A Visual Treatment of the N-Puzzle

Matthew T. Hatem
University of New Hampshire
mtx23@cisunix.unh.edu

Abstract

Informed search strategies rely on information or heuristics to find the shortest path to the goal. Understanding the intricacies of these search strategies and the heuristics they employ is of great importance to AI research. Visualizations of a search space may provide some insight. However, visualizing large hierarchical data can be challenging. Traditional graph visualizations such as top-down or left-right tree layouts would yield massive images that are difficult to navigate or distill any useful information at a glance. In this paper we evaluate an interactive visualization for large tree structures. We use these visualizations to explore the search spaces of the 8-Puzzle. We show how this interactive visualization can be used to compare two common heuristics for the 8-Puzzle.

Introduction

The most well-known best-first search strategy is A*. The A* search strategy evaluates nodes in a state space by combining the cost to reach a node n with the estimated cost of the optimal path to the goal from n. The estimated cost is defined by a heuristic function, h(n). It follows that if h(n) does not overestimate then A* is both complete and optimal.

If we look at how A* performs on a two-dimensional grid world we can see contours develop as A* fans out from the initial state. The contours represent nondecreasing ranges of f-values. F-values represent the cost to reach a node n plus the cost of the optimal path to the goal from n.

Visualizations of these contours and other patterns may serve as tools to develop a deeper understanding of heuristic search algorithms. They may also be used to measure the quality of a heuristic function or compare two heuristics. Models like the two-dimensional grid world are ideal for developing these visualizations because of their geographic nature. The physical representation of the model can be used as the basis of the visualization (see figure 1). There exists models that are more abstract like the N-Puzzle where the physical model is not as useful (see figure 3).

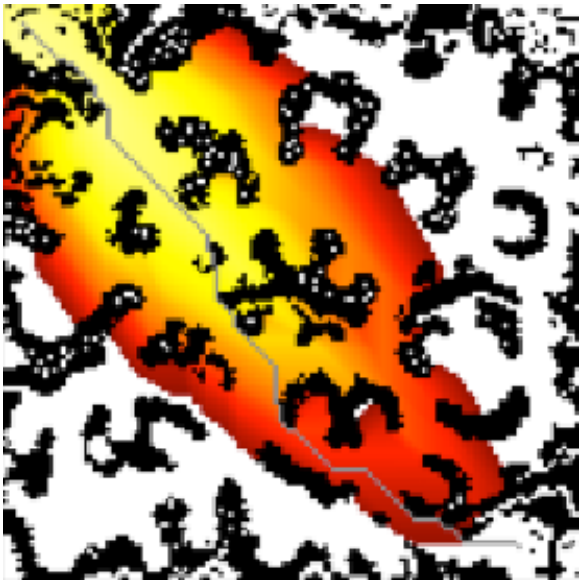

Figure 1 Visualization showing contours in an A* search on grid world (courtesy of Jordan Thayer).

All hope is not lost for these abstract models. Instead of visualizing the physical representation of the model we can visualize the search space. The search space can be viewed as a tree of search nodes that represent a state. The root node represents the initial state. A vertex between two nodes in the tree represents the action taken from the state of the parent to generate the state for the child.

In some forms of heuristic search it is possible for the search to revisit a state. If we were to construct a graph of states for a search space it would in some cases be a graph and not a tree. But what if we are to consider the search space at a higher level as a graph of search nodes that refer to a state which may or may not be duplicated. Each node is guaranteed to be unique and has a unique path to the root. A search space represented this way is clearly a tree by definition. It follows that any visualization for trees would be applicable to a search space.

In this paper we evaluate the Sunburst visualization as a tool for exploring search spaces. The Sunburst visualization is a proven technique for exploring

large tree structures. We use this visualization to explore the state space of the popular 8-Puzzle model and compare two common heuristics used for 8-Puzzle with the A* search strategy.

