Behavior Capture: Building Believable and Effective AI Agents for Video Games

Maxim Mozgovoy, University of Aizu, Japan
Iskander Umarov, TruSoft Int'l Inc., USA

Abstract: Rapid development of complex video games introduces new challenges for the creation of AI agents. Two important problems in this area are believability and effectiveness of agents' behavior, i.e. human-likeness of the characters and their high ability to achieve own goals. In this paper, we shortly examine the concepts of believability and effectiveness, and show how the combination of learning by observation and case-based reasoning ("behavior capture") can be used to create a believable and effective AI, suitable for practical systems.

Keywords: AI for games, learning by observation, believability, behavior capture.

Introduction

Most modern video games are "inhabited" both with human-controlled and non-player characters (NPCs) — AI-controlled virtual agents, serving as neutral characters, hostile opponents or friendly teammates. Since the quality of NPCs significantly affects the gameplay, the requirements for the corresponding AI system are especially high. However, the idea of high-quality AI as of strong, effective AI does not always work in computer games. The primary factors, contributing to the user satisfaction in most computer games are fun and realism; so an NPC is usually expected to behave realistically and be fun to interact with. The effectiveness of AI in terms of its capability to reach own goals may be not the best criterion of quality in this scenario.

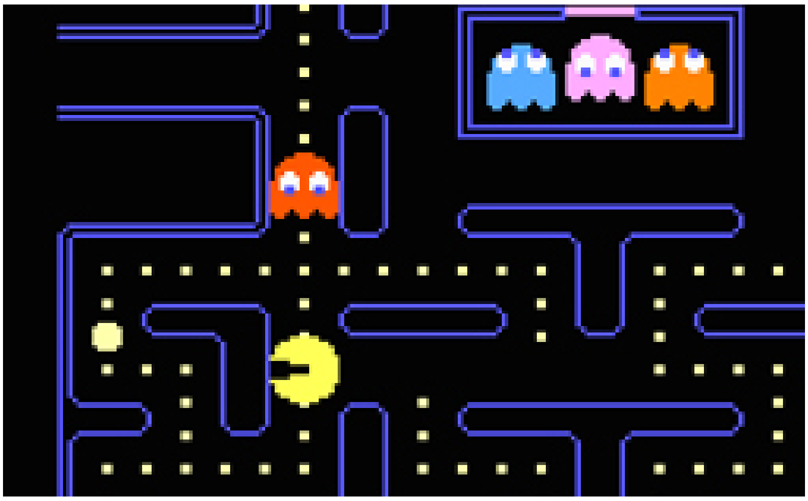

Figure 1. Pac-Man game screen fragment

This idea can be illustrated with the classic Pac-Man game, where computer-controlled ghosts hinder the player from collecting all the pills in the maze (see Figure 1). It is quite easy to develop “optimal ghosts”, being able to catch the player quickly, but such an AI system would destroy the gameplay, since the player would have no chances to win.

In this paper, we shortly discuss the topics of *belivability* and *effectiveness* of AI-controlled agents as important features of a successful AI system for a video game. We examine existing experimental methods to create believable and effective agents and introduce our approach, based on the combination of learning by observation with case-based reasoning and reinforcement learning. This method is explained with the example of a 3D boxing video game.

AI Believability and Effectiveness as Game Fun Factors

As seen from the Pac-Man example, the quality of an AI-controlled agent’s decision making (in terms of achieving own goals in the game) is not the main criterion to measure the successfulness of the used AI system. A game should be *enjoyable*, and AI has to contribute to the overall enjoyability of the game, becoming one of the game’s major *fun factors*.

In words of Paul Tozour, Deus Ex 2 AI programmer: “The whole point is to entertain the audience, so no matter what you do, you need to make sure the AI makes the game more fun. If a game’s AI doesn’t make the game a better experience, any notions of ‘intelligence’ are irrelevant” [1].

One of commonly mentioned features of a successful game AI system is *believability*. A belivable agent is defined in [2] as the “one that provides the illusion of life, and thus permits the audience’s suspension of disbelief”, and usually characterized with human-like capabilities, such as to learn, to “think” between the decisions, to make mistakes, and to adjust own strategy in response to opponent’s actions.

