# Believable Judge Bot That Learns to Select Tactics and Judge Opponents

Ruck Thawonmas *Senior Member, IEEE*, Seiji Murakami and Takumi Sato

*Abstract*— This paper describes our believable judge bot ICE-CIG2011 that has an ability to learn tactics from a judge player and an ability to judge an opponent character as a human or a bot. We conjecture that a bot with these two abilities should be considered human-like in a competition environment, such as BotPrize, where human players participate to compete not only for being the most human-like player but also the best judge. Main contributions of this work lie in our mechanisms for achieving these two abilities.

To achieve the former ability, we develop a system and GUI that allow a selected judge player – whose role is to train ICE-CIG2011 – to control his or her character by only deciding which tactic to use under a given situation. We then obtain the judge's tactic log and use it for training tactic selection of ICE-CIG2011 with neuro evolution of augmenting topologies. To achieve the latter ability, we acquire additional logs when the judge character interacts with other opponent characters. In order to represent the play of a known (bot or human) character, we train a neural gas – a kind of self-organizing neural network – from its log. For an unknown character, once its neural gas is trained after a certain period of observation, ICE-CIG2011 decides if it is a human or bot by using the $K$-nearest-neighbor algorithm; this algorithm considers the majority in the labels of the $K$-nearest neural gases, of known characters, to the neural gas of that unknown character. Experimental results are given and discussed concerning these two abilities of ICE-CIG2011.

## I. INTRODUCTION

Since the first BotPrize[1] in 2008, our laboratory has participated in this AI competition, the aim of which is to compete for implementation of believable first-person-shooter bots in an FPS game called Unreal Tournament 2004. Our main design policy was to develop a bot that looks like a moderate-level human player. This policy worked well in that our bot always ranked high at each competition. However, we have never won yet!

After a related competition recently held at the Human-like Bot competition at CEC 2011[2], a couple of human judges mentioned that some bots were easy to be detected because they kept moving during each round while those judges considered characters who occasionally stopped as humans. These comments surprised us because, according to our FPS experiences, most non-novice players will never stop moving in order to prevent themselves from being shot by their opponents. We conjecture that such comments stemmed from a situation that every human player has two main roles in this kind of competition, one as a judge and the other as a player. As a result, they compete with each other to not only be the best judge but also the best human-like player. This might cause them to stop their movements, observe other characters, and make a judgment if those characters are humans or bots.

Based on this conjecture, we present in this paper two mechanisms used in ICE-CIG2011, our bot for BotPrize 2011[3] at CIG 2011, developed on the top of the Pogamut 3 platform [1] for Java-bot coding. The first mechanism allows ICE-CIG2011 to learn tactic selection from a certain judge player, the third author in this work. The second mechanism allows ICE-CIG2011 to make a judgment on an unknown character based on the similarity between the unknown player and known characters previously seen. The contributions of this paper are as follows:

1) It is the first work, to our knowledge, on believable FPS bots that focuses on mimicking judge behaviors, rather than general FPS player behaviors, which should improve the humanness of ICE-CIG2011 in the competition.
2) We present a mechanism in Sec. III – for learning tactic selection – that consists of (i) a system and GUI for allowing the trainer player to control his or her character by only deciding which tactic to use under a given situation as well as (ii) a module for learning tactic selection from a resulting log using neuro evolution of augmenting topologies (NEAT) [2].
3) We present another mechanism in Sec. IV – for judging an unknown character – that consists of (i) acquisition of the play logs of opponent characters, (ii) representation of each of those play logs with a neural gas [3], and (iii) assignment of the label (bot or human) to an unknown opponent character by using the $K$-nearest-neighbor algorithm, which considers the majority of the labels of the $K$-nearest neural gases of known opponent characters to the neural gas of that unknown opponent character.
4) We describe the ICE-CIG2011's states and rules in Sec. V.
5) We report our experiment and give discussions on it in Sec. VI, followed by conclusions and future work in Sec. VII.

