Computational Science Technical Note **CSTN-105**

Behavioural AI in Movies and Games: The Last 20 Years

A. P. Gerdelan

2010

Computerised actors using agent-driven models have been used heavily in entertainment (in both film and interactive games) and serious simulation for decades. Simulated environments have become more complex and approach the information "noise" levels of the real world. This presents the agent with more more decisions to make with more input variables. As graphical realism has improved, audiences now also expect the behaviour of the actors to be more perceptually convincing. The problem of navigating or steering actors through these simulated worlds has existed, albeit with differing requirements, throughout the history animation. We will look at some of the popular approaches to agent steering in entertainment, and also discuss some emerging technologies that make use of evolutionary algorithms.

Keywords: artificial intelligence; games; movie; simulation

BiBTeX reference:


```
@TECHREPORT{CSTN-105,
        author = {A. P. Gerdelan},
        title = {Behavioural AI in Movies and Games: The Last 20 Years},
        institution = {Computer Science, Massey University},
        year = {2010},
        number = {CSTN-105},
        address = {Albany, North Shore 102-904, Auckland, New Zealand},
        month = {August},
        keywords = {artificial intelligence; games; movie; simulation},
        owner = {kahawick},
        timestamp = {2009.12.16},
        url = {http://www.massey.ac.nz/~kahawick/cstn/105/cstn-105.pdf}
}
```

Technical Report CSTN-105

A Brief History of Motion Control in Animation

A.P. Gerdelan
IIMS, Massey University, Albany, New Zealand, and
GV2, Trinity College Dublin, Dublin, Ireland
Email: gerdelan@gmail.com

April 2009

Abstract

Computerised actors using agent-driven models have been used heavily in entertainment (in both film and interactive games) and serious simulation for decades. Simulated environments have become more complex and approach the information "noise" levels of the real world. This presents the agent with more more decisions to make with more input variables. As graphical realism has improved, audiences now also expect the behaviour of the actors to be more perceptually convincing. The problem of navigating or steering actors through these simulated worlds has existed, albeit with differing requirements, throughout the history animation. We will look at some of the popular approaches to agent steering in entertainment, and also discuss some emerging technologies that make use of evolutionary algorithms.

Keywords: motion control; artificial intelligence; games; boids; flocking; crowds.

1 Introduction

Autonomous computer-controlled entities can make excellent extras in film scenes, entertaining opponents in computer games, and to some extent useful autopilots or robotic drivers in real vehicles. The problem domain in each of these applications is similar; an actor or vehicle has to be moved through a noisy environment. The desired motion is some combination of:

- realistic; a simulation of real movement patterns
- convincing; what an audience perceives to be realistic
- efficient; minimising energy or reducing collisions
- entertaining; motion may aim for cartoon-like motion

The distinction and balance between these four aims is often unclear in the literature; the aim is not always well understood or clarified by researchers. It is very easy to make false assumptions that efficient motion is realistic motion, or that realistic motion is convincing or desirable in games or film. From this complicated mixture of motion and steering aims several distinct schools of thought have emerged:

- The principle of least effort (PLE) (aim: combination)
- Data-driven crowds (aim: realistic)
- Simulation based on perceptual studies (aim: convincing)
- Level of detail (LOD) crowd behaviour (aim: balance efficient/convincing)

A popular approach to motion control for crowd simulation is based on the principle that pedestrians will always try to minimise effort in going from A to B; often referred to as the *principle of least effort* (PLE) [1], and that introducing individual variation or detailed autonomy into the control algorithm is less important because the overall or aggregate motion flow will be realistic and produce emergent effects that are consistent with collected real-world data.

Controllers designed for realism tend to use a data-driven approach to evaluate the effectiveness of the controllers, and are used for architectural design planning applications such as the Sydney 2000 and London 2012 Olympic Games [2]. Modern data-driven studies show that people in real crowds in fact do not behave like the established efficiency-driven automaton models [3, 4], and that there is also a high degree of variation of movement of individuals within a crowd based on age, gender, disability, emotional state, and other factors that is visible in aggregate motion [4]. This makes data-driven motion controllers very difficult to validate, and renders simulations driven by data from a single region non-generalisable [5] to other populations.