Naturally, AI developers often try to make their agents belivable. For example, the official Counter-Strike bot, according to its authors, can “behave in a believably human manner”, and “create behavior variation among bots” in terms of aggression, skill, teamwork, reaction times, and morale [3].

However, from the technical point of view, the approaches used to build virtual players are generally conservative. In particular, the aforementioned Counter-Strike bots rely on preprogrammed rule-based scenarios [3]. The main problem with the manual approach to the design of agents’ behavior patterns lies in its limited scalability. Complex behavior requires complex data structures and algorithms, and the consistency of decision-making rules is difficult to maintain.

While believability of NPCs in game worlds is a relatively new topic, *effectiveness* of decision-making (the capability of a character to reach own goals) is a classical aim for virtual agents’ design. To program AI that can be a strong opponent for a human-controlled player or a smart and supportive teammate is a challenging task for a high variety of computer games. As noted before, believability and effectiveness in many cases can be considered as independent aims: believability does not ensure effectiveness and vice versa. Effectiveness of AI is another fun factor of a computer game: computer-controlled characters should be reasonably skillful and, therefore, fun to play with.

One example of a virtual environment where good AI is completely identified with effective AI is RoboCup [4]. Within this project, researchers and developers create teams of virtual soccer players that try to beat teams, created by others. A winning team should exhibit effective behavior, which does not have to be similar to strategies of human players in a real soccer match.

Experimental Approaches to Create Belivable Behavior

Most research projects follow a natural way of constructing human-like believable behavior by analyzing actual human behavior patterns and subsequently implementing them in AI. Even hand-coded algorithms can contain specific behavioral patterns, considered as “human-like” by their creators [3]. However, a greater interest is evoked by the methods that can automatically construct agents’ knowledge by observing behavior of human players. Let us shortly introduce several projects of this kind.

Thurau et al. [5] examine the possibility to apply a Bayesian network trained with sample human-generated actions in Quake 2 environment. This approach was successful in the following relatively simple experiments on motion planning: (a) reach the same target map position as human player; (b) reach the same target map position as human player, while visiting a certain other map position on the route. In the later experiments this approach was shown to produce believable behavior [6].

A similar Bayesian model was implemented by Le Hy et al. [7] to create a believable agent for Unreal Tournament. The decision-making system is represented as follows. A bot’s internal state is encoded with a vector of parameters. Each possible state has a number of outgoing transitions to the future probable states. Every outgoing transition (representing a bot’s action) has an associated probability. Altogether, these probabilities form behavioral patterns of the bot. Authors study both manual probability distribution specification (to obtain clearly shaped behavior — e.g., aggressive or cautious) and distributions obtained through learning by observation. The experiments prove the approach to be suitable for creating believable bots with good level of skill: a character, trained by aggressive human player, demonstrates aggressive behavior and beats and average-leveled built-in AI agent.

Choi et al. [8] apply a general-purpose cognitive architecture ICARUS to train an agent for Urban Combat — a modification of Quake 3 Arena game. Initially the agent is given hand-coded basic knowledge about the objectives and acting capabilities. Then the automated learning algorithm is executed every time whenever the agent has achieved a certain goal. This algorithm generates new behavioral skills that help to achieve the same goal with less effort, therefore, improving agent’s performance. The experiments were conducted as a series of capture-the-flag games, where an agent had to find and pickup a flag, blocked by various obstacles. The authors make a conclusion that the agent exhibits reasonable human-like behavior by exploring the area, learning how to avoid obstacles, and utilizing obtained knowledge in other environments.

Proving Belivability

An essential phase of every research project, devoted to the construction of believable AI, is *proving* belivability. If a certain system is claimed to exhibit belivability, there should be a method to support this claim objectively.

However, evaluating believability is not an easy task. One of the possible approaches to the problem is to adapt a well-known Turing test [9] to intelligent agents, behaving in a game world. Glende [10] lists the following features that the agents should possess in order to pass Turing test: (a) a reasonable combination of predictability and unpredictability in agent's decisions; (b) creativity in problem-solving; (c) clear personality; (d) internal intensions and autonomy in acting; (e) the capability to plan and improvise; (f) the ability to learn.

