Towards Immersive 3D Visualisations of Game AI Algorithms

Edward J. Powley[1]

Abstract. We propose, as a promising future direction of research and development, the use of immersive technologies (particularly Virtual Reality (VR)) to visualise the operation of game AI algorithms. This has obvious applications when developing AI agents for VR experiences, but the affordances of VR may also provide wider insights and more generally applicable tools.

1 Introduction

Visualisation is one of the most important tools of the researcher or developer working in game AI [3]. Visualising the operation of an AI algorithm helps to identify bugs, observe the effects of tweaking parameters, and gain insight into the operation of a system.

Monte Carlo Tree Search (MCTS) [7, 4] is a game tree search algorithm which has proven particularly successful in many challenging game AI domains and decision problems [2]. MCTS requires only a forward simulation model, and is an anytime algorithm which can generally yield a reasonably good strategy after a short amount of computation time (though is guaranteed to converge upon an optimal strategy as the computation time tends to infinity). MCTS is a reinforcement learning algorithm [19], however its basis in state-action trees makes it easier to visualise than some other machine learning algorithms.

The principles of effective visualisation of data on a 2-dimensional page or screen have been well studied [17]. Technologies such as Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR) introduce new possibilities for visualisation, allowing for immersive 3D renderings of complex data. These possibilities are beginning to be explored across a variety of domains. We propose that immersive visualisation technology holds a great deal of promise for developers to explore, refine and understand game AI techniques, both for interactive VR/AR/MR experiences and for more traditional screen-based games.

2 Visualising game AI

Visualisation of AI algorithms is useful from a software engineering point of view. Champandard [3] identifies three key benefits of visualising AI systems during game development: ensuring code correctness, identifying bugs, and to assist tweaking and tuning.

[1] MetaMakers Institute, Games Academy, Falmouth University, UK. Email: `edward.powley@falmouth.ac.uk`

Less commonly, AI visualisations can be used as a gameplay mechanic. In the stealth game *Third Eye Crime* [6], the game display is overlaid with a heat-map representing the enemies' beliefs regarding the location of the player. This allows the player to predict the behaviour of the enemies, leading to unique gameplay possibilities. Treanor et al [16] identify AI visualisation as a game design pattern for foregrounding AI, however it is relatively under-explored in commercial games.

3 Visualising MCTS

Visualisation is also useful when developing new AI methods in a research context. This has the benefits identified by Champandard [3], and additionally can lead to new insights into how the algorithm works. Figure 1 shows an example of visualisations created by the author for the Node Recycling MCTS algorithm described in [12]. These give some insight into the operation of the algorithm, though this is limited by the restrictions of reproducing the visualisation in a static form. The author also developed a dynamic visualisation which shows the process "live" as the search progresses, which gives much greater insight. Figure 2 shows a different visualisation of the same algorithm, showing the relative frequencies with which available moves are explored and how the identity of the "best" move changes as the search progresses. This visualisation *is* effective on the static page: the information it displays is one-dimensional, allowing the second dimension to be used for time.

Figures 3 and 4 show two other visualisation applications developed by the author in order to understand and debug implementations of MCTS. TreeViewer (Figure 3) displays an MCTS search tree, loaded from disk in a simple XML file format. It allows nodes to be sorted, expanded, collapsed and interrogated for various property values. It is possible, though cumbersome, to achieve some of these tasks using the built-in debugger in an IDE such as Microsoft Visual Studio, however a custom application allows the developer much more control over how the tree is laid out on the screen.

Figure 4 shows an interactive demo application which allows a user to play a range of simple board games against an MCTS opponent. The search tree built by the MCTS player is shown on the right-hand side of the screen, and builds in real-time as the search progresses. The colour of the lines shows the average reward for the corresponding node in the tree, and the thickness of the line shows the number of visits, giving an at-a-glance picture of the most important quantities in the search tree. For Connect 4, the visualisation also has the

