Dynamic Skeleton Based Wayfinding

Hans Guesgen and **Paul Shotbolt**[1]

Abstract. A system is proposed in which case-based reasoning principles are applied to the wayfinding problem for computer games. As case-based reasoning is thought to mimic aspects of human thinking, applying this methodology can provide more convincing non-player character behaviour. Additionally, knowledge re-use could provide performance gains through a reduction in redundant computation. Agents store a skeleton set of known paths and adapt these to solve new wayfinding problems. New solutions, along with paths obtained through communication with other agents, are reincorporated to create a dynamic, evolving case-base of skeleton paths.

1 INTRODUCTION

Most humans perform wayfinding tasks heuristically [1], whereas computer wayfinding is often approached from a mathematical search perspective. This disparity in approach leads to qualitative differences in the solutions obtained.

One goal of computer game designers is the suspension of disbelief of the player. As many games feature simulated 'human' non-player characters (NPCs), bestowing human-like behaviour to these characters is important. Differences between human and computer behaviour result in the onset of disbelief, which detracts from enjoyment of the game. Recent investigations have shown that there is still tremendous scope for improvement in the simulation of convincing human behaviour and wayfinding [2].

The aim of this experiment was to give NPC agents the ability to obtain, store and reuse data about paths in their wayfinding processes. Agents should be able to obtain path data through both derivation from level geometry and inter-agent communication. Paths should be retained in the agent's knowledge base (KB), and future wayfinding decisions should incorporate this knowledge.

This *skeleton set* of known paths should grow and change dynamically as the agent moves about the level, affecting future decisions to an increasing extent. It is hoped that as this approach approximates a more human-like perspective on the wayfinding problem [1], the agents will reflect this by displaying more human-like behaviour. Additionally, knowledge re-use may allow for more efficient wayfinding, providing performance bonuses in games where resources are traditionally scarce [3]. Interestingly, routes do not always have to be optimal or near optimal; an agent may make errors, as long as they are mistakes that a human might make under similar circumstances and are not blindly repeated. Another research goal is to allow this deeper simulation of NPC wayfinding to contribute to emergent gameplay as discussed and defined in [4].

[1] University of Auckland, Auckland, New Zealand email: hans@cs.auckland.ac.nz

2 APPROACH

Each agent stores any paths it traverses in its KB, and re-uses this knowledge in solving new problems. Cooperating agents also trade cases with each other through some communication system, providing a more realistic model of team wayfinding ability.

A query is specified in this system by a start position and an intended goal position for some agent. When given a query, an appropriate skeleton path from the case base is first retrieved, and then adapted into a solution. If no path suitable for adaptation is found in the case-base, a solution must be generated entirely.

Retrieval of an appropriate case may be approached in many ways [5]. For simplicity, a weighted combination of the following factors was used to evaluate the utility of each candidate path, *S*.

- A measure of how often *S* has been used by the agent
- The length of *S*
- Measures of how well the orientation and curvature of *S* match the orientation of the general direction of travel for the query
- An approximation of the distances from the start point to *S* and from the destination to *S*

The obvious approach for the adaptation phase is the same as that described in [1], finding connections between the start and end points, and the selected skeleton path. By selecting a suitable path as a subgoal[2] in the search process, this 'skeleton-based wayfinding' can increase the efficiency of the wayfinding process significantly. Once a candidate path *S* has been selected, most conventional wayfinding techniques can be adapted to find the missing connections from the start and goal nodes to *S*.

The A* algorithm is a well-known form of guided search that is both complete and optimal[3] [6], and has been widely used in computer games [7]. The template A* algorithm in [8] was modified to find routes between a given node and a skeleton path.

Limited communication of path knowledge between agents is accomplished by storing all agent knowledge in semantic nets[4] as described in [6]. An agent, *A*, may then probabilistically query nearby peer *P* about some node *o* in *A*'s semantic net. *P* responds by returning a subgraph of *P*'s semantic net obtained by a breadth-first search outwards from *o*. This subgraph is then reintegrated into *A*'s semantic net. Skeleton path data can then be propagated between agents in a semi-realistic fashion, causing each agent's case base to grow and change dynamically through experience and interaction with other agents. As their case base begins to store an increasing number of useful paths, their wayfinding abilities should improve. As the CBR approach suffers when the case base is too large, a 'forgetting' procedure is used to remove older, less used paths from the KB.

[2] The use of nodes as subgoals in Island search provides similar advantages.
[3] Provided that the heuristic function *h* is admissible.
[4] Conceptual graphs would be equally appropriate.

3 RESULTS

The test system was built using the Torque Game Engine [9]. Tests were carried out at different waypoint resolutions over two game-like environments, a simple outdoor environment, and another with more complex geometry, non-axis-aligned buildings and a higher obstacle count. Results indicate that the approach can provide some benefits in terms of simulating human behaviour.

Due to the smooth curvature of the landscape and simple features in test environment *A*, agents did not have significant trouble navigating the terrain at a waypoint resolution of 50x50 units, and agents visibly re-use past paths in their solutions to subsequent queries. Additionally, a preliminary analysis shows that the wayfinding process appears to complete quicker when using the dynamic skeleton based system, rather than the original A* system, provided there are a significant number of paths in the agent's KB. At higher waypoint resolutions, these time savings appear more significant. Agents also do not often make poor choices in their selection of which skeleton path to adapt to solve a query. Both the complexity savings and the path selection are very difficult aspects of the system to test, and a more thorough investigation would be appropriate for future work.