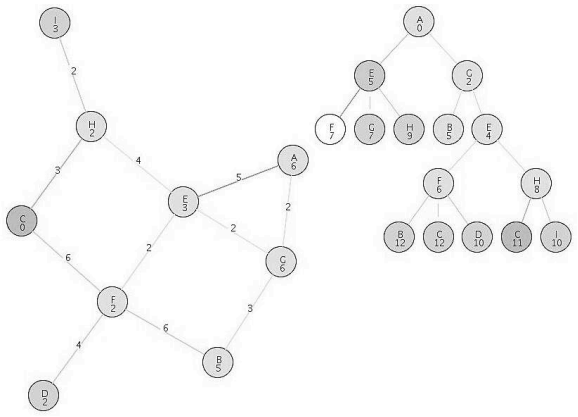

Figure 2 Small search spaces represented as a graph and a tree.

The N-Puzzle

Also known as the sliding tile puzzle, the N-Puzzle consists of a grid of sliding tiles numbered 1-N with the (N+1)th tile removed. The objective of the puzzle is to slide the tiles from some random initial state to a goal state.

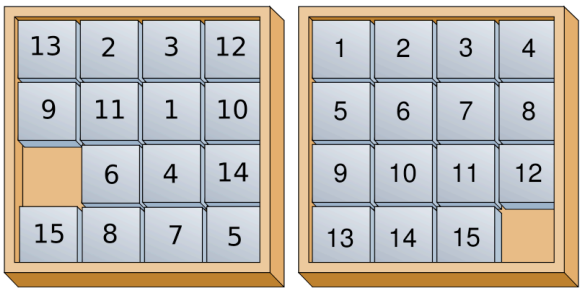

Figure 3 (left) A random state of the 15-Puzzle. (right) The goal state of the 15-Puzzle.

The initial state of the tiles is not simply a random permutation of the tiles. In fact the set of initial states that are solvable is exactly half of the total permutations of the tiles. This set is equal to the set of even permutations of the tiles. Solving an N-Puzzle from a starting state that is an odd permutation of the goal state is impossible. This was proven mathematically a few months after the puzzle was invented in 1879.

N-Puzzle is a popular model for analyzing algorithms used in shortest path problem solving. For N-Puzzle this means solving the puzzle in the fewest possible moves.

Heuristics for 8-Puzzle

Commonly used heuristics for the N-Puzzle include the misplaced tile and Manhattan distance heuristics. The misplaced tile heuristic is the sum of misplaced tiles in the puzzle at any given state. The Manhattan distance heuristic is the sum of all Manhattan distances of each tile. The Manhattan distance heuristic has been proven to perform better than the misplaced tile heuristic. Both heuristics are admissible, as they never overestimate the cost to reach the goal state from any other state.

A more advanced heuristic, Manhattan pair distance, combines the Manhattan distance with a pairing of misplaced tiles. This is guaranteed not to overestimate and is more accurate than Manhattan distance alone. For the purposes of this evaluation we are concerned with the misplaced tile and Manhattan distance heuristics.

Sunburst Visualization

Sunburst is a space filling visualization that uses a radial layout to represent large hierarchical tree structures. The root of the tree is placed at the center of the Sunburst and child segments are bound to the angle of the parent. The colors of the segments represent one or more dimensions of the data. The size of each segment is strictly a function of the layout. Segments that have children are given enough space to fit the children along its perimeter.

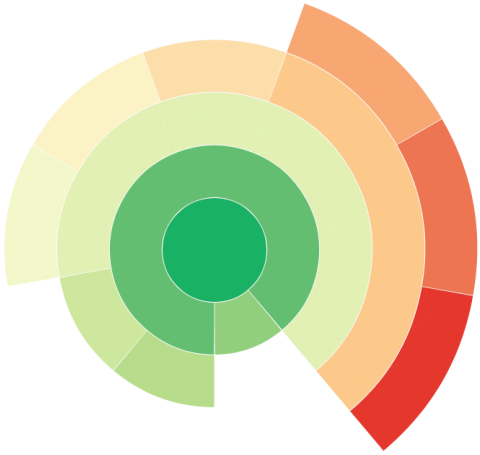

Figure 4 A small Sunburst visualization.

Experiments have shown that the Sunburst visualization is preferred over other space filling visualizations, like the TreeMap, because of its explicit representation of the tree structure.

Interactions with Sunburst

The Sunburst visualization we used for our evaluation supported two types of interactions. By moving the cursor over a node, the node is highlighted and a tooltip window is shown containing information about the highlighted node. Selecting a highlighted node transitions the Sunburst visualization into "drill down" mode. An overview of the entire tree is presented in the center of the canvas and a new Sunburst is constructed surrounding the center with the selected node as the root.