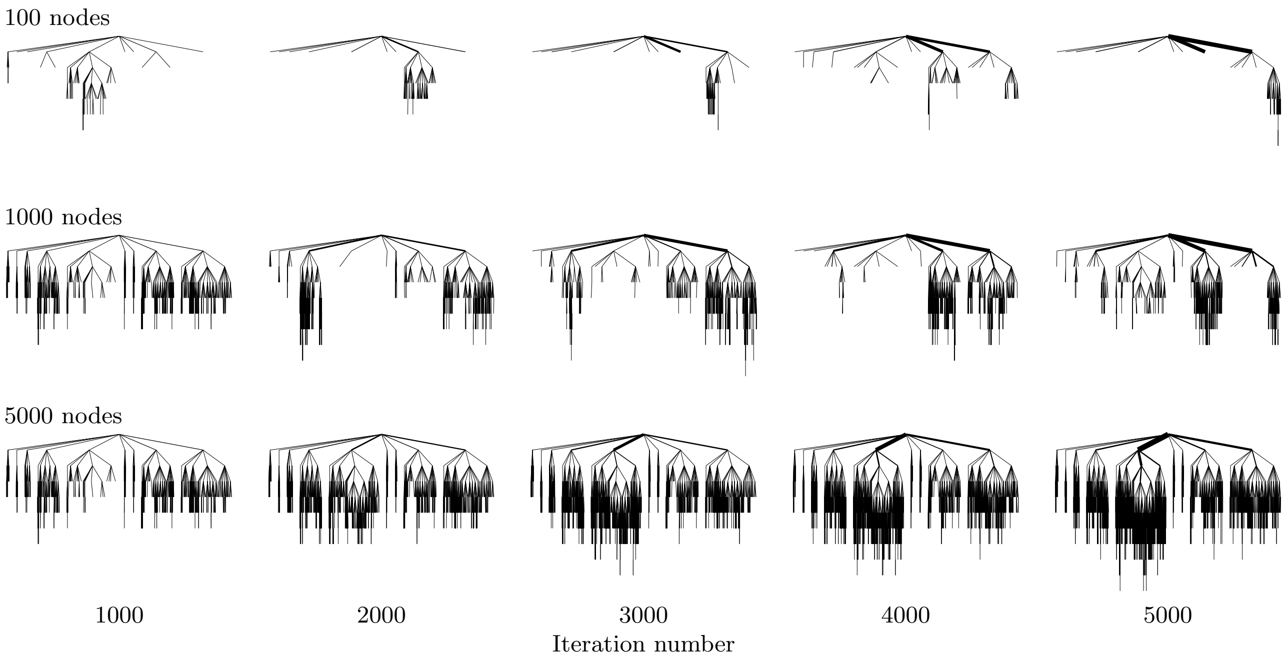

Figure 1. Sample visualisations of the Node Recycling MCTS algorithm described in [12], showing how the search tree is built and destroyed over time under different parameter settings.

nice property that the children of a node from left to right correspond to the moves of the game, i.e. placing a counter in a column from left to right, making it easy to see how the tree corresponds to how the game plays out. The author and his colleagues frequently use this demo application in outreach events, in teaching and in and other presentations, and find it to be an extremely effective aid to explaining how the algorithm works.

Figure 5 shows a screenshot from the Multi-Objective Physical Travelling Salesman Problem (MO-PTSP) [11]. This is a game in which a spaceship must be piloted around a maze, passing through a number of checkpoints in any order. The screenshot shows an MCTS-based controller [13] playing the game. Overlaid onto the display of the game environment are the expected trajectory of the spaceship according to the most explored line of play in the search tree (visible as a green line protruding from the ship), and the distance map used to provide heuristic guidance to the search (visible as white contour lines). Though quite simple, these visualisations provided much insight when developing and tweaking the agent, and it is difficult to imagine the final agent being as effective were it not for this.

4 VR/AR visualisations

In 2000, when VR technology was much less sophisticated than today, van Dam et al [18] discussed the promise of VR for scientific visualisation and highlighted some examples of its use. More recently, VR and other immersive technologies have been applied to the visualisation of graphs [5], molecular structures [15], medical data [10] and urban planning [8], among others. The aesthetic appeal of VR visualisations and the sense of immersion and presence they afford often blurs the line between visualisation and art, as in the Mutator VR project [14] for example.

AR has seen similarly wide deployment, particularly in visualisations for archtecture and engineering; see [9] for a recent survey, and [1] for an example of commercial deployment.