Actual experiments with Turing tests on AI agents were conducted by Livingstone [11] and Gorman et al. [6]. As one might expect, Turing test can be relatively easily passed by a conventional AI agent in case of simple games, due to the overall simplicity of gaming process. As the game world becomes more complex, user expectations of AI quality increase.

Typical experiments are conducted as follows. First, several video clips, showing actual gaming process, are recorded. Then a number of people are asked to identify human-controlled and AI-controlled characters. If the observers fail to identify AI reliably, the system is considered believable.

An alternative semi-automatic method to evaluate believability is described by Tencé and Buche [12]. An experimenter has to select a vector of numerical attributes, describing the current state of each game character. A recording module saves a sequence of snapshots of these vectors during the game process. Then the distance between the vectors that correspond to different agents can be calculated with conventional methods (e.g., dot product). As a result, the agents are grouped according to behavioral similarity. Believable AI agents should be close to human-controller characters.

Behavior Optimization

The projects listed in the last sections are primarily concentrated on believability of AI agents. A number of other experimental projects try to optimize agents' behavior, making them more effective in achieving their own goals. Usually such experiments start with an agent having a hardcoded decision-making system that reflects the authors' abilities to design a winning strategy. Next, the agent's behavior is adjusted with the help of a certain automatic optimization method. For example, having a selection of applicable behavior patterns, it is possible to encourage the use of winning patterns, meanwhile discouraging inferior action sequences (usually through reinforcement learning methods [13]). Since this process optimizes AI effectiveness rather than believability, the result is evaluated in terms of achieving game goals rather than in being "human-like".

Bonse et al. [14] extended the standard Quake 3 bot with a neural network that maps currently observed game state parameters into the vector of weights of applicable agent's actions, and applied Q-learning algorithm to adjust these weights. The network was trained on a large number (100 000 – 200 000) of one-on-one games (the game continues until one of the opponents is killed). The results showed that the improved bot has a 65% chance to win in a match with the original Quake 3 AI agent.

The project by Bonacina et al. [15] continues this research by learning better dodging (i.e. hit-avoiding) behavior. In their experiments, a neural network-enabled Quake 3 bot has to survive as long as possible in an empty room with an enemy computer-controlled bot equipped with a rocket launcher. Experiments show that the bot quickly learns dodging, which allows it to survive for more than 150 game tics, while the original bot has an average lifetime of 100 tics.

Cole et al. [16] successfully applied genetic programming to the task of optimizing Counter-Strike built-in bot's behavior. A bot's playing style is defined with a set of parameters, such as adjusting weapon selection preferences and agressivity. Different parameter configurations result in different AI effectiveness, and the optimal configuration is not easy to find. The idea of the experiment was to find optimal or sub-optimal configuration with the help of genetic algorithms. The authors have demonstrated that their approach can configure bots as effectively as human experts. Moreover, genetic optimization procedure found several different winning sets of parameters, and therefore produced a number of highly effective bots with distinct playing styles.

Behavior Capture Approach

The authors of this paper used a combination of learning by observation with case-based reasoning to create a believable AI agent for a 3D boxing video game [17]. This agent was then optimized with reinforcement learning in order to obtain higher behavior effectiveness [18]. We selected these methods in order to satisfy the following AI design requirements:

- complex, non-repetitive behavior of AI agents;
- distinct personalities of AI boxers, exhibiting a variety of skill levels and playing styles;
- the capability to design, edit and adjust AI's behavior (for a game designer);
- "train your own boxer" mode as a user-end feature.

Learning and Acting

Our system operates in two basic modes: learning mode and acting mode. In learning mode, the system watches the actions of a human player and creates a game's acting graph — a variation of finite state machine that stores game states (snapshots of the game world) and transitions between them, caused by human's actions (see Figure 2). In acting mode, a variation of case-based reasoning algorithm is used to find possible actions that correspond to the current situation in the game. Some heuristic techniques are utilized to increase chances of finding appropriate game states in the acting graph (if ideal match is not found), and to preserve action sequences, demonstrated by a trainer. When a set of applicable actions are found, a weighted random choice is used to select one of them. More frequent actions have more chances of being chosen.