The authors are with the Intelligent Computer Entertainment Laboratory, Graduate School of Science and Engineering, Ritsumeikan University, Shiga, Japan (phone: +81-77-561-5048; fax: +81-77-561-5203; email: ruck@ci.ritsumei.ac.jp).
[1] http://botprize.org/
[2] http://botprize.org/HumanlikeBots.html

[3] At the time when we are preparing the camera-ready version, the competition's name has not been fixed yet. In this paper, we customarily call it BotPrize 2011.
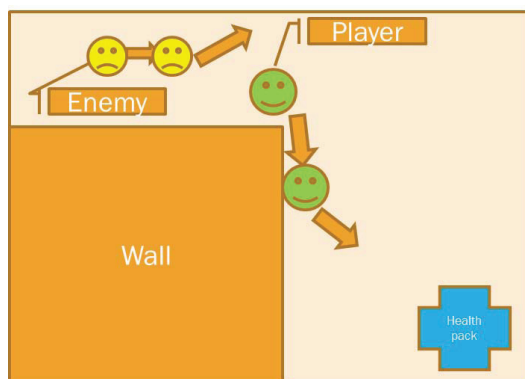
Fig. 1. Example of a movement that the player's intention is hard to derived from the game log

## II. Related Work

There exist a number of work on strategy decision in FPS games in recent years. For example, Hoang et al. [4] developed a team of bots that use Hierarchical Task Network planning for deciding the bot strategy in a team-battle. McPartland and Gallagher [5] used the Sarsa algorithm for the purpose of reinforcement learning on item-collection and battle-strategy. Soni and Hingston [6] adopted neural networks for their bot in order to improve selection of battle strategies by mimicking human strategies.

Existing research on humanness in terms of movements and actions in the FPS includes the work of Gorman and Humphrys [7], [8]. Therein, a bot was proposed that imitates human behavior in Quake II, another FPS game. In the former paper, the bot imitates a path used by a human player to collect items. Based on the human-player play log, it decides the next item to collect by using fuzzy clustering and the path by using reinforcement learning. In the latter paper, the bot imitates human player's behavior during the battle. It decides which weapon to use, position to aim, and timing to fire by using three interconnected neural networks. However, both do not indicate the bot's humanness in a real match such as the BotPrize competition.

An outline of BotPrize 2008 and that of each participating team were given in Hingston [9]. The research in Wang et al. [10], by one of those teams 2008, is worth mentioning. Their bot learns weapon selection and battle strategy with unsupervised learning Fusion Architecture for Learning, COgnition, and Navigation. However, there is not much learning about humanness because a reward is given when the bot hits or kills his opponent. More recently, Conscious-Robots [11], the winner bot of the BotPrize 2010 competition and the Human-like Bot competition at CEC 2011, was developed using a multi-layer cognitive architecture based on team members' research findings on machine consciousness. To our knowledge, however, there exists no work that considers improvement of BotPrize bots' humanness through mimicking a judge player as done in our present work.

## III. Tactic Selection

Learning of tactic selection is a challenging issue. Figure 1 shows an example where it is hard to know the player's real intention from his or her movement and/or action log. This is because there might be three intentions behind those kinds of movements, i.e., to obtain the health pack item, escape from the opponent, or hide behind the wall. Figure 2 shows the overview of our mechanism for coping with this issue by acquiring a tactic log directly from a selected judge player and learning his or her tactic selection from this log.

### A. Acquisition of Tactic Log

A player character (PC), a character that the judge player plays, in our log acquisition system has two states: item-collection and battle-and-judge (Fig. 3). In the former state, the PC autonomously conducts its actions, as with our previous work for learning fighting tactics in ICE-CEC2011 [12], according to pre-defined rules described in Sec. V. In the latter state, however, its tactic is determined by the judge player; the rest is done automatically. In this work, we focus on the following four tactics; where the targeted opponent (TO) mentioned below is the nearest opponent character among all characters in sight and will change to another character, say, B, if the distance to B is less than two thirds of the distance to the current one:

- **retreat** (RT) by which the PC retreats from the TO in a zigzag movement,
- **approach** (AH) by which the PC moves forward to the TO in a zigzag movement and moves around it when their distance is below a given threshold,
- **recover** (RR) by which the PC acquires curative items in the nearest order,
- **observe** (OE) by which the PC stops moving to generate room for the judge player to observe and judge other characters.