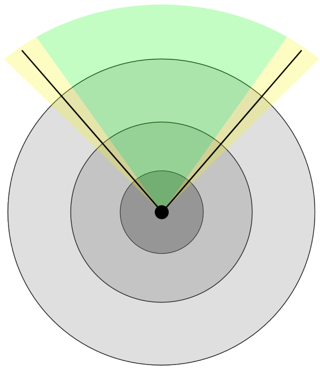

Figure 1: A spatial representation of classification of discrete levels of detail based on a camera position and orientation. Darker areas are awarded higher levels of detail. Information from outside the frustum (coloured area) is discarded. Image reproduced with permission.

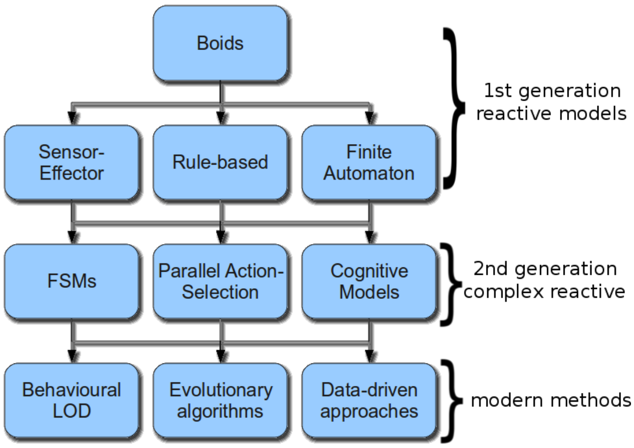

Figure 2: The evolution of behavioural control in animation. Popular methods for agent control from Boids to Finite State Machines (FSMs) to Behavioural Level-of-Detail (LOD) algorithms.

Motion control and animation that aims for perceived realism can be evaluated by psycho-visual tools and user-studies. Such works aim to evaluate the perceptual importance of new motion techniques and to find thresholds of perceptual importance in order to optimise techniques. Recent works have studied the importance of formation in believable crowds [6], the amount of variety required within a crowd for it to be believable [7], and compared pedestrian motions using perception-based metrics [8].

The level of detail (LOD) principle [9] is used in 3D graphics to reduce the rendered detail of objects that are deemed perceptually less important [9, 10]; which usually this means further from the camera (see Figure 1). The idea is that the simulation is then more efficient and thus more detailed or more numerous objects can be simulated. This principle has also been applied to behavioural detail [11]. The computational time slice allowed for steering vehicles or actors in perceptually important zones is then larger than those off-screen or in the background, making this a technique that aims to balance computational efficiency with perceived realism.

Controllers that aim for entertaining motion are evaluated by artists, film directors, and preview audiences [12, 13]. Entertainment-driven motion is very difficult to design for as realistic motion may be not actually desirable in entertainment. Designers of such controllers take great care in providing suitable “knobs and dials” that allow the motion to be customised during preview screenings [12]. A special cartoon-like motion was developed for a character in the Wall-E film, using a customised spring equation [12]; a technique that could not have been based on a realistic model. MASSIVE [14] software was used in both the Wall-E film and the Narnia film [13] for motion of large groups of characters. Amongst the many techniques used by MASSIVE, the reactive motion uses physics-based objects that encompass a space larger than the characters, which when colliding give the impression that the characters are ’avoiding’ each other, when computationally their physics bodies are colliding and sliding around each other [14].

2 Overview of Key Algorithms

Approaches to solving motion control and steering problems for animation have evolved in several distinct generations. Donikian (2009) [15] classifies reactive behavioural algorithms (of which motion control is one) into two previous generations, with proposed cognitive science models considered part of a new generation. The first generation in this classification is divided into three parallel streams of research; sensor-effector models, behaviour rules models, and finite automaton models. In this paper we extend this classification system, by considering fuzzy, evolutionary, data-driven and level of detail (LOD) algorithms to be part of the latest generation of research. Figure 2 illustrates our extension of Donikian’s model.

2.1 Boids and Flocking

Reynolds’ 1987 SIGGRAPH article [16] introduces an animation system for bird-oid (bird-like) actors called boids. The article was accompanied by a short film featuring boids, shown in Figure 3. The key idea of boids is to approximate the real world motion of groups of animals (flocks, herds, and schools) by simulating each individual within the group, and having the overall (aggregate) motion of the group emerge as a result. Each individual is moved using a balance of attractive and repellent forces. The actors in boid system are also able to work within 3D physics constraints. The flight dynamics required for realistic bird ani-

Figure 3: A screen capture from the SIGGRAPH Electronic Theatre short film Stanley and Stella in: Breaking the Ice (1987), featuring boid-driven birds and fish.

mation and motion are detailed in the article.