Figure 5 Sunburst in drill down mode

Approach

Our goal is to evaluate the Sunburst visualization as a tool for analyzing search spaces and heuristics. We take the position that search spaces can be represented as a tree whereby the Sunburst visualization is applicable. For the purposes of our evaluation we have implemented a framework for A* search that supports weighting and pluggable heuristic functions. We also developed a basic model for the 8-Puzzle and two common heuristics, misplaced tile and Manhattan distance. The search traces were exported as XML documents that are later used as input for the Sunburst visualization tool.

The Sunburst visualization tool we used is based on TreeViz, an open source project designed to support the generation of visualizations of large tree structures. We made some modifications to TreeViz for the purposes of our evaluation (see appendix C).

To generate the search spaces needed for our evaluation we expanded the entire 8-Puzzle state space using a breadth first expansion from the goal state until all of the 181,440 unique states were generated. A state space file was generated for each heuristic. These state space files were necessary in order to visualize the error in search spaces. Next, we generated a collection of search spaces using initial states of varying distances from the goal (see figure 6 for the actual states used and their distances from the goal through an optimal path).

Initial State	Distance from Goal
023145786	3
542710863	12
751364208	22
647805321	29
321674850	29

Figure 6 Initial states used in our evaluation. The states are a sequence of digits where each span of 3 digits represents a row in the puzzle with 0 as the blank tile. The goal state is 123456780.

The information stored in the XML files for each node in the state space consisted of the f(n), g(n) and h(n) values as well as the relative time at which the node was generated (see appendix B for an example state space file).

Evaluation

In our evaluation of the Sunburst visualization we were interested in a few of the features of a typical search space. These features include the f-value, h-value, error and time or f(n), h(n), e(n) and t(n) respectively. The e(n) function is the difference between the computed h-value for node n and the actual cost to reach the goal from node n. The t(n) function is the time at which node n was generated. We used a diverging color scheme in all of the visualizations that we generated where green represents the lowest value and red represents the highest value.

You may refer to appendix A for our complete results. Here we will present the results we thought were most relevant to the utility of the Sunburst visualization as a tool for exploring search spaces and analyzing heuristics.

Shape

We found Sunburst to be most effective at showing the shape of a tree. We compared the shapes of the trees generated using the misplaced tile heuristic and the Manhattan distance. As expected the Manhattan distance heuristic branches less and generates far fewer nodes. The pruning power of the superior heuristic is evident in the visualizations.

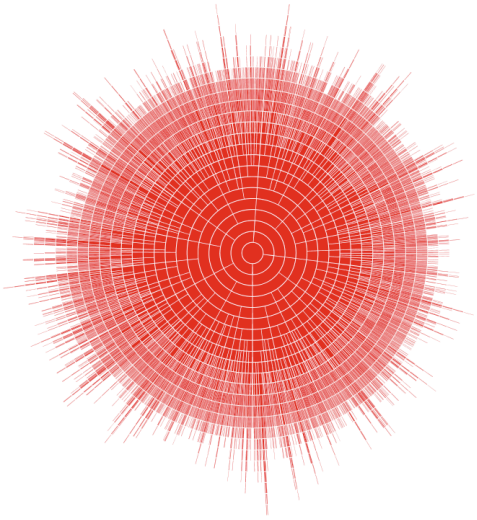

Figure 7 Shape of search space tree with initial state 751364208 using misplaced tile heuristic.

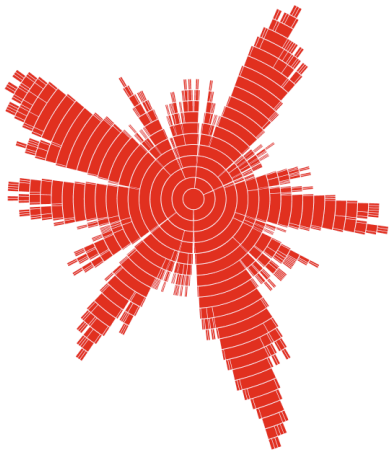

Figure 8 Shape of search space tree with initial state 751364208 using Manhattan distance heuristic with a weight of 1.

