Comparison of Different Grid Abstractions for Pathfinding on Maps

Yngvi Bjornsson, Markus Enzenberger, Robert Holte, Jonathan Schaeffer and Peter Yap
Department of Computing Science, University of Alberta,
Edmonton AB, Canada T6G 2E8
{yngvi, emarkus, holte, Jonathan, petery}@cs.ualberta.ca

Abstract

Pathfinding on a map is a fundamental problem in many applications, including robotics and computer games. Typically a grid is superimposed over the map where each cell in the grid forms a unique state. A state-space-based search algorithm, such as A* or IDA*, is then used for finding the optimal (shortest) path. In this paper we analyze the search behavior of both A* and IDA* using different grid representations, providing various new insights via analytical and empirical results.

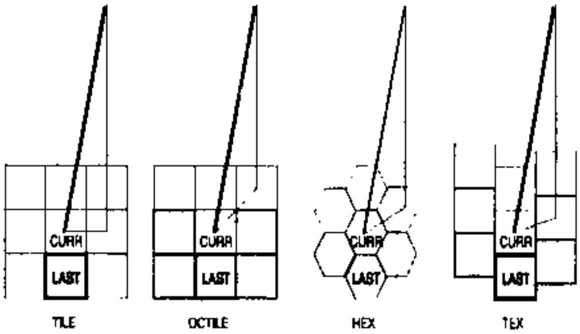

Figure 1: Different grid representations

1 Introduction

Commercial games are a multi-billion-dollar industry and still growing. In the past better graphics have been one of the main driving forces of sales; however, this is no longer true and good graphics alone is not sufficient to fuel sales. Instead consumers are increasingly looking for a more realistic gaming experience. For many type of games, especially real-time-strategy (RTS) and role-playing games (RPG), realistic real-time pathfinding is one of the fundamental hurdles to overcome. In many state-of-the-art games a grid is typically superimposed over the map and an algorithm such as A* or IDA* is used to find a path in the abstract grid space.

In this paper we compare four different grid topologies for abstraction: *tiles, octiles, hexes,* and *texes.* Tiles are squares where movement is restricted to the four cardinal compass directions (N, E, S, W). Octiles are also squares but allow additional diagonal moves (NE, SE, SW, NW). A hex grid consists of hexagons with six degrees of movement. One problem with a hex grid is that it cannot be implemented as effectively as a square-based grid (an important consideration in computer games). Thus we introduce texes, a square-based representation of a hexagonal grid. The four different grid structures are depicted in Figure 1.

2 Analytical Comparison

In this section we analyse the search complexity of IDA* for each of the different grid topologies discussed above. In our analysis we assume unit-cost edge weights (as is customary).

The asymptotic search complexity of A* on a grid is $0(D^2)$ independent of grid representation, where *D* is the depth of the solution. On the other hand, the search complexity of IDA* is $O(b^D)$ where *b* is the effective branching factor (in this analyses we exclude the effect of the heuristic estimate). The choice of a grid representation affects not only the branching factor *b,* but also the solution depth *D.* This poses the question: which grid representation is the most efficient?

As for the branching factor, there is no need to explore again the cell that the search just departed from, thus the effective branching factor of IDA* is at least one less than the number of adjacent cells. However, for hexes, texes and octiles we can further reduce the branching factor by observing that *none of the neighbors shared with the previously visited cell need to he expanded.* For example, in Figure 1 the shaded cells need not be expanded at node *curr* given that it was last entered via node *last.* Intuitively, because of unit-cost terrain, the cost of reaching these neighboring cells from *curr* is necessarily greater than if they were instead visited directly from *last.* The effective branching factor of hexes and texes is thus only 3, same as for tiles, whereas for octiles the asymptotic branching factor is 4.2 (3 for horizontal/vertical moves and 5 for diagonal moves)[1]. Surprisingly, this observation has been overlooked in previous studies.

As for the search depth *D,* it can be mathematically shown that given the same distance from a *start* to a *goal,* if a tile grid searches to a depth *D* then a hex grid searches only to a depth 0.81D and an octile grid to a depth $D/\sqrt{2}$. The analytical result is summarized in Table 1.