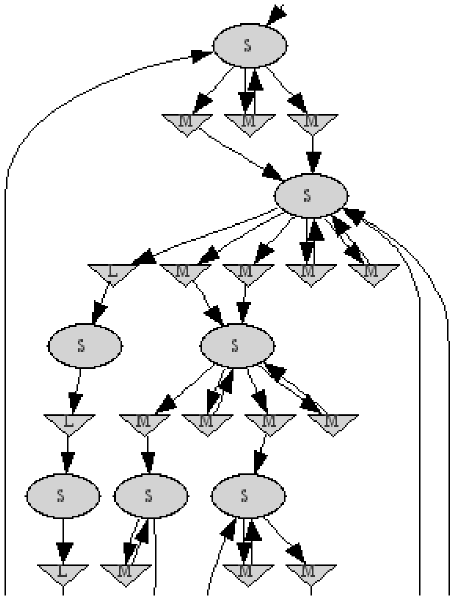

Figure 2. 3D boxing game's acting graph (a fragment)

In contrast to many other types of knowledge representation structures (such as neural networks), acting graphs are easy to edit manually. AI programmers can manually combine acting graphs, delete unwanted parts, or adjust action weights. It is also relatively easy to debug such a system, since all AI decisions are traceable to concrete graph vertices and edges. One can note the lack of long-term planning instruments in this mechanism, but strategic decision making is hardly needed in our arcade-style boxing game.

The resulting AI agent has no built-in capability of distinguishing strong and weak actions; it repeats human-demonstrated behavioral patterns, even if they lead to inevitable defeat. Furthermore, as our experiments show, the AI agent, trained by a certain human player, is less skillful than its trainer. It happens due to imperfectness of the case-based reasoning system: sometimes it fails to find the most appropriate actions that should be chosen according to the training session.

These problems can be addressed with manual adjustments of the acting graph and with an automated optimization procedure. Manual adjustments are handy for removal of unintentional weak actions, mistakenly performed by a trainer. Automatic optimization technique, based on reinforcement learning, encourages the use of effective action sequences, leading to a victory in the game. The algorithm analyzes the outcome of the just applied action (we simply evaluate whether our boxer got more or less damage than its opponent), and modifies its weight accordingly. Successful actions get higher weights and higher probabilities of being chosen. A backpropagation routine distributes the calculated action reward among preceding actions.

Testing Believability and Effectiveness

The effectiveness (skill level) of an AI agent is easy to evaluate in a series of games between the agent and a human player or between the agent and another AI system. We have watched a sequence of matches between our self-learning agent and a strong rule-based AI engine. First five-six matches usually end with our agent's defeat, and at its sixth-seventh game our agent becomes able to beat its opponend. The subsequent games end with the agent's victory.

Testing believability is a harder challenge. In our case this task can be divided into the following subtasks: (a) to make sure that our agents are truly believable, i.e. perceived as human players by the observers; (b) to make sure that our agents remain believable after modifications, introduced by reinforcement learning.

Believability testing was accomplished by employing Turing test. We asked several people to watch video clips of games between different opponents, including human-controlled boxers, our AI agents, and boxing engine's built-in handcrafted AI players. As reported in [17], the observers hardly distinguish our AI agents from human players. At the same time, handcrafted AI system is easily identifiable.