As van Dam et al [18] point out, the benefit of scientific visualisation is to exploit the human brain's aptitude for visual processing and pattern recognition. VR brings several benefits over screen-based representations. The addition of an extra spatial dimension (though simulated through stereoscopic displays and head tracking) allows larger and more complex data sets to be visualised without clutter. The advanced motion tracking and haptics of modern input devices such as the HTC Vive, Oculus Touch and Leap Motion allow the provision of more intuitive and expressive ways of interacting with the data. AR has the obvious benefit that visualisations can be overlaid onto real environments; in contrast, VR lends itself to overlaying information onto simulated or reconstructed environments, or removing the environment entirely.

5 Future directions

Visualisations of game AI are often most effective when overlaid onto the game world [3]. If the game world is experienced in VR/AR rather than on a screen, it makes sense for the overlay to be present in the virtual/augmented space as well.

However, VR visualisations have potential even for applications of AI outside of VR/AR. When visualising MCTS trees, a limiting factor tends to be the number of nodes that can fit onto the screen before the information becomes too dense to be useful. The extra spatial dimension added by VR, as well as the potential to visualise information at room-scale or even larger rather than confined to the page or screen, could

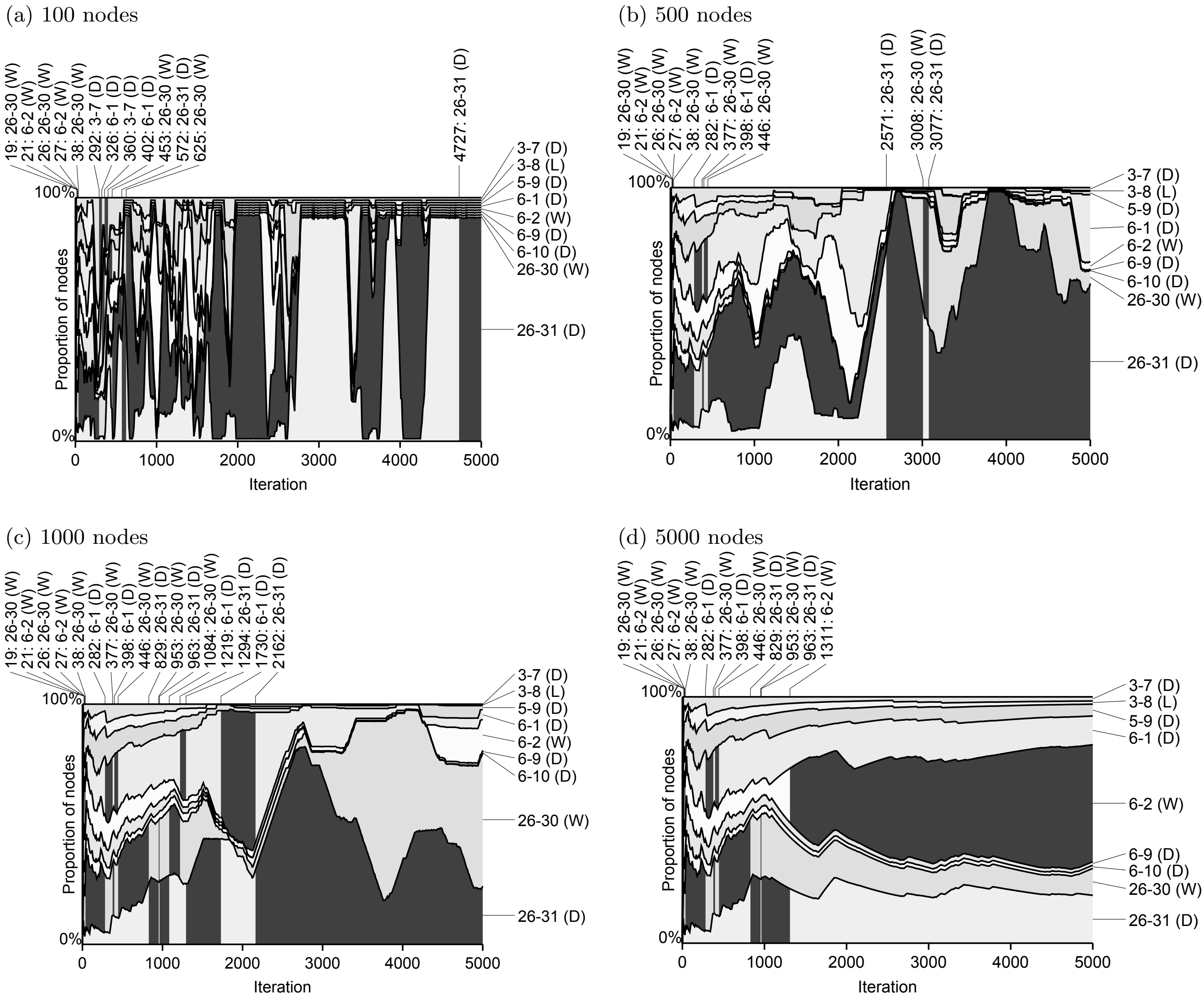

Figure 2. Sample visualisations of the Node Recycling MCTS algorithm described in [12], showing how the visit frequencies of moves change over time. The x-axis represents the progress of the MCTS algorithm. Each vertical cross-section of the graph shows the relative sizes of the trees below each move from the root. The dark region shows the move with the most visits, i.e. the move which would be selected if the search were halted at this point.