Table 1: Summary of Grids

Grid Type	Adj. Nodes	Branch. Factor	Average Depth	Complexity A*	Complexity IDA*
Hex	6	3	0.8 I D		
Tex	6	3	0.8ID	$0\{D^2)$	$0(2.43^D)$
Octile	8	4.2	0.7ID	$0(D^2)$	$0(2.77^D)$
Tile	4	3	1.00D		$O(3.00^D)$

3 Testing Framework

The above analytical study gives us a good idea of the merits of the different grid representations, at least when used with IDA*. However, for A* it does not provide much insight. An empirical study of these algorithms is thus important, especially given that no comparison study exists on their behavior when pathfinding on a map. A problem with previous work is that different studies use disparate testing environments making individual results hard to compare. Also, the "random" maps typically used are not necessarily representative of maps one would encounter in practice.

To address both of the above mentioned problems we developed a generic framework for testing state-space based search algorithms. This framework consists of three layers, allowing implementation of each layer independently of the others. The bottom layer abstracts away the details of the grid topology, the middle layer consists of a state-space-based search algorithm, whereas the top layer provides facilities for automatic testing, collection of statistics, and tools for importing game maps. The plan is to make this software public to encourage more realistic and uniform test-beds for pathfinding research.

In this framework, we implemented both A* and IDA* in the middle-layer, and tile-, octile-, and tex-representations of maps in the bottom layer (there is no need to include hex grids because tex grids have the same search properties while allowing a more efficient implementation).

4 Empirical Comparison

We used the aforementioned test environment to empirically evaluate A* and IDA* using different grid structures. The following map categories were used: obstacle-free maps (J5), maps with randomly placed obstacles (7?), and maps extracted from the popular computer game *Baldur's Gate II* by *BioWare (B).* For each category 120 maps were used, ranging in size from approximately 50 x 50 to 300 x 300. For each map 400 randomly chosen *start/goal* states were searched.

The result of the A* experiment is summarized in Table 2. Each entry shows the average number of nodes expanded over 48,000 searches (400 trials x 120 maps) as well as the standard error. The heuristic used for each grid representation is perfect given that no obstacles are present (e.g. Manhattan distance for tiles). We can see that the exact graph representation does not make a big difference for A*. However, any improvement is beneficial because multiple units are typically traversing a game world simultaneously, each requiring a path be found in a matter of milliseconds. For example, we can state with a high statistical significance for all three map categories that octile grids expand fewer nodes on average than the alternative grid structures.

Table 2: Summary of A* Experiments

Tile		Tex		Octile	
72.4	(±0.3)	58.5	(±0.2)	50.8	(±0.2)
493.7	(±5.0)	516.9	(±5.0)	466.7	(±4.9)
606.0	(±8.2)	588.6	(±8.0)	511.4	(±7.5)

Identical experiments were conducted for IDA*. Because of its exponential growth rate, normal IDA* performed poorly on the larger maps, often unable to find a path even after expanding millions of nodes. Instead we used a memory-enhanced variant of IDA* for our evaluation (a simplified version of the transposition table enhancement described in [2]). The result is summarized in Table 3. When comparing this result with the A* result we notice that IDA* expands many more nodes on average (because it needs to revisit nodes on each iteration). However, despite expanding more nodes, the overall run-time for IDA* was considerably less than A*\s for both empty and random maps (less work per-node). Also of interest is that tiles clearly work better than the other grid topologies when used in conjunction with enhanced IDA*. The data also shows a big performance difference between random and game maps. On game maps IDA* searches an order of magnitude more nodes than on random maps, resulting in A* being the preferred algorithm for these game maps (even when considering run-time).

Table 3: Summary of IDA* Experiments

Tile		Tex		Octile	
72.4	(±0.3)	58.5	(±0.2)	50.8	(±0.2)
1514.9	(±25.2)	2346.7	(±38.6)	1904.7	(±38.4)
19597.1	(±501.3)	28899.4	(±757.8)	21188.7	(±568.9)

5 Conclusions

Analytical results show that normal IDA* performs best on tex (or hex) grid abstraction. Unfortunately, normal IDA* is impractical for larger maps, including realistic game maps. In practice, A* performs best on octile grids and memory-enhanced IDA* on tile grids. This result is based on maps typical of computer games in terms of size and level of detail. For IDA* and A* we use the fact that only a subset of neighboring nodes need to be visited, an often overlooked fact. The result also shows that one must be careful when carrying insights gained from random maps over to "realistic" maps.

Acknowledgments

This research was supported by grants from NSERC, iCORE, and IRIS.

References

[1] P. Yap. Grid-based Pathfinding. *Lecture Notes in Artificial Intelligence,* vol 2338,44-55, 2002.

[2] A. Reinefeld and T. Marsland Enhanced Iterative-Deepening Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI),* vol 16, 701-710, 1994.