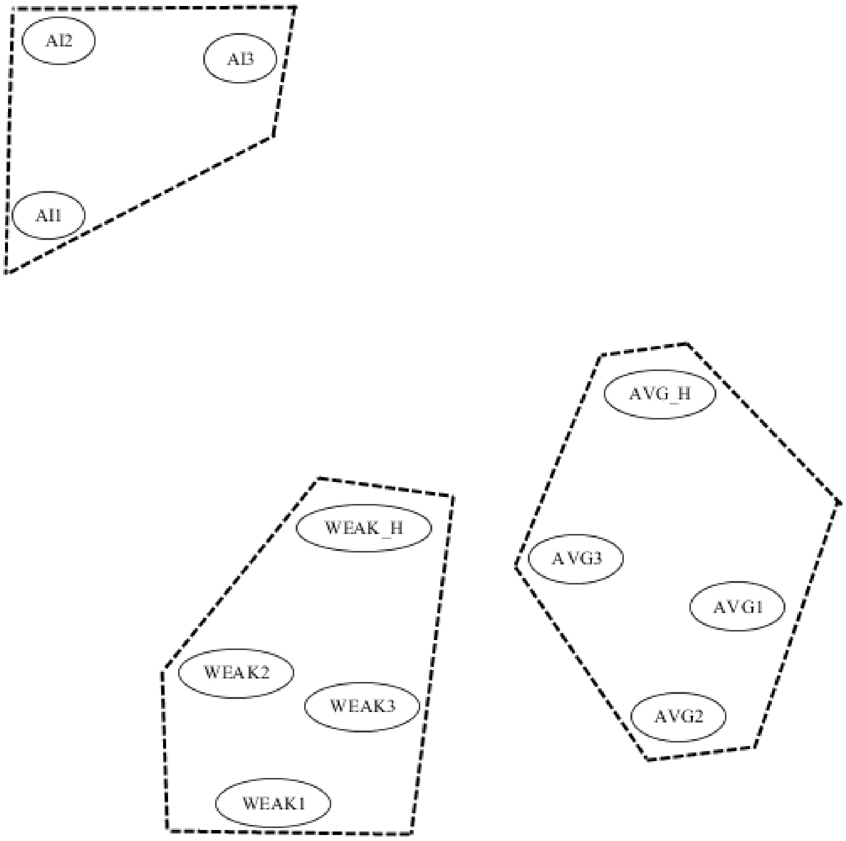

Figure 3. Behavioral similarity of different boxers

We were also interested to try an automated believability evaluation method, similar to the one proposed in [12]. For each boxer, we have computed a simple "behavioral fingerprint", represented with a vector of probabilities of each possible combination of two successive actions. A dot product of a pair of such vectors was used as a degree of behavioral similarity of the corresponding boxers. The results we got agree with manual evaluation. A human-trained boxer is closer to its teacher (in terms of behavior similarity) than to other boxers in the experiment.

An automatically created plot, demonstrating the behavioral similarities between the boxers, is shown in Figure 3 (the dashed lines are drawn manually). Three built-in handcrafted AI players (AI1-AI3) form a clearly isolated cluster. Other players include two human experts WEAK_H and AVG_H, and their corresponding behavior-capture AI agents (trained on distinct subsets of WEAK_H and AVG_H recordings) WEAK1-WEAK3 and AVG1-AVG3. As it can be seen, behavior-capture AI agents are close to their human trainers and form easily identifiable clusters.

Naturally, we were curious to find out, how does reinforcement learning change the behavioral portrait of an AI agent. In order to do this, we have built a plot of similarities between the human-controlled player (*hum*), behavior-capture AI agents, trained on different subsets of *hum* recordings (*bc1*-*bc3*), and their optimized versions *rl1*-*rl3*. As shown in Figure 4, reinforcement learning increases the distance between the agent and its teacher, but they still remain in the same cluster and preserve believability.

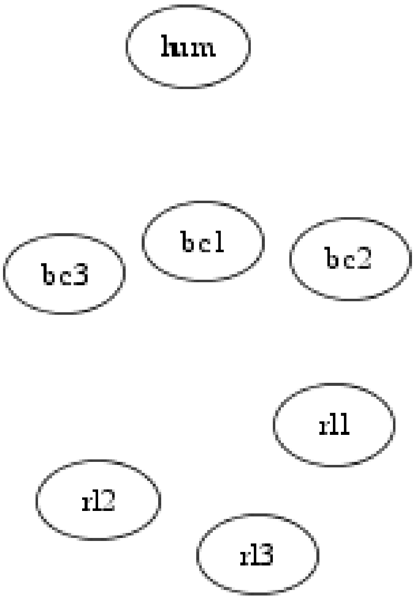

Figure 4. The effect of reinforcement learning on similarity measure between the agents

Conclusion

One of the primary challenges for AI methods in designing virtual game worlds lies in creation of believable and effective agents that should inhabit them. After a certain degree of complexity, the flaws of hand-crafted agents become evident, reducing the overall quality of the game world. Modern game projects require advanced approaches that can provide more realistic and effective behavior of NPCs.

Two important features of a high-quality modern AI system are belivability and effectiveness of its decision making. This claim is supported both by academic researchers and game developers. However, current commercial games rarely implement AI algorithms that go beyond traditional finite state machines and rule-based systems.

