Using The ER1 Mapper

By: Abraham L. Howell

Abstract:

This paper will attempt to introduce and provide usage instructions for the ER1 Mapper (ER1M) software. Robot mapping and control are the primary focus of the ER1M. The ER1M has been designed to work with a mobile robot platform, the ER1, and thereby provide a simple yet effective mapping tool. The ER1 is a fully autonomous mobile robot manufactured by Evolution Robotics, Inc. What is unique about the ER1 is that an attached laptop provides the "Brains" and computational power. The attached laptop not only controls the motion and behavior of the ER1, but can also monitor a connected camera and/or additional sensors. A simple to use graphical user interface (GUI) named the Robot Control Center (RCC) is supplied with the ER1 and helps a user to control and create behavior for their ER1. The RCC is great for the first time robot enthusiast, but for more seasoned robotics experts it will quickly loose any initial appeal. However, this problem can easily be remedied since Evolution Robotics built a robot application programmers interface (API) into the RCC. The entire robot API can be accessed through a simple socket connection. This means that a programmer can write code in Visual Basic, Visual C++, Python, Java, or any language that supports TCP Sockets. Essentially, this is how the ER1M program is controlling the ER1 robot.

Introduction to ER1M:

The ER1M is a relatively simple to use program, but there are several concepts and ideas that must be understood before attempting to use the software. Currently, a total of 2500 cells or a 50x50 grid of cells is provided for creating an environment map. Each cell is 2 feet by 2 feet in size and can fully encompass the ER1 thereby allowing for changes in heading. Each cell will maintain one of the following possible states: clear, blocked, starting cell, or goal cell. A cell's state will be denoted by the following conventions: red circle = blocked, blue circle = start cell, green circle = goal cell. Obviously, a clear cell will maintain its original grey color until the user decides to change this. The start and goal cells denote the starting and ending positions for the robot. A clear cell can be traversed by the robot and therefore is a possible solution path. Blocked cells can't be traversed by the robot and thereby constrain the solution path.

Once a map has been properly created, the solution path can be solved for using the ER1M's built-in path solver algorithm. The A* algorithm has been implemented as the ER1M's path solver. The current implementation of the A* algorithm in the ER1M makes use of two heuristics: the distance from the start cell to the current cell, and the distance from the current cell to the goal cell. Setting one of the heuristics to be dominant will change the solution behavior. For example, by favoring the first heuristic, Dg, the solver will search more cells and try to find the optimal path. Whereas when favoring the second heuristic, Dh, the solver will explore fewer cells, find a path quicker, but in most

cases it will not find the optimal path. In this paper the optimal path is defined as the path with the shortest length.

Using The ER1M To Create & Solve Maps:

Now that a general overview has been given we can proceed to the actual usage of the software. You will hopefully find that using the ER1M is quite simple. Upon running the program you be presented with a form similar to the one shown below in figure1.

