown below. [1]

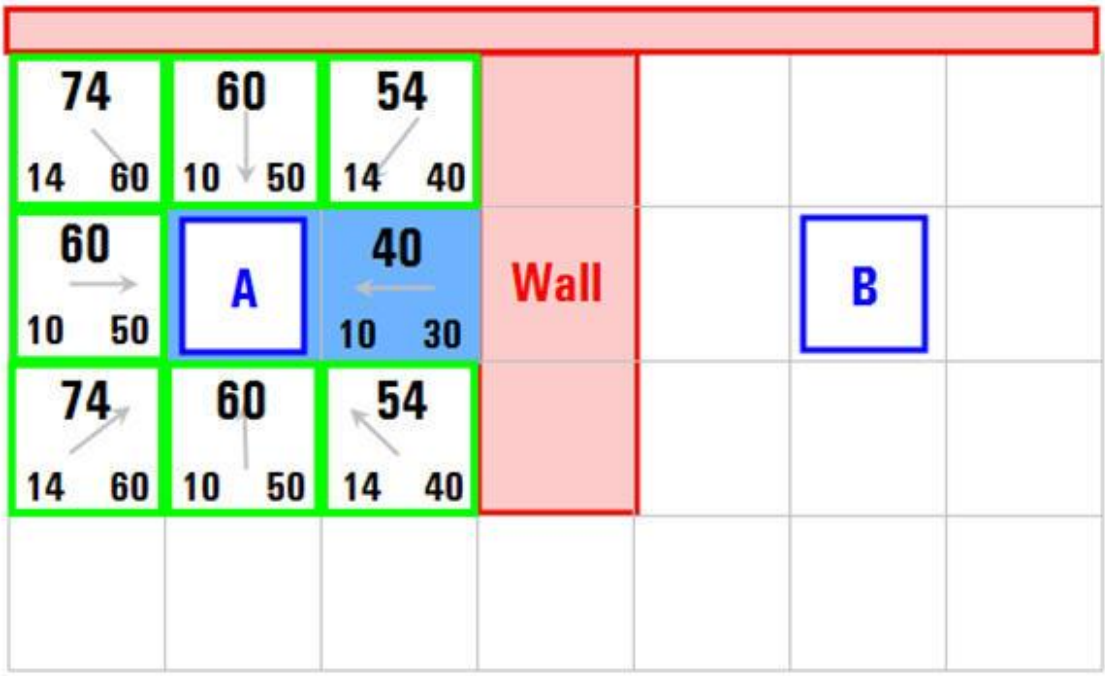

Figure 3. Nodes after being scored [1]

The node with lowest score is chosen and then added to the closed list. A search is then conducted again with the current node being the one with the value of 40 now. The obstacle is ignored and the one with the lowest score is the node with the value 54. Since the other four nodes already have values assigned to them, a test must be done to observe whether a better path may be obtained by going through the current node to get to the next node. For example take the node above the blue node, the current G score is 14. If we went through the current node to get there, then it would have a value of 20. This value comes from going horizontally once to get to the current node and then moving vertically to get to the node above it. Since this G score is higher than the current one, this is not the better path to take. When repeated for the other nodes, it turns out that none of them are a better path to take. Thus the next node to take is the one below the current node. This whole process is repeated until the destination node is added to the closed list. The result of our example is shown below. [1]