A number of current research projects demonstrate that believable agents should possess certain features, hardly achievable without methods that rely on observing and simulating actual human behavior. Effectiveness (in terms of agent's skill level) is also harder to achieve with traditional methods in complex environments, where a winning strategy might require non-trivial goal planning and real-time adjustment to human actions. Internally these human-observing methods can be based on different approaches, such as case-based reasoning systems, decision-making trees, neural networks or rule-based cognitive architectures. We suggest that this approach will become the most popular way to design believable agents in the near future.

Learning by observation can be combined with automatic winning strategy-selecting algorithms to provide both believable and effective behavior, as we demonstrated in our projects [17] and [18]. The game of 3D boxing is relatively simple and does not reveal numerous challenges for our approach, related, for example, to team behavior and goal planning. These topics will be researched in further experiments.

References

[1] D. Diller, W. Ferguson, A. Leung, B. Benyo and D. Foley, "Behavior Modeling in Commercial Games," *Proc. of the 13th Conference on Behavior Representation in Modeling and Simulation*, 2004.

[2] J. Bates, "The role of emotion in believable characters," *Communications of the ACM*, 1994, vol. 37, pp. 122-125.

[3] M. Booth, "The Official Counter-Strike Bot," 2004.

[4] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawai and H. Matsubara, "RoboCup: A Challenge Problem for AI and Robotics," *Lecture Notes in Computer Science*, 1998, vol. 1395, pp. 1-19.

[5] C. Thurau, T. Paczian and C. Bauckhage, "Is bayesian imitation learning the route to believable gamebots," *Proc. of GAME-ON North America*, 2005, pp. 3-9.

[6] B. Gorman, C. Thurau, C. Bauckhage and M. Humphrys, "Believability testing and Bayesian imitation in interactive computer games," *Lecture Notes in Computer Science*, 2006, vol. 4095, pp. 655-666.

[7] R. Le Hy, A. Arrigoni, P. Bessiére and O. Lebeltel, "Teaching bayesian behaviours to video game characters," *Robotics and Autonomous Systems*, 2004, vol. 47, pp. 177-185.

[8] D. Choi, T. Konik, N. Nejati, C. Park and P. Langley, "A believable agent for first-person shooter games," *Proc. of the 3rd Annual AI and Interactive Digital Entertainment Conference*, 2007, pp. 71-73.

[9] A. Turing, "Computing machinery and intelligence," *Mind*, 1950, vol. 59, pp. 433-460.

[10] A. Glende, "Agent design to pass computer games," *Proc. of the 42nd annual ACM Southeast regional conference*, pp. 414-415.

[11] D. Livingstone, "Turing's test and believable AI in games," *Computers in Entertainment*, 2006, vol. 4.

[12] F. Tencé and C. Buche, "Automatable evaluation method oriented toward behaviour believability for video games," *International Conference on Intelligent Games ans Simulation*, 2008, pp. 39-43.

[13] L. Kaelbling, M. Littman and A. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, 1996, vol. 4, pp. 237-285.

[14] R. Bonse, W. Kockelkorn, R. Smelik, P. Veelders and W. Moerman, "Learning Agents in Quake III," *Technical Report*, University of Utrecht. 2004.

[15] S. Bonacina, P. Lanzi and D. Loiacono, "Evolving Dodging Behavior for OpenArena using Neuroevolution of Augmenting Topologies," *PPSN'08 Workshop (Computational Intelligence and Games)*, 2008.

[16] N. Cole, S. Louis and C. Miles, "Using a genetic algorithm to tune first-person shooter bots," *Proc. of the International Congress on Evolutionary Computation*, pp. 139-145.

[17] M. Mozgovoy and I. Umarov, "Building a Believable Agent for a 3D Boxing Simulation Game," *Proc. of the 2nd International Conference on Computer Research and Development*, 2010, pp. 46-50.

[18] M. Mozgovoy and I. Umarov, "Building a Believable and Effective Agent for a 3D Boxing Simulation Game," *Proc. of the 3rd IEEE International Conference on Computer Science and Information Technology*, 2010, vol. 3, pp. 14-18.