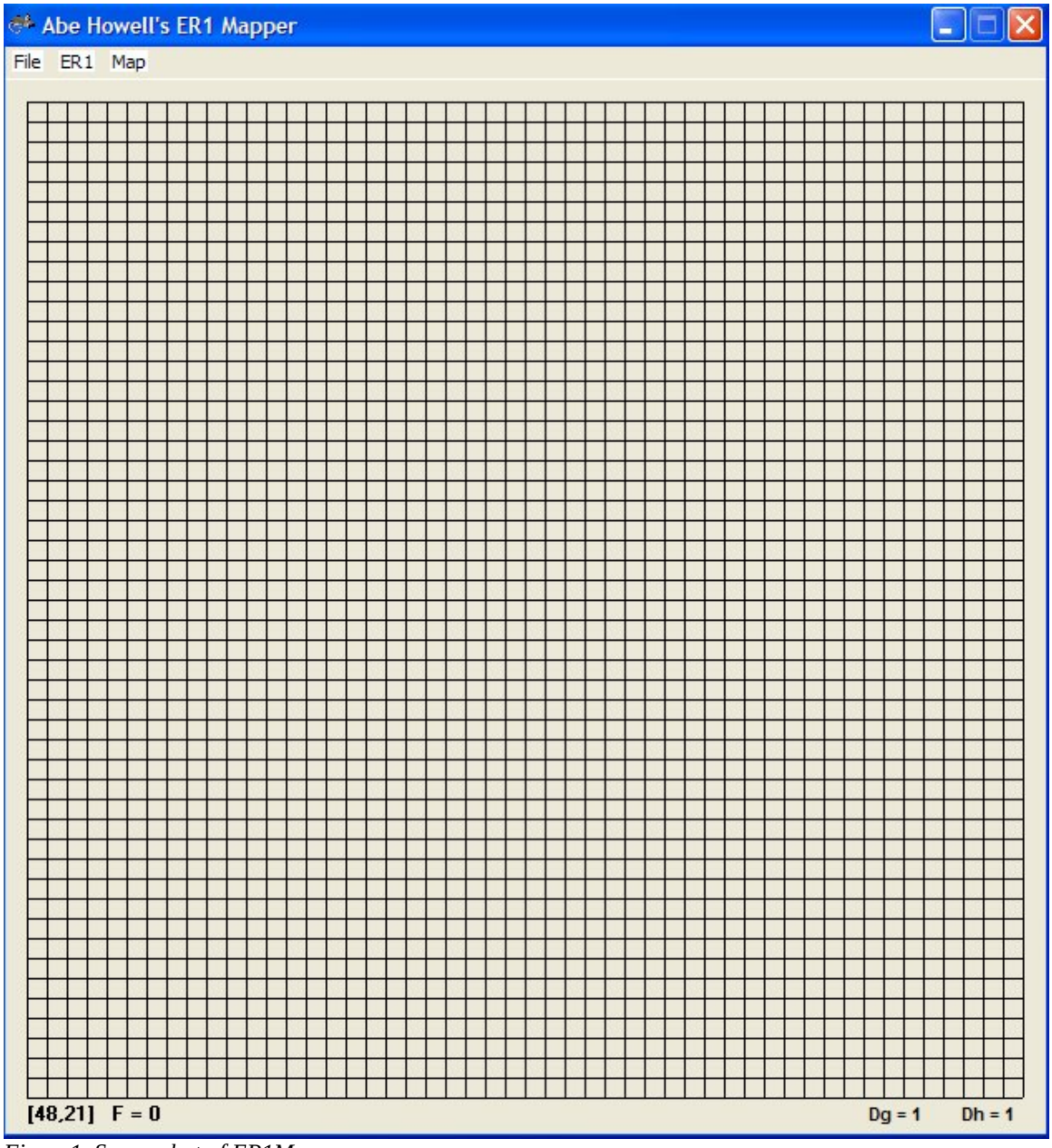


Figure1. Screenshot of ER1M.

The first step is to create the start and goal cells of the map. Start and goal cells are created using the keyboard and left mouse button. To create the start cell simply press the

“s” key and then press the left mouse button, but first place the mouse cursor over the destination cell. The goal cell is created in much the same way except that the “g” key must be pressed instead of “s” key. A blocked cell is also created in a similar manner; first place the mouse cursor over the destination cell, press the “b” key and finally press the left mouse button. A simple map below illustrates the above-mentioned concepts.

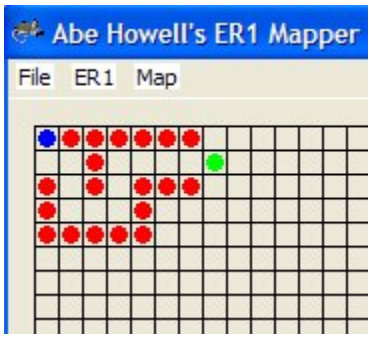


Figure2. Simple Map.

Now that a map has been created the next step is to set the heuristic values and also decide whether or not you want the investigated cells highlighted. To set the Dg and Dh heuristics simply use the Map menu and select either Set Dg or Set Dh. By default both heuristics are set to 1 and the investigated cells will be highlighted. If you don’t want to highlight the investigated cells simply use the Map menu and uncheck the Show Investigated Cells menu item. For this example I will simply accept the default setting and move on to the path solution.