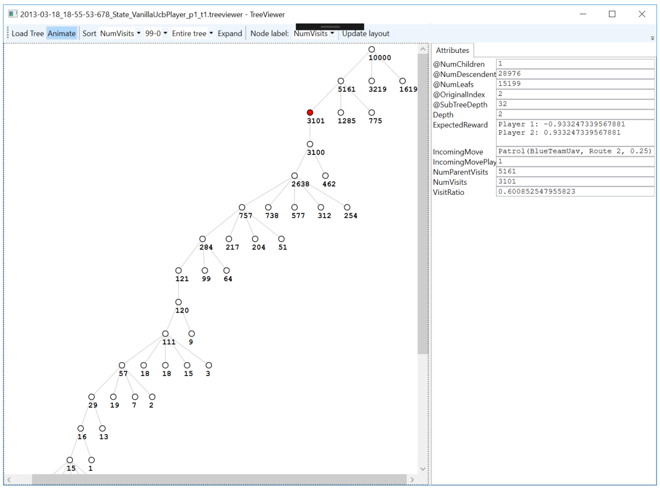

Figure 3. Screenshot of TreeViewer, an application for exploring MCTS search trees.

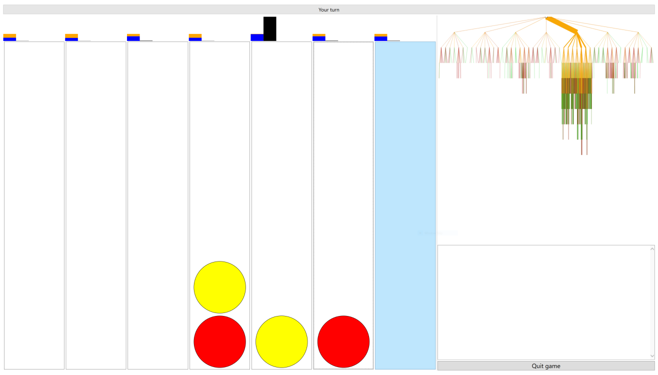

Figure 4. Screenshot of InteractiveDemo, an application which visualises the MCTS tree built by an AI opponent in the game Connect 4.

Figure 5. Visualisation for an MCTS-based agent in the Multi-Objective Physical Travelling Salesman Problem.

allow for much larger trees to be visualised effectively. Graph visualisation and interaction techniques like those proposed by Erra et al [5] could also prove useful.

The MCTS visualisations described in Section 3 are functional rather than aesthetically pleasing, however there is still an appeal to watching the trees grow and evolve in real-time. VR visualisations lend themselves naturally to crossover with the visual arts, and the automatic sense of presence and immersion given by modern VR hardware means that visualisations that would look relatively unsophisticated on a screen can look much more impressive and appealing. More attractive and engaging visualisations are beneficial in scientific outreach and education, and may also fit better with the high level of visual polish expected of video games. This may lead more game developers to treat AI visualisations not merely as a debugging tool but as a potential source of game mechanics, leading to wider exploration of Treanor et al's [16] "AI is visualised" design pattern.

REFERENCES

[1] Aviad Almagor. Mixed reality for the AEC industry: Extending Trimble's product capabilities with Microsoft HoloLens. `https://buildings.trimble.com/sites/buildings.trimble.com/files/white_papers/Trimble_White_Paper_Mixed_Reality_for_The_AEC_Industry.pdf`, 2017.

[2] Cameron Browne, Edward J. Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton, 'A Survey of Monte Carlo Tree Search Methods', *IEEE Transactions on Computational Intelligence and AI in Games*, **4**(1), 1–43, (2012).