We implemented these tactics such that the PC always gazes at the TO. This system allows the judge player to select one of these four tactics by clicking a panel in the corresponding GUI as shown in Fig. 4 or a reserved key in the keyboard (Q, W, A, S for RT, AH, RR, OE, respectively). We note here that OE is newly introduced in this work replacing the hide-and-shoot tactic in ICE-CEC2011. OE plays an important role in making ICE-CIG2011 occasionally stop and, thereby, look like a human judge player.

### B. Training of Tactic Selection

Data in the log obtained by the log acquisition system are processed to generate a training data set for NEAT. We use NEAT for this task because it has an ability to evolve the neural-network structure accordingly. Each data in this data set consists of a tactic label and 16 normalized input features shown in Table I.

We train NEAT in a hierarchical manner. First, we train NEAT so that it can distinguish between two groups of tactics: active judging (AH and OE), and passive judging (RT and RR). Then, we additionally introduce two networks
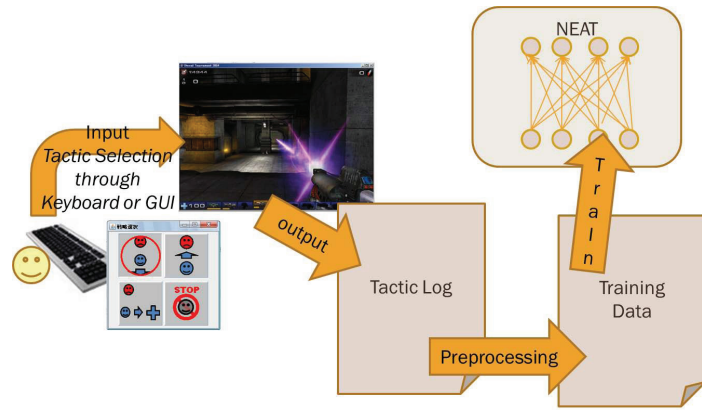
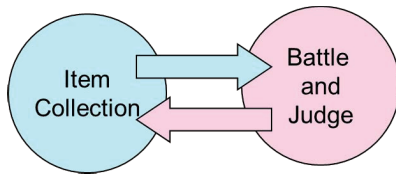Fig. 2. Overview of our system for acquisition of tactic logs
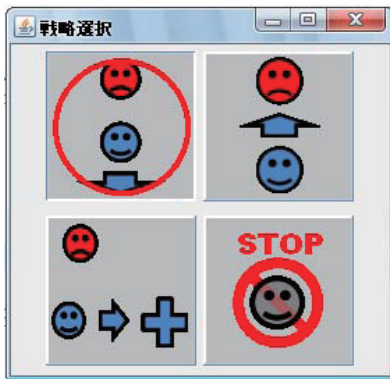


Fig. 3. ICE-CIG2011's finite state automaton



Fig. 4. GUI for tactic selection (RT, AH, RR, OE)

1) endurance level (Health Level+ Armor)
2) distance to TO
3) corridor width
4) recent transition of Health Level
5) distance to the nearest curative item
6) optimal shooting distance of the weapon in use
7) destructive power of the weapon in use
8) optimal shooting distance of the TO's weapon in use
9) destructive power of the TO's weapon in use
10) relative direction towards TO
11) number of nearby path nodes
12) number of opponent characters in sight
13) accumulated TO observation time
14) whether or not previously shot TO with the Link Gun
15) whether or not TO is moving
16) five previous tactics in use

of NEAT and train the first one for distinguishing between AH and OE in the former group and the second one for distinguishing between RT and RR in the latter group. After the training phase, ICE-CIG2011 selects its tactic according to the NEAT output.

## IV. Judge Decision

An ability to correctly judge is another important issue for ICE-CIG2011 to have a high degree of humanness in BotPrize. Our idea for achieving this ability is that of using a neural gas (NG), for representing each TO, whose neurons are organized (trained) according to the TO's play log. In particular, we use a batch version of the NG which allows fast training. We then compare the NG of an unknown TO with the NG of each character whose label (human or bot) is known and determine the label of the unknown TO according to majority voting of the labels of the $K$-nearest known characters. This mechanism is incorporated as a component in the weapon-selection step in Sec. V-B.