Boids extends particle systems [17] by adding orientation, collision avoidance, and semi-autonomous motion. Each *actor* (a concept now synonymous with *agent* [18]) is given a limited perception of the world with which to make steering decisions. Reynolds finds that if the actors are given complete information (the locations of all of the actors in the flock) then the aggregate motion of the flock is unrealistic or produces undesired centrifugal formations [16]. For this reason each actors are given only the location of its immediate neighbours in the flock, and a limited field of view as depicted in Figure 4.

Reynolds' *flocking* mechanism has become a standard in computer animation. Flocking has three components:

1. Separation: actors steer to avoid neighbours
2. Alignment: actors steer toward the average heading of neighbours
3. Cohesion: actors steer toward the average position of neighbours

Reynolds does not detail how the various forces are balanced other than to suggest the use of a dampening function [17]. This is no doubt a constraint satisfaction problem that requires quite some tedious manual trial and error (a problem common to many motion controllers).

The flocking mechanism is easily confused when presented with more than the most simple geometric obstacle problems. Reynolds states that this is due to the lack of a detailed perception model of virtual computer vision [17], but we now know that the problem is more complex than this, and requires inclusion of a path-planning algorithm at a higher level. Finding a good balance between reactive steering methods such as the flocking mechanism, and path-planning navigation methods which guide them is an active research topic [11, 19].

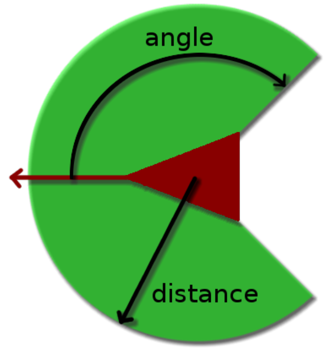

Figure 4: A boid's neighbourhood is limited to a distance, and an angle from its heading orientation, such that it only considers immediate neighbours in its steering behaviour.

An adapted copy of Reynolds' original boids software was used to animate swarms of bats and flocks of penguins in the 1992 action movie *Batman Returns*. The theory also contributed to the emerging Artificial Life (AL) [20, 21] field, with the aggregate group behaviour an example of the AL principle of emergence.

Boids has also been notably extended to use fuzzy controllers in place of the original mathematical decision-making model [22]. The authors found that fuzzy logic controllers were a solution to dealing with environment noise (many variable inputs to the decision process) and enabled more human-like or animal-like decision making.

2.2 Helbing's Crowds

Another notable work is Helbing's 1995 article on crowd simulation "Social force model for pedestrian dynamics" [23]. The article appears in the Physical Review E journal, which is home to key works in the area of traffic microsimulation [24], that is, traffic simulations based on fluid dynamics models which model gas and liquid flows but simulate the individual vehicles within the flow, rather than just the overall pattern of movement [24–26]. Helbing adapted the microsimulation model and applied it to the movement of pedestrians in crowds. The model is a basic attractive/repellent force model as seen in earlier the Boids [16] flocking mechanism. Interestingly Helbing has not drawn on Reynolds' attractive/repellent force model for computer animated animal groups, despite its similarities.

In the article Helbing gives us basic simulation layouts but no experiment result data which would indicate the effectiveness of the system in terms of crashes or expediency. Like Reynolds' Boids article it is then really a proposal rather than a scientific study, but along with

Boids it has since been adopted as a benchmark system for evaluating the effectiveness of new crowd simulation systems [11, 15, 19]. At this stage we can see that the emerging crowd animation field had still not crossed domains or levered the considerable body of similar research that had been done in psychology or robotics where the limitations of such approaches had already been analysed in detail [27]. Helbing's agent architecture; the "process leading to behavioural changes" [23] is almost identical to the concept of the Belief-Desire-Intention (BDI) agent model introduced by Bratman in 1987 [18], but without providing any of the concrete implementation details that BDI-based hierarchical architectures had developed for robot applications [28]. The concept of planning (of paths) or higher-level behaviour has not been given any special consideration; thus the system is completely reactive and therefore has similar limitations to Reynolds' flocks, as discussed in the previous section. It is not until Reynolds' later works [29] that forward-planning ideas are given serious consideration within the crowd simulation field.

