

ニューラルネットワークによる格闘ゲームAI の難易度調整及び 行動多様性向上手法

中川 明紀* 柴崎 智哉* 逢坂 翔太* Ruck Thawonmas*

*立命館大学大学院 理工学研究科

抄録：商業用ゲームにおけるゲーム AI の難易度調整は様々な研究がなされているものの、未だ容易な作業ではない。本研究ではニューラルネットワークを用いた格闘ゲーム AI の改良を行いその性能について検証、考察を行う。その改良に、ニューラルネットワークの学習の教師信号を現在の成績に応じて調整する事で、AI をユーザスキルに見合った強さにする手法を提案する。実験で格闘ゲームのシミュレータを用いて既存手法との比較実験を行い、提案手法の有効性を示す。

キーワード：ニューラルネットワーク, 機械学習, ゲーム AI

Adjustment of Game Difficulty and Improvement of Action Variety in Fighting Action Games Using Neural Networks

Akinori Nakagawa* Tomoya Shibazaki* Shota Osaka* Ruck Thawonmas*

*Graduate School of Science and Engineering, Ritsumeikan University

Abstract: Although there have been various researches on adaptive game AI for commercial games, the task of implementing adaptive AI is still difficult. In this paper, we present a method for improving an exiting fighting-action-game AI using neural networks, conduct performance evaluation, and give discussions on the results. To make the skill of AI correspond to that of the user, the presented method defines a supervised signal based on the current fighting performance. Evaluation is done using a game simulator where the effectiveness of the present method is shown when compared with the existing method.

Keywords: neural network, machine learning, game AI

1. はじめに

対人ゲームにおいてAIが操作するキャラの強さはゲーム自体の難易度に大きく関わり、ゲームそのものの楽しさにも大きく影響する[1]。しかし難易度の感じ方は個人の適性や習熟といったユーザスキルにより違うため、その調整は難しくまた限界がある。そこで学習AIによる動的な難易度調整が求められる。

本稿では、ニューラルネットワーク（以下、NN）を格闘ゲームに適応した手法[2]に着目した。この手法は様々な検証が行われており、ユーザより強くなるという目的に関しては有効性が示されている。しかし先に述べたように実際のゲームにおいてはユーザスキルとのバランスを考慮した難易度が期待されており、適度な弱体化が必要となる。また従来手法では学習するにつれて行動が収束していき、最終的に行動の多様性が失われるという問題もあった。

そこで本論文ではNNの学習に用いる教師信号を現在の成績に応じて調整する事で、AIをユーザスキルに見合った強さに調整し、またAIの行動多様性を向上させる手法を提案する。そして提案手法と従来手法の比較検証を行う。

2. 関連研究

昨今 AI 技術は色々な側面でデジタルゲームに応用されている[3][4]。Khooら[5]はキャラクタの振る舞いが階層化された仮想ユーザと、単純なテキスト処理を用いたチャットボットを実装している。学習機能を持つAIに関しては、馬場ら[6]はNNと遺伝的アルゴリズムをコモンゲームに適用してプレイヤーのコピーを作成する手法を提案している。伊藤ら[7]はクラシファイアシステムの学習を速める手法を提案し、色当てゲームに適用してその有効性を検証している。また[8]では進化的アルゴリズムを用いた複数キャラクタの操作手法について述べられている。

機械学習の技術には強化学習がある。強化学習とは、ある状況に対してどのような行動をとるべきかを得られる報酬によって学習していく機械学習の一種である[9]。強化学習を実行する際には状態空間の膨大さが問題となる事が多い。そこでNNで状態空間を圧縮する手法[10, 11]などが用いられる。また他の問題として報酬をいかに与えるかと言う評価関数の問題がある。提案手法ではこの問題に焦点を当てている。

表1 行動の詳細

行動名	idle	forward	backward	jump	down	guard
待機時間	1	2				
スコア	-					
有効距離	-					
行動名	low-punch	high-punch	low-kick	high-kick	special	
待機時間	4	6	8	10	12	
スコア	1	2	3	4	5	
有効距離	0-2		2-3		3-5	

3. シミュレータ

本手法の検証のために[2]を元にしたシミュレータを作成した。これは1対1の対戦アクションゲームを模したもので、距離と時間の概念を持っている。このシミュレータの状態数は3366である[2]。

それぞれの行動には表1に示す待機時間が設定されており、さらに攻撃には有効距離が設定されている。待機時間とは行動を実行してから効果が出るまでの時間であり、待機時間の単位はクロックとする。

また、行動名にlowと付く攻撃は相手のjumpで、highと付く攻撃は相手のdownで無効とされる。またguardは相手のspecialの獲得スコアを半減させ、その他の攻撃を無効とする。

行動待機中に攻撃を受けた際は行動がキャンセルされる。また待機中に自発的に行動を変更およびキャンセルすることは不可能とする。待機が終了し攻撃を相手に当てると自分にスコアが加算される。このスコアを学習の教師信号決定や手法の評価指標とする。

キャラクタAとBの戦闘の一例を図1に示す。この例ではAよりBの方が先にhigh-punchの待機時間を終了させているため、Bが攻撃を当ててスコアを取得し、Aのhigh-punchはキャンセルされる。