Due to inter-agent communication, certain paths become much more frequently used than others, and Kuipers' [1] positive feedback loop appears to be emulated to some extent. This preference for common routes, and while not absolutely convincing, still appears more reminiscent of the human school of problem solving than the non-skeleton based wayfinding system.

Test environment *B* was more complex, but by no means provides a complete evaluation. Results showed that certain cases arise where an agent will take an extremely poor choice of route, as illustrated in Figure 1.

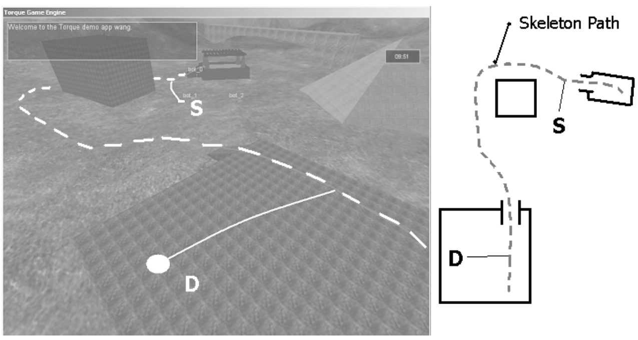

Figure 1. An illustration of the poor choice problem. Here, the agent decides to use a known skeleton path (dashed). It takes the most direct route to that path from S, and follows this path before departing towards destination D. The resulting route is significantly worse than optimal.

The choice that the agent makes here is far from optimal and hence undesirable. Better adaptation might result in the agent choosing a better rendezvous with the skeleton path. However, observe that skeleton path selected is already significantly worse than optimal, even if travelling between two points directly on each end of the skeleton path itself. This skeleton path, then, does not have high utility and should not be in the case-base.

Curiously, the agents performed well for a moderate time period, but in longer sessions performance degraded significantly. Initially, agents do much of their wayfinding through A*, guaranteeing that many new paths are optimal. As these cases are used to solve new queries, less optimal, derivative paths are created. These derivative paths are added to the case base, in process that initiates a positive feedback loop for poor path choices, causing degenerating performance over time.

Compounding this is another problem involving the 'forgetting function'. The most optimal paths created near the beginning of the session are shorter, likely to be used, and hence become early candidates for the 'forgetting system'. In its current form, the forgetting system may be a liability overall.

4 CONCLUSIONS

Overall, while the system was not as immediately successful as initially hoped, the approach is now more thoroughly understood, however from the many intrinsic problems it appears that the entire 'dynamic skeleton-based wayfinding system' would need an overhaul to make it viable. These problems range from moderate flaws, such as imperfections in the 'path selection' system, to fundamental flaws such as the conflux of factors that cause positive feedback to more derivative path choices, resulting in a continuously degenerating case base.

The degenerating case base issue might be solvable by only storing, say, optimal and first-level derivative paths. Genetic algorithms may be able to provide better weights for the case retrieval process. With further research, agents may be able to re-use skeletal path information to help predict opponent movements.

There have also been a number of successes with this investigation, including the occurrence of several emergent phenomenon, e.g. the positive feedback loop present when agents interact with each other. Information about more useful paths is propagated between agents, and this encourages more interaction between other agents in future.

As efficient algorithms are critical for computer games, more empirical testing of this system is important, however, there is significant work to be done resolving critical issues with the approach before efficiency should be examined in depth.

REFERENCES

[1] B. Kuipers, 'The Skeleton in the Cognitive Map: A Computational Hypothesis', *Space Syntax: Proceedings of the Third International Symposium,* (2001).

[2] J. E. Laird, 'It knows what you are going to do: Adding anticipation to a QuakeBot', *Proceedings of the Fifth International Conference on Autonomous Agents*, 385-392, (2000).

[3] C. Fairclough, M. Fagan, B. Mac Namee, and P. Cunningham, 'Research Directions for AI in Computer Games', *Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science,* 333-344, (2001).

[4] H. Smith, *The Future of Game Design: Moving Beyond Deus Ex and Other Dated Paradigms,* Speech given to the Multimedia International Market, 2001. [Retrieved from http://www.igda.org/articles/hsmith_future.php on 16 Nov. 2003]

[5] B. Bartsch-Spörl, M. Lenz, and A. Hübner, 'Case-Based Reasoning Survey and Future Directions', *Knowledge-Based Systems, Lecture Notes in Artificial Intelligence,* Springer Verlag, 67-89, (1999).

[6] S. Russell, and I. Norvig, *Artificial Intelligence: A Modern Approach,* 5-6,96-106, 306-325, Prentice Hall Inc., 1st edn., 1995.

[7] K. Sergent, *Smart Guys,* Morrowind developer diaries, 2003. [Retrieved from http://www.elderscrolls.com/codex/team_smartguys.htm on 1 Feb. 2004]

[8] J. Heyes-Jones. *A* algorithm tutorial*. STL A* Template C++ source. [Retrieved from http://www.geocities.com/jheyesjones/astardownloads.html on 5 May 2003]

[9] Garage Games Inc., *The Torque Game Engine*, 2003. [Retrieved from http://www.garagegames.com on 8 Jan 2003]