The Sunburst was also effective at showing the impact that weight has in WA* search.

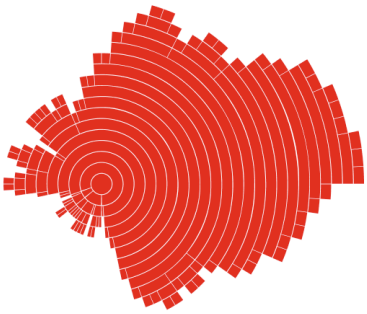

Figure 9 Shape of the search space tree with initial state 751364208 using Manhattan distance heuristic with a weight of 2.

H-Values

We generated some visualizations of the entire 8-Puzzle state space to get a sense of the h-value distribution as well as the error in computed h-values. The Sunburst visualization is effective in showing the higher degree of error in the misplaced tile heuristic (see figure 11 and 12).

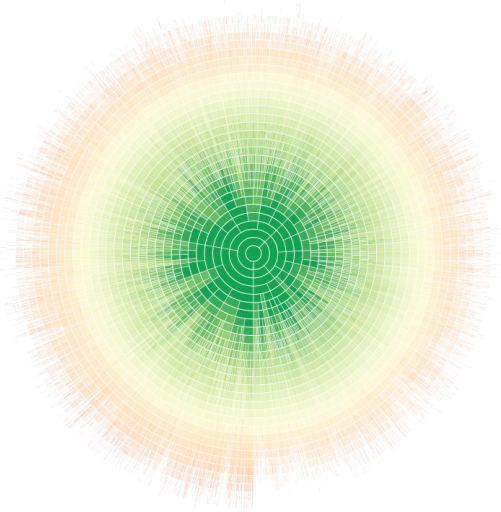

Figure 10 Error in h in the entire 8-Puzzle state space using misplaced tile heuristic.

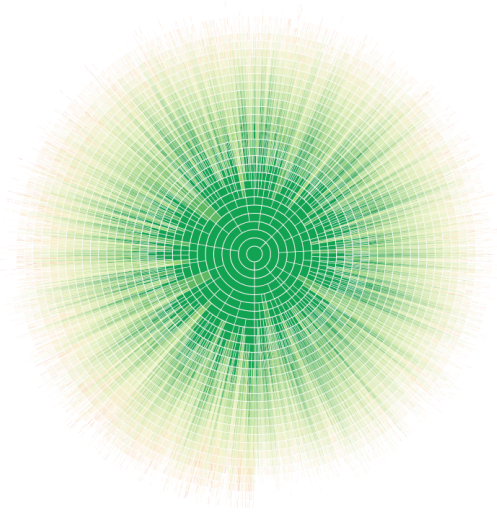

Figure 11 Error in h in the entire 8-Puzzle state space using Manhattan distance.

With the misplaced tile heuristic we see large bands of red and yellow indicating a higher degree of error. With the Manhattan distance heuristic we see more greens and yellows indicating a lower degree of error throughout the state space. Streaks of green that stretch from the root of the tree to the fringe can be seen with the Manhattan distance heuristic. This may suggest that there is a set of search paths whereby the heuristic is more accurate than for other paths. There are no such streaks visible with the misplaced tile heuristic.

F-Value

In visualizing the f-value of an admissible heuristic there are two things we expected to see. First since f(n) is nondecreasing we expected to see contours as well as a monotonic color gradient fanning out from the initial state.

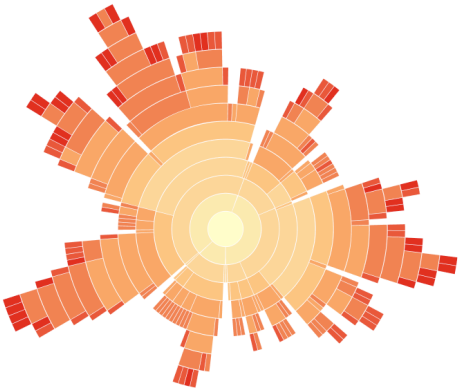

Figure 12 Contours and monotonicity in f-values

The monotonicity is clearly evident in the visualizations (see figure 12). The contours become less clear as the trees get larger (see appendix A for an example of a larger tree showing f-values).