The article makes much mention of its relevance to empirical data, and is built on the assumption that the emergent flow movement of real crowds is predictable, and gas or fluid-like. This assumption has been carried by the following generation of crowd simulations, but evidence of this correlation is not provided other than to cite a 1970 comparison of pedestrian flow data to the Navier-Stokes equation, and two of eight references to Helbing's own works which introduce a similar study with Boltzmann-like models. Modern data-driven models [4] contradict many of these claims; which find that the movement of pedestrians in crowds is indeed chaotic, containing many "curious" or otherwise inefficient patterns of movement which are not predictable by gas and fluid-dynamic equations.

Some emergent behaviour arrives from Helbing's crowd simulations; lane-following in open spaces, which is now widely regarded as unrealistic and undesirable [11], and oscillating direction changes at intersections which Helbing points out is a limitation of the approach.

For large-scale crowd simulations a Helbing-like particle or fluid model focusing on aggregate motion is still by far the most computationally efficient approach for producing massive crowds of animated pedestrians [30].

3 Evolving Motion Controllers

There are two basic problems with agent controllers that act in complex environments;

- Control systems have a lot of functions, variables, and rules that need optimising
- Behaviours may need to adapt to change in the environment

Therefore designing motion controllers by hand is a very difficult task, and in a stochastic environment there is usually no guarantee that they are optimal. Hand-designed controllers are also not able to adapt to unexpected change in conditions. Evolutionary algorithms have the potential to solve either of these problems. These are algorithms that are based on biological "survival of the fittest" type selection. The mechanism behind this is that the agents can try a range of behaviours, evaluate their effectiveness, keep the best behaviours, and use them to generate a new set. This should lead to a kind of self-tuning controller.

There are two categories of evolutionary motion controller;

1. Static tuning: Continually tries to solve a fixed (static) problem in batches.
2. Dynamic learning: Learns "on the fly"; adapts its behaviour continuously as it moves.

The static approach tunes a controller against a fixed problem. This approach is unable to solve the problem of adapting to change. Static tuning is characterised by batches of evaluation runs that are done prior to "final" implementation. A good example of this approach is Reynolds' 1994 algorithm for evolving corridor-following motion [31], where it was initially found that learning a fixed problem in a strongly-objective evaluation forced the agent to learn the quirks of that exact problem such that the resulting behaviour did not generalise to similar problems [31] (over-training). Artificial noise ("jitter") was injected into the training to improve the generalisation of the result.

The corridor-following approach was extended to automatically discover agents that would learn how to play the game of "tag" [32], which moves more into the dynamic learning area. Reynolds' discusses the potential of using physically simulated vehicles (with mass, momentum properties etc.) as being a future work of merit as it requires the optimisation of a more complicated control programme [32].

There are very few dynamic learning algorithms for motion control, and self-adapting motion is regarded as a "holy grail" problem due to the complexity of controller design required, however, special mention in this category must be given to the rtNEAT (rtNEAT) algorithm.

3.1 rtNEAT

A significant recent advance made possible by more powerful computer hardware is the discovery that machine learning techniques can be adapted to run in real-time, evolving vehicle controllers in a matter of minutes. A recent academic work of note in this area is the rtNEAT algorithm [33] which uses an adapted genetic neural network (GNN) (the NEAT algorithm) to perform very fast learning by augmenting a neural network (NN)). rtNEATwas implemented in a research-orientated video game called NeuroEvolving Robotic Operatives (NERO) [34] which used a user-in-the-loop approach to learning. In NERO the user (a player) directs the training of a team of GNN-enabled characters by changing the training environment itself in real-time, and by adjusting a punishment-and-reward scheme (reinforcement learning). The characters are able to learn reactive movement, basic path planning, and some basic military-type tactics in less than an hour [35, 36], with the created behaviour unique to the user's input [37].

The most noteworthy feature of rtNEAT as implemented in NERO is the treatment of fitness evaluation. Defining a good fitness function is the paramount problem of all genetic systems. A fitness function is responsible for defining a problem to be solved by a genetic algorithm (GA) in clear, mathematical terms. A fitness function that is too simple or too complex will prohibit or confuse the learning process. The weighting given to different inputs to the fitness function will affect the priority of tasks; the ideal weighting is usually not known prior to testing. NERO addresses these problems by simply passing them onto the lap of the human user who acts as a training supervisor. The advantage of this approach is that the fitness function itself is dynamic; the supervisor can adjust the reinforcement learning to learn additional behaviours [38]. The disadvantage is that the user needs to be present during the whole training process; therefore this approach is only suitable for a selected sub-set of training applications.