### A. Acquisition of Opponent Log

As for the type of play log for this task, we focus on how the TO makes its decision to approach to or retreat from our character. We consider that the TO makes such a decision based on the distance between the TO and our character as well as on the TO's current weapon in use. Thereby, we define each vector data $\mathbf{x}(t)$ at time $t$ in the play log such that it consists of three features:

- $x_1(t)$ the distance between our character (the PC in the log acquisition system or ICE-CIG2011) and the TO at time $t - 1$,
- $x_2(t)$ the weapon that the TO uses at time $t - 1$, and
- $x_3(t)$ the distance that the TO moves during time $t - 1$ and $t$, where the value is plus if the TO approaches to our character; otherwise minus.

We incorporate into our character a module for acquiring the play log of each TO. This module accumulates the play log of each TO that our character encounters in a given game. We define known TOs as the TOs whose labels are known. Examples of them include human characters played by students in our laboratory and bot characters such as our previous BotPrize bots.

### B. Training of Neural Gas

A NG, with the number of $n$ neurons, is assigned and trained for each TO. Each neuron has a weight of three dimensions, corresponding to the three features discussed in Sec. IV-A. We assign every NG the same initial weights. Thereby, the distance between the weights of a pair of trained NGs indicates the difference in decision making of the corresponding characters.

For each NG, let $m_i(t)$ denote the three-dimensional weight vector of the $i$th neuron, where $i = 1, \ldots, n$. We propose a variant of a batch-type NG as follows:

$$m_i(t+1) = m_i(t) + \sum_{1 \le k_i \le l} h_{k_i}(t) \Big( \frac{1}{|K_i|} \sum_{x_j \in K_i} \big( x_j - m_i(t) \big) \Big),$$

where

$$h_{k_i}(t) = \exp\big( \frac{-k_i}{\alpha^{(T-t)/T}} \big),$$

and $K_i$ is the set of data that $m_i(t)$ is the $k_i$th nearest; $l$, $\alpha$, and $T$ are user-defined parameters and set to 15, 5, 100, respectively. The number of neurons $n$ is set to 27.

For an unknown TO, once the size of its play log is above a threshold, called log-size threshold, its NG will be trained and compared with the NG of each known TO using the Mahalanobis distance. We use this distance metric because it takes into account the correlation of the data and is scale-invariant. The label of this unknown TO is then determined by majority voting of the labels of the $K$ nearest NGs of those known TOs.

## V. STATES AND RULES

Following the same recipe in [13], the initial state of our character is the item-collection state. It is changed to the battle-and-judge state when an opponent is found in its sight. The battle-and-judge state is changed to the item-collection state when one of the following three conditions is met:

- Our character defeats the TO.
- Our character loses sight of the TO and is in bad condition, i.e., not having bullets for its weapons, except for the initial assault rifle, or having the health level below 30.
- Our character follows the TO to the last seen location, but cannot find it.

### A. Item-Collection State

In this state, our character runs to collect items (weapons, bullets, health items and armor). This state contains four steps:

- **Reflexive Action:** If our character does not jump when it crashes to a wall, all human judges will easily size it

up as a bot. The same holds for a situation where it is shot from behind. As a result, our character is designed to jump or change its target item in the former case and look back occasionally in the latter one.
- **Determination of the Target-Item Type:** Our character makes a decision on which type of item among weapons and curative items (health and armor items) that it needs. If the resulting item type is the same as the previous one, the next step will be skipped, and our character will stick to its present path.
- **Determination of the Target Item and Path:** If the target-item type is "any weapon", our character will choose its target weapon at random. Otherwise, it determines its target item as the nearest item of the type decided at the previous step. However, in this regard, ineffective items for the current situation are ruled out in advance, i.e., our character does not select the health pack if its life level is already at the upper limit because it cannot further increase the level. Once the target item is decided, the shortest path to it is determined using $A^*$.
- **Movement:** In this step, our character moves following the determined path. To proceed smoothly, we consider our character reaches its goal if within a specific distance from that goal.