With WA* we expected to see something different since weight can cause a heuristic function to overestimate the cost to the goal and potentially arrive at a suboptimal solution. This is evident in the visualization as a rippling effect (see figure 13).

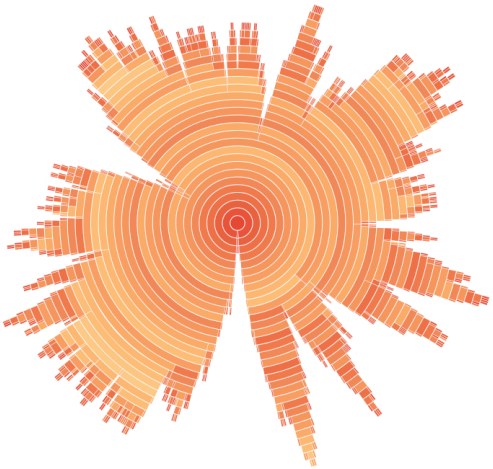

Figure 13 Ripple effect caused by overestimation when weight is greater than 1 in WA*.

Time

The time at which a node is generated is also an important feature of a search space. In our evaluation we used Sunburst to compare time with f-values.

Figure 14 F-Values

Figure 15 Time

We expected to see a relationship between the time a node was generated and the f-value of the parent node. Nodes whose parents have higher f-values should be generated after nodes whose parents have lower f-values. The Sunburst is effective at showing this relationship. However two Sunbursts need to be compared side by side. This is true in general when looking at relationships or differences between features using Sunburst.

Conclusion

Search spaces generated by heuristic search algorithms such as A* can be represented by tree structures. The Sunburst is an effective interactive visualization for exploring these search spaces. It is especially effective at showing the shape of a search space. It may also be useful in comparing the quality of heuristics and validating the relationships between features in the search space.

The Sunburst visualization tool we used was restricted to two visual processing channels, color and orientation. Because of this it is not very effective in showing more than one dimension of the data in a single diagram.

Future Work

Extending Sunburst
The Sunburst utilizes the orientation and color processing channels for visual information. The size channel is reserved as a function of the layout. This leaves animation as a channel that may be used to extend the Sunburst to show more dimensions of the data in a single diagram.

We may also consider overloading the orientation or color channels to make the Sunburst more expressive. It is not clear how we might do this. One possibility is to use alpha-blending to fade or merge nodes.

Exploring the Fringe
The open-list is an important area of study in developing search strategies such as A*. The open-list, also referred to as the fringe, is the set of nodes that is of immediate consideration for expansion during a search. Visualizations of the fringe and how it relates to the search space may deepen understanding in this area.

References

Russell, S., and Norvig, P. 2003. Artificial intelligence, a modern approach.

Ware, Colin 2004. Information Visualization: Perception for Design, 2nd Ed. San Francisco, Morgan Kaufman

A.F. Archer, A modern treatment of the 15 puzzle, Amer. Math. Monthly 106 (1999) 793-799.

Ivan Herman, Member, I.C.S.G.M., and Marshall, M.S. 2000. Graph visualization and navigation in information visualization: a survey. VOL. 6, NO. 1

Johnson, Brian and Shneiderman, Ben, Tree-maps: A Space Filling Approach to the Visualization of Hierarchical Information Structures, Proceedings of the IEEE Visualization '91, Oct. 1991, San Diego, CA, pp. 284-291.

Andrews, Keith and Heidegger, Helmut, Information Slices: Visualising and Exploring Large Hierarchies using Cascading, Semi-Circular Discs, IEEE Information Visualization Symposium 1998, Late Breaking Hot Topics Proceedings, Oct. 1998, pp. 9-12.

Stasko, John, Guzdial Mark, and McDonald, Kevin 1999. Evaluating Space-Filling Visualizations for Hierarchical Structures, Unpublished article.

Stasko, John "Sunburst"
http://www.cc.gatech.edu/gvu/ii/sunburst

David Furcy, A. J., and Naps, T. Blocktree pedagogical information visualization for heuristic search.

Werner Randelshofer, Visualization of large tree structures http://randelshofer.ch/treeviz/

Color Brewer
http://www.personal.psu.edu/cab38/ColorBrewer/ColorBrewer.html