Because rtNEAT is built on GA and artificial neural network (ANN) technology it has all of the advantages and disadvantages of these algorithms; NNs are a black box; it is very difficult to dissect or manually tweak the resulting solution (the network itself) after training. Networks will almost certainly not solve a given problem as well as a specific hand made solution taking all problem features into account. The black box nature of neural networks can become a major frustration to designers seeking to improve the problem-solving efficiency of the network. With these limitations taken into account we can say that the designers of rtNEAT have found an ideal application in the NERO game; a user-driven fitness evaluation scheme to address the fitness design problem, a stochastic and hard-to-specify problem domain which is ideal for neural-network learning, where the outcome is the effectiveness of a competitive team of characters versus another GNN team with its own training, so even if the neural network is not an ideal solution to a problem, it is at least on an even playing field.

References

[1] Stephen Guy, Jatin Chhugani, Sean Curtis, Pradeep Dubey, Ming Lin, and Dinesh Manocha. PLEdestrians: A Least-Effort Approach to Crowd Simulation. In *Symposium on Computer Animation (SCA'10)*. Eurographics/ACM SIGGRAPH, July 2010.

[2] Sébastien Paris, Jean-Louis Berrou, and James Amos. Legion: genesis of a commercial crowd simulator. In *Computer Animation and Social Agents (CASA'10)*, Saint-Malo, France, June 2010.

[3] Linda C. Malone, D.J. Kaup, Rex Oleson, Mario Rosa, Florian Jentsch, T. L. Clarke, Jennifer Faulkner, and Renee Jaggie. Validation of crowd simulations. In *Summer Computer Simulation Conference (08SCSC)*, pages 363–370, Edinburgh, Scotland, June 2008.

[4] Alon Lerner, Yiorgos Chrysanthou, Ariel Shamir, and Daniel Cohen-Or. Data Driven Evaluation of Crowds. In *Motion in Games (MIG2009)*, volume 5884, pages 75–83, Zeist, the Netherlands, 2009. Springer-Verlag LNCS.

[5] D. J. Kaup, T.L. Clarke, Rex Oleson, and Linda C. Malone. Crowd dynamics simulation research. In *16th Conference on Behavior Representation in Modeling and Simulation (BRIMS)*, pages 173–180, 2007.

[6] Cathy Ennis, Christopher Peters, and Carol O'Sullivan. Perceptual evaluation of position and orientation context rules for pedestrian formations. In *SIGGRAPH Symposium on Applied Perception in Graphics and Visualization (APGV'08)*, pages 75–82, 2008.

[7] Rachel McDonnell, Micheal Larkin, Simon Dobbyn, Steven Collins, and Carol O'Sullivan. Clone attack! perception of crowd variety. *ACM Transactions on Graphics (SIGGRAPH 2008)*, 27(3):26:1 – 26:8, 2008.

[8] Martin Pražák, Rachel McDonnell, Ladislav Kavan, and Carol O'Sullivan. A perception based metric for comparing human motion. In *Eurographics Ireland Workshop*, pages 75–80, 2009.

[9] James H. Clark. Hierarchical Geometric Models for Visible Surface Algorithms. *Communications of the ACM*, 19(10):547–554, October 1976.

[10] Lance Williams. Pyramidal Parametrics. *SIGGRAPH Comput. Graph.*, 17(3):1–11, July 1983.

[11] Sébastien Paris, Anton Gerdelan, and Carol O'Sullivan. CA-LOD: Collision Avoidance Level of Detail for Scalable, Controllable Crowds. In *Motion in Games (MIG2009)*, volume 5884, pages 13–28, Zeist, the Netherlands, 2009. Springer-Verlag LNCS.

[12] Paul Kanyuk. Brain Springs: Fast Physics for Large Crowds in WALL-E. *IEEE Computer Graphics and Applications*, 29:19–25, 2009.

[13] David Ryu and Paul Kanyuk. Rivers of rodents: An animation-centric crowds pipeline for ratatouille. Technical report, Pixar Technical Memo 07-02, May 2007.

[14] Daniel Thalmann, Christophe Hery, Seth Lippman, Hiromi Ono, Stephen Regelous, and Douglas Sutton. Crowd and Group Animation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes*, page 34, New York, NY, USA, 2004. ACM.

[15] Stéphane Donikian. A Comparative Review of Reactive Behaviour Models as Proposed in Computer Graphics and Cognitive Sciences. In *Motion in Games (MIG2009)*, volume 5884, pages 63–74, Zeist, the Netherlands, 2009. Springer-Verlag LNCS.