### B. Battle-and-Judge State

In this state, our character encounters or chases the TO. This state consists of four steps: tactic selection, weapon selection, aim-point determination, and shooting-and-moving. Tactic selection is determined by the judge player for the PC in the log acquisition system and by NEAT for ICE-CIG2011. The other three steps are performed according to the rules similar to those used in our bots submitted to BotPrize 2008 [13], 2009 and 2010, as well as the Human-like Bot competition at CEC 2011 [12].

The weapon-selection step is divided into two components: one for TOs with the play-log size less than the log-size threshold and the other for TOs with the play-log size above the log-size threshold. The former component, also introduced in our bot submitted to the BotPrize 2010, selects the weapon for battling against the TO according to NEAT specially trained for this task; however, the detail of this NEAT is beyond the scope of this paper. The latter component uses the judge mechanism in Sec. IV for selecting the mode of the Link Gun. Following a recent competition rule, ICE-CIG2011 uses the primary fire mode against the bot TO and the alternate fire mode against the human TO.

## VI. EXPERIMENT AND DISCUSSIONS

We conducted an experiment similar to the competition style at CEC 2011, where a shooter gains 10 points and instantly kills his target if the correct fire mode of the Link Gun is used to judge the target; otherwise, the shooter instantly dies and loses 10 points. This experiment consists of six rounds. Each round, running on one of the six maps used in the CEC 2011 competition, lasts for ten minutes

TABLE II

RANKING OF THE BEST BELIEVABLE CHARACTERS - CHARACTER
(HUMANNESS)

| Ranking | Bots | Humans | Epic Bots |
|---|---|---|---|
| 1 | ICE-CIG2011 (48%) | H3, H4 (75%) | Epic 5 (100%) |
| 2 | ICE-CIG2010 (46%) | | Epic 3 (58%) |
| 3 | ICE-CEC2011 (40%) | H2 (61%) | Epic 4 (33%) |
| 4 | Hunter (38%) | H1 (56%) | Epic 1 (25%) |

TABLE III

RANKING OF THE BEST JUDGE CHARACTERS - CHARACTER (ACCURACY)

| Ranking | Bots | Humans |
|---|---|---|
| 1 | ICE-CIG2011 (56%) | H2 (73%) |
| 2 | ICE-CEC2011 (40%) | H1 (66%) |
| 3 | | H3, H4 (51%) |

where four bot characters, four human players, and a couple of Epic original bots participate, where Epic $x$ indicates an Epic bot of level $x$. Three of the bot characters are ICE-CIG2011, ICE-CEC2011 (2nd place at the CEC 2011 competition), ICE-CIG2010 (3rd place at the BotPrize 2010 competition). The other one is Hunter which is the default bot of Pogamut. The four human players, H1 (moderate-level male), H2 (advanced-level female), H3 (advanced-level male), H4(advanced-level male) are students in our laboratory, but not involved in this work. We asked them to play with the aim to be both the best human-like player and best judge at the same time.

Tables II and III summarize the results. ICE-CIG2011 receives promising humanness of 48%, very close to the bottom line of 50% in order to achieve the BotPrize major prize. Its humanness outperforms all bots, except Epic 3 and Epic 5, but there is still a gap between the humanness of all human players and that of ICE-CIG2011. As for the judging ability, ICE-CIG2011 outperforms not only ICE-CEC2011 but also some human players, i.e., H3 and H4. With continuing improvements until the BotPrize 2011's deadline, we believe that ICE-CIG2011 place itself in a very promising position for the best human-like bot and judge bot.

## VII. CONCLUSIONS AND FUTURE WORK

We presented our approach for implementing our BotPrize 2011 bot (ICE-CIG2011) that behaves like a human judge in terms of both tactic selection and judge decision. We used NEAT and a combination of the neural gas and $K$-nearest-neighbor algorithms for learning tactic selection from a trainer human player and making a judge decision from the similarity of an unknown character to known characters, respectively. Our results reveal the effectiveness of the presented approach in that ICE-CIG2011 has promising humanness compared to the other non-player characters and a judging ability superior even to some human players in a BotPrize-like environment.