Solving for the solution path is just a click away. Again, using the Map menu select Solve Path and the solution process will begin and should be completed almost instantly. You will notice that a blue line has been drawn to show the solution path. Also, at the bottom of the cell grid you will see a text message, which provides the details of the solution.

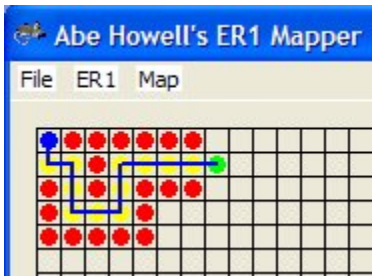


Figure3. Solution Path.

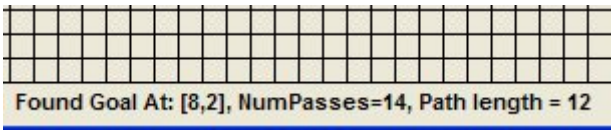
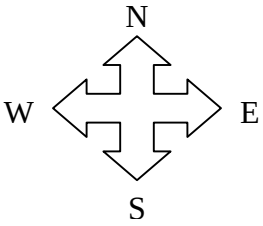


Figure4. Details of current solution.

For our solution you should observe that the goal was in fact found, a total of 14 passes were made to discover the solution and the path length is 12. If you were to change the heuristics for this map and resolve you wouldn’t notice any change in the path length because this map completely constrains the path to only one possible solution. You may want to build a less constraining map and then solve the path for different heuristic scenarios and see how the solution path changes.

Using The ER1M To Control An ER1 Robot:

Now that you've created and solved the path for a specified map you want to be able to command an ER1 Robot to actually traverse the solution path. The first step is to connect to the ER1 robot by using the ER1 menu and then selecting Connect to ER1. If a connection is successful then you will see a text message at the bottom of the cell grid, which reads as follows: Connected to ER1. However, if the connection fails you will be provided with the following message: ER1 Connection Timed Out! Assuming that you have a good connection the next step is to physically place the ER1 robot at the start cell and set its direction or heading. The ER1 Mapper follows the standard North, South, East, West map convention illustrated below.



As an example say that we decide to place the ER1 at the start cell shown below facing in the direction of the orange arrow. How would we set the ER1's direction?

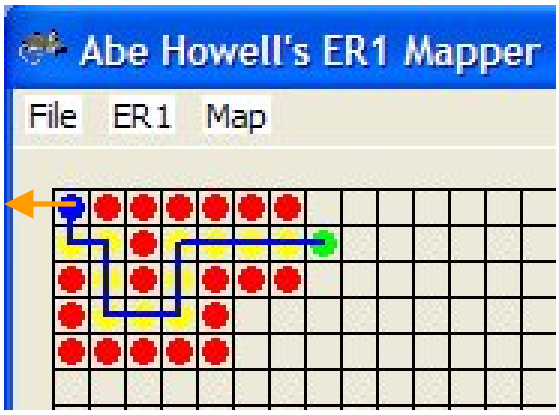


Figure5. Setting the ER1's heading.

According to our map convention and the orange arrow's direction we would need to set the ER1's direction to be west. To do this simply use the ER1 menu and select Set ER1's Direction and finally check the West sub-menu item as shown below.

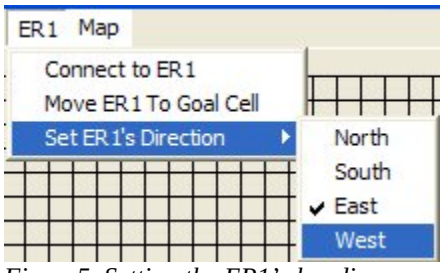


Figure5. Setting the ER1's heading.

The final step is to actually run the ER1 robot along the solution path by using the ER1 menu and selecting the Move ER1 to Goal Cell menu item. The ER1 robot should immediately begin moving along the solution path and stop once it arrives at the goal cell. The ER1 Mapper has several other built-in features that weren't mentioned and these include the ability to save created maps to file and then later retrieve these maps for editing and use.

Figure6. Picture of the ER1 Robot.