[3] Alex J. Champandard. Visualization of AI and gameplay: 5 useful examples in practice. `http://aigamedev.com/open/tutorial/rushing-bases/`, 2011.

[4] Guillaume Maurice Jean-Bernard Chaslot, Jahn-Takeshi Saito, Bruno Bouzy, Jos W. H. M. Uiterwijk, and H. Jaap van den Herik, 'Monte-Carlo Strategies for Computer Go', in

proceedings of BeNeLux Conference on Artificial Intelligence, pp. 83–91, Namur, Belgium, (2006).

[5] Ugo Erra, Delfina Malandrino, and Luca Pepe, 'Virtual reality interfaces for interacting with three-dimensional graphs', *International Journal of HumanComputer Interaction*, 1–14, (2018).

[6] Damián Isla, 'Third Eye Crime: building a stealth game around occupancy maps', in *proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, (2014).

[7] Levente Kocsis and Csaba Szepesvári, 'Bandit based Monte-Carlo Planning', in *proceedings of European Conference on Machine Learning*, eds., Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, pp. 282–293, Berlin, Germany, (2006). Springer.

[8] Xiaoming Li, Zhihan Lv, Weixi Wang, Baoyun Zhang, Jinxing Hu, Ling Yin, and Shengzhong Feng, 'WebVRGIS based traffic analysis and visualization system', *Advances in Engineering Software*, **93**, 1–8, (2016).

[9] Medhi Mekni and André Lemieux, 'Augmented reality: applications, challenges and future trends', in *proceedings of International Conference on Applied Computer and Applied Computational Science (ACA-COS)*, pp. 205–215, (2014).

[10] Koosha Mirhosseini, Qi Sun, Krishna Chaitanya Gurijala, Bireswar Laha, and Arie E. Kaufman, 'Benefits of 3d immersion for virtual colonoscopy', in *proceedings of IEEE VIS International Workshop on 3DVis*, pp. 75–79, (2014).

[11] Diego Perez, Edward J. Powley, Daniel Whitehouse, Spyridon Samothrakis, Simon M. Lucas, and Peter I. Cowling, 'The 2013 Multi-Objective Physical Travelling Salesman Problem Competition', in *proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 2314–2321, (2014).

[12] Edward J. Powley, Peter I. Cowling, and Daniel Whitehouse, 'Memory bounded Monte Carlo Tree Search', in *proceedings of AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, pp. 94–100, (2017).

[13] Edward J. Powley, Daniel Whitehouse, and Peter I. Cowling, 'Monte Carlo Tree Search with Macro-Actions and Heuristic Route Planning for the Multiobjective Physical Travelling Salesman Problem', in *proceedings of IEEE Conference on Computational Intelligence in Games (CIG)*, pp. 73–80, Niagara Falls, Ontario, Canada, (2013).

[14] Lance Putnam, William Latham, and Stephen Todd, 'Flow fields and agents for immersive interaction in mutator vr: Vortex', *Presence: Teleoperators and Virtual Environments*, (2016).

[15] A. Salvadori, D. Licari, G. Mancini, A. Brogni, N. De Mitri, and V. Barone, 'Graphical interfaces and virtual reality for molecular sciences', in *Reference Module in Chemistry, Molecular Sciences and Chemical Engineering*, Elsevier, (2014).

[16] Mike Treanor, Alexander Zook, Mirjam P Eladhari, Julian Togelius, Gillian Smith, Michael Cook, Tommy Thompson, Brian Magerko, John Levine, and Adam Smith, 'AI-based game design patterns', in *proceedings of the International Conference on Foundations of Digital Games (FDG)*, (2015).

[17] Edward R. Tufte, *The Visual Display of Quantitative Information*, Graphics Press USA, 2001.

[18] Andries van Dam, Andrew S. Forsberg, David H. Laidlaw, Joseph J. LaViola, Jr., and Rosemary M. Simpson, 'Immersive VR for scientific visualization: A progress report', *IEEE Computer Graphics and Applications*, **20**(6), 26–52, (2000).

[19] Tom Vodopivec, Spyridon Samothrakis, and Branko Šter, 'On Monte Carlo Tree Search and reinforcement learning', *Journal of Artificial Intelligence Research*, **60**, 881–936, (2017).