[16] Craig W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioural Model. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, New York City, New York, 1987. ACM.

[17] W. T. Reeves. Particle Systems—a Technique for Modeling a Class of Fuzzy Objects. *ACM Trans. Graph.*, 2(2):91–108, 1983.

[18] M. E. Bratman. *Intentions, Plans, and Practical Reason, Agents*. Harvard University Press, Cambridge, Massachusetts, 1987.

[19] Ioannis Karamouzas, Peter Heil, Pascal van Beek, and Mark H. Overmars. A Predictive Collision Avoidance Model for Pedestrian Simulation. In *Motion in Games (MIG2009)*, volume 5884, pages 40–52, Zeist, the Netherlands, 2009. Springer-Verlag LNCS.

[20] Chris J. Scogings and Ken A. Hawick. Altruism Amongst Spatial Predator-Prey Animats. In *Artificial Life XI*, Winchester, United Kingdom, August 2008.

[21] Ken A. Hawick, Chris J. Scogings, and Heath A. James. Defensive Spiral Emergence in a Predator-Prey Model. *Complexity International*, 12:37, 2008.

[22] Iztok Lebar-Bajec, Nikolaj Zimic, and Miha Mraz. Simulating Flocks on the Wing: The Fuzzy Approach. *Journal of Theoretical Biology*, 233(2):199–220, 2005.

[23] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51(5):4282–4286, May 1995.

[24] S. Krauß, P. Wagner, and C. Gawron. Metastable states in a microscopic model of traffic flow. *Physical Review E*, 55(5):5597 – 5602, May 1997.

[25] R. Wiedemann. Simulation des Straßenverkehrsflusses. Technical report, Schriftenreihe des Instituts für Verkehrswesen, Heft 8, Universität (TH) Karlsruhe, Karlsruhe, Deutschland, 1974.

[26] P. G. Gipps. A behavioural car following model for computer simulation. *Transportation Research Part B: Methodological*, 15(2):105111, 1981.

[27] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *IEEE Conference on Robotics and Automation*, pages 1398–1404, Sacramento, California, April 1991.

[28] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, 1986.

[29] Craig W. Reynolds. Steering behaviors for autonomous characters. In *Game Developers Conference*, 1999.

[30] Rahul Narain, Abhinav Golas, Sean Curtis, and Ming C. Lin. Aggregate dynamics for dense crowd simulation. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, volume 28, page 122:1122:8, 2009.

[31] Craig W. Reynolds. Evolution of Corridor Following Behavior in a Noisy World. In D. Cliff, P. Husbands, J-A Meyer, and S. Wilson, editors, *Third International Conference on Simulation of Adaptive Behavior (SAB94)*, pages 402–410, 1994.

[32] Craig W. Reynolds. Competition, Coevolution and the Game of Tag. In R. Brooks and P. Maes, editors, *Artificial Life IV*, pages 59–69, 1994.

[33] Kenneth O. Stanley, Bobby D. Bryant, and Risto Miikkulainen. Evolving Neural Network Agents in the NERO Video Game. In *IEEE 2005 Symposium on Computational Intelligence and Games (CIG'05)*, Piscataway, New Jersey, 2005. Best Paper Award.

[34] Kenneth O. Stanley, Ryan Cornelius, Risto Miikkulainen, Thomas D'Silva, and Aliza Gold. Real-Time Learning in the NERO Video Game. In *Artificial Intelligence and Interactive Digital Entertainment Conference Demonstration Program (AIIDE 2005)*, 2005.

[35] Risto Miikkulainen, Bobby D. Bryant, Ryan Cornelius, Igor V. Karpov, Kenneth O. Stanley, and Chern Han Yong. *Computational Intelligence: Principles and Practice*, chapter Computational Intelligence in Games, pages 155–191. IEEE Computational Intelligence Society, 2006.

[36] Risto Miikkulainen. Creating Intelligent Agents in Games. *The Bridge*, 36(4):5–13, 2006.

[37] Kenneth O. Stanley, Bobby D. Bryant, and Risto Miikkulainen. Real-Time Neuroevolution in the NERO Video Game. *IEEE Transactions on Evolutionary Computation*, 9(6):653–668, December 2005.

[38] Kenneth O. Stanley, Bobby D. Bryant, and Risto Miikkulainen. Real-Time Evolution of Neural Networks in the NERO Video Game. In *Twenty-First National Conference on Artificial Intelligence (AAAI06)*, pages 1671–1674, Boston, Massachusetts, 2006.