Until the deadline of the BotPrize 2011 competition, we plan to increase the performance of ICE-CIG2011 by adding more states, fuzzifying certain input features to NEAT, and adjusting the number of neural-gas neurons. In addition, for tactic selection, we plan to select and use a new judge player with higher humanness than the one in this work (the third author) for training ICE-CIG2011. For judge decision, we continue acquisition of play logs of known characters. After the BotPrize 2011 competition, we plan to directly modify NEAT or the neural gas so as to make them more suitable to their tasks in our approach.

REFERENCES

[1] J. Gemrot, R. Kadlec, M. Bida, O. Burkert, R. Pibil, J. Havlicek, L. Zemcak, J. Simlovic, R. Vansa, M. Stolba, T. Plch, C. Brom, "Pogamut 3 Can Assist Developers in Building AI (Not Only) for Their Videogame Agents," Agents for Games and Simulations, *LNCS 5920*, Springer, pp. 1–15, 2009.

[2] K.O. Stanley and R. Miikkulainen, "Evolving Neural Networks Through Augmenting Topologies," *Evolutionary Computation*, Vol. 10, No. 2, pp. 99–127, 2002.

[3] M. Cottrell, B. Hammer, A. Hasenfu, and T. Villmann, "Batch neural gas", *WSOM 2005*, 2005.

[4] H. Hoang, S. Lee-Urban, and H. Munoz-Avila, "Hierarchical Plan Representations for Encoding Strategic Game AI," *Proc. of Artificial Intelligence and Interactive Digital Entertainment 2005*, 2005.

[5] M. McPartland and M. Gallagher, "Learning to be a Bot: Reinforcement Learning in Shooter Games," *Proc. of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference*, pp. 78–83, 2008.

[6] B. Soni, and P. Hingston, "Bots Trained to Play like a Human Are More Fun," *Proc. of IJCNN 2008*, pp. 363–369, 2008.

[7] B. Gorman and M. Humphrys, "Towards Integrated Imitation of Strategic Planning and Motion Modeling in Interactive Computer Games," *Proc. 3rd ACM Annual International Conference in Computer Game Design and Technology (GDTW 05)*, pp. 92–99, 2006.

[8] B. Gorman and M. Humphrys, "Imitating Learning of Combat Behaviors in First Person Computer Games," *Proc. 10th International Conference on Computer Games: AI, Mobile, Educational & Serious Games*, 2007.

[9] P. Hingston, "A Turing Test for Computer Game Bots," *IEEE Transactions on Computational Intelligence and AI In Games*, Vol. 1, No. 3, pp. 169–186, 2009.

[10] D. Wang, B. Subagdja, A.-H. Tan, and G.-W. Ng, "Creating Human-like Autonomous Players in Real-time First Person Shooter Computer Games," *Proc. 21st Annu. Conf. Innovat. Appl. Artif. Intell.*, pp. 173–178, 2009.

[11] R. Arrabales and J. Munoz, "The Awakening of Conscious Bots: Inside the Mind of the 2K BotPrize 2010 Winner," http://aigamedev.com/open/articles/conscious-bot/, Oct. 21, 2010 (last accessed Jun. 23, 2011).

[12] S. Murakami, T. Sato, A. Kojima, D. Hirono, N. Kusumoto, and R. Thawonmas, "Outline of ICE-CEC2011 and Its Mechanism for Learning FPS Tactics," *Human-like Bot Workshop at CEC 2011*, http://botprize.org/hlabstracts2011/ICE-CEC2011.pdf, 2011 (last accessed Jun. 23, 2011).

[13] D. Hirono and R. Thawonmas, "Implementation of a Human-Like Bot in a First Person Shooter: Second Place Bot at BotPrize 2008," *CD-ROM Proc. of Asia Simulation Conference 2009 (JSST 2009)*, Shiga,Japan, Oct. 7-9,2009, http://www.ice.ci.ritsumei.ac.jp/~ruck/PAP/jsst09-hirono.pdf (last accessed Jun. 23, 2011).