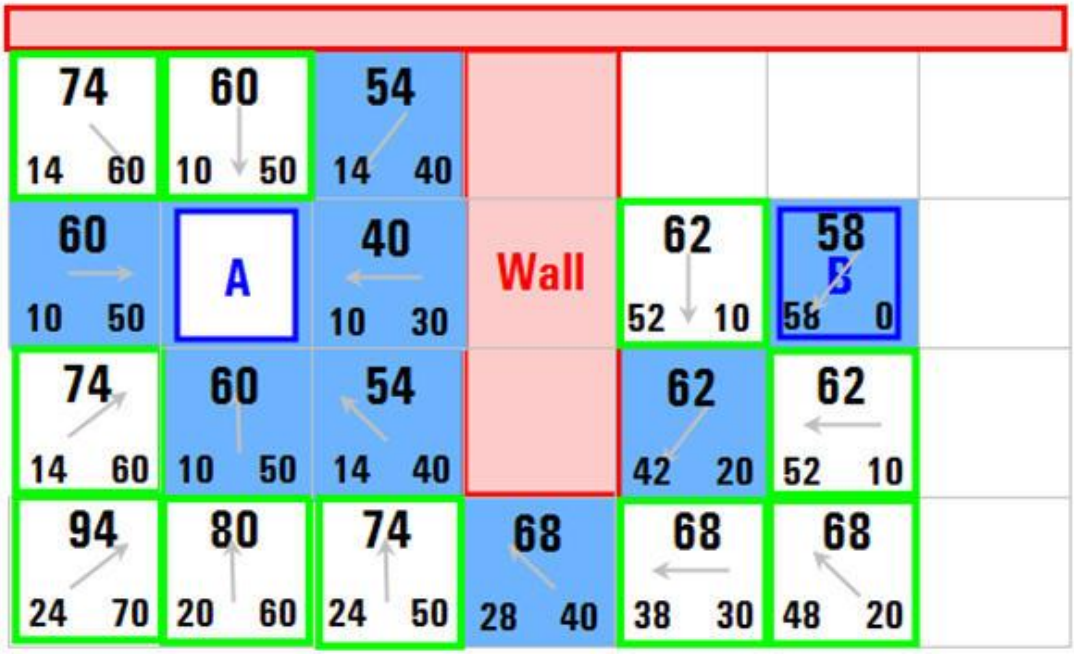

Figure 4. General path to reach destination [1]

The process is not yet complete. To find the actual path to take to the destination, one must start at the destination node and work backwards until you reach the starting node. The actual path is shown in Figure 5. [1]

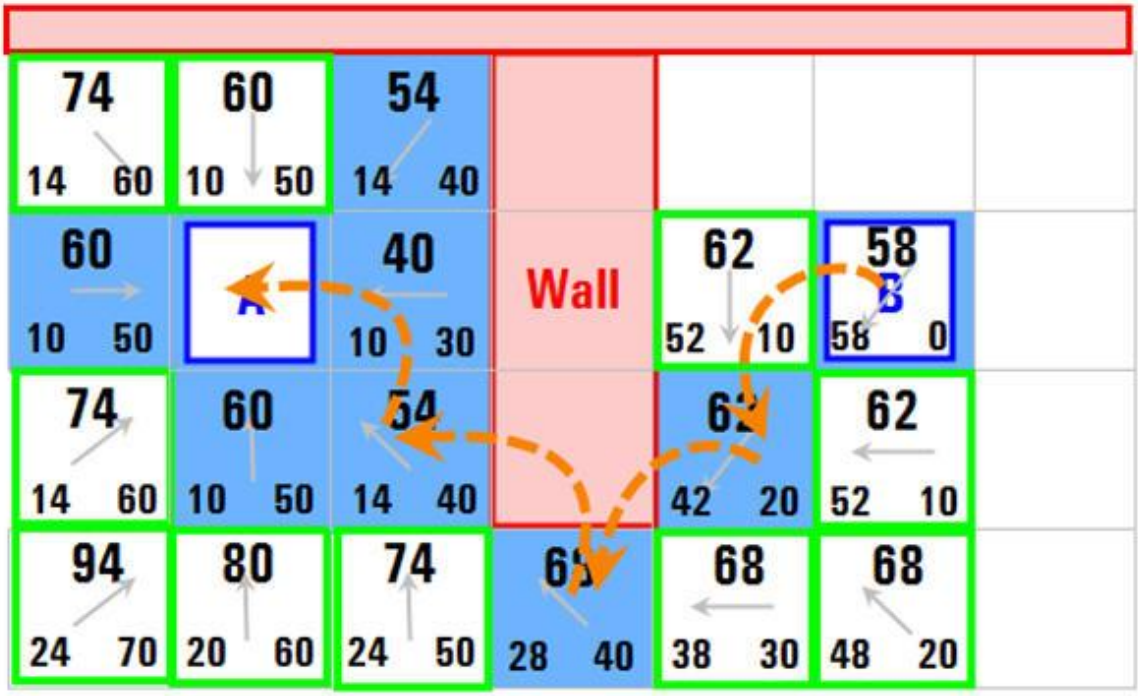

Figure 5. Final path generated by algorithm [1]

Pre-Lab – A* Path Planning

1.) How do you go about building the map to use for the A* Path Planning algorithm?

2.) Explain the procedure involved after building the map. (Note: A detailed discussion can be found on NI.com by searching for A* Path Planning)

Requirements:

Req. 1 No sensors will be used.

Req. 2 The algorithm will be implemented to have the robot navigate and make decisions on its own.

Req. 3 Robot Will Operate Autonomously.

Req. 4 Students will have access to the testing area prior to final demonstration.

Coding:

1. Decide what weights you will assign for moving horizontally, vertically, and diagonally.
2. Create the arrays that will hold the areas that are to be examined and that will hold the areas that need not be examined again.
3. Implement the equation for the algorithm that will calculate the best path in order to reach its destination.

References

[1] *NI Code Exchange*. National Instruments. n.d. Web. 1 Dec. 2010. <http://www.ni.com/code>