4. 既存手法

本章では、本研究の元となった既存研究[2]について基本構成、学習方法、問題点の流れで述べる。

4.1 基本構成

既存研究ではゲームAIに使用するNNの構成として中間層が1層の多層パーセプトロンを用いる[12]。

NNの入力は以下の6種とする。

- ・ 現在の相手の行動
- ・ 相手との距離
- ・ 相手の残り行動待機時間
- ・ 相手の過去の行動（最近3回分）

入力ノードは同じ6個とし、各入力値はそのまま自然数

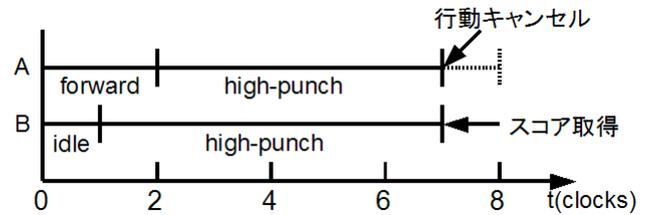


図1 戦闘の一例

でノードに入力する。

出力値の計算にはフィードフォワード型ニューラルネットの仕組みを用いる。出力を得るまでの計算は以下ようになる。

$$h_j = f\left(\sum_{i=1}^{N_i} (x_i \cdot w_{ij})\right)$$

$$z_j = f\left(\sum_{i=1}^{N_h} (h_i \cdot v_{ij})\right)$$

$$f_x = \frac{1}{1 + e^{-x}}$$

ここで $f(x)$ はシグモイド関数である。 h_j はj番目の中間ノードの出力値、 N_i は入力ノードの数、 x_i はi番目の入力ノードの入力値、 w_{ij} はi番目の入力ノードからj番目の中間ノードへの結合係数（以下重み）である。 z_j はj番目の出力ノードの出力値、 N_h は中間ノードの数、 v_{ij} はi番目の中間ノードからj番目の出力ノードへの重みである。

出力ノードは行動の種類と対応した11個とする。その中で最も出力が大きいノードに対応した行動を行う。

4.2 学習方法

学習は誤差逆伝播法を用いて重みを更新する事で行う。学習は行動完了時またはキャンセル時に行われ、その時に実行された行動に関係する重みだけを更新の対象とする。NNを用いたキャラクタをAIで表し、現時点tでのAIの行動を行動j（すなわち、j番目の出力ノードに対応する行動）とすると、学習に用いる教師信号は以下の式で決定する。

$$d_j = \begin{cases} \Delta/5 & (\text{GET}(\text{AI}, t) > \text{GET}(\text{OC}, t)) \\ 0 & (\text{GET}(\text{AI}, t) \leq \text{GET}(\text{OC}, t)) \end{cases}$$

ここで d_j はNNの学習に用いるj番目の出力ノードの教師信号とする。OCは相手のキャラクタを指し、実際のゲームではユーザとなる。GET(x, t)はxがt時点での行動によって得たスコア、 Δ は

$$\text{GET}(\text{AI}, t) - \text{GET}(\text{OC}, t) \text{とする。}$$

行動はそれを開始した状況に対して、教師信号を用いて学習させていく。移動行動はそれ自体でスコアを

得ることが出来ないが、スコアを得た行動の直前の移動については、 Δ を半分にして学習を行う。

出力層から中間層への重みの更新式は以下の通り。

$$v_{ij} = v_{ij} + \alpha(t) \cdot \delta_j \cdot h_j$$

$$\delta_j = z_j \cdot (1 - z_j) \cdot (d_j - z_j)$$

ここで $\alpha(t)$ は学習率である。

中間層から出力層への重みを更新したら次は入力層から中間層への重みを更新する。更新式は以下のようになる。

$$w_{ij} = w_{ij} + \alpha(t) \cdot \delta_j \cdot x_i$$

$$\delta_j = h_j \cdot (1 - h_j) \cdot \sum_{k=1}^{N_h} w_{jk} \delta_k$$

4.3 問題点

論文[2]ではランダムな行動を行う仮想ユーザ(以下OC)と一定のパターンを繰り返すOCの2種を相手にAIの性能検証を行っている。前者の実験では3倍、後者の実験では最大で30倍弱のスコアで仮想ユーザに勝利している。

しかし先にも述べたように、実際のゲームではユーザスキルに合わせたAIが求められており、このままでは勝ちすぎていると言える。また行動が収束していき、最終的に一定のパターンを繰り返すようになるという性質もある。これらは面白さという観点からゲームの対戦相手となるAIとしてはふさわしくないと言える。

5. 提案手法

AIの強さを調整する手法として、学習時の教師信号を調整することで学習の方向を変化させる手法を提案する。提案手法では現在までの双方の成績を常に監視し、相手に負けている場合は通常の学習を行う。そして勝っている場合は負けるように学習を行う。学習の教師信号の決定は以下の通りである。

$$\text{flag} = \begin{cases} 0 & \text{threshold} \geq \text{SCORE}(\text{AI}, t) / \text{SCORE}(\text{OC}, t) \\ 1 & \text{threshold} < \text{SCORE}(\text{AI}, t) / \text{SCORE}(\text{OC}, t) \end{cases}$$

If flag=0 then

$$d_j = \begin{cases} \Delta/5 & (\text{GET}(\text{AI}, t) > \text{GET}(\text{OC}, t)) \\ 0 & (\text{GET}(\text{AI}, t) \leq \text{GET}(\text{OC}, t)) \end{cases}$$

表2 仮想OCの行動パターン

1	f, hp, b, lk, b, s, f, hk
2	f, hp, b, lk, f, lp, b, hk
3	b, s, f, lk, b, s, f, hk
4	lp, lp, b, lk, lk, f, lp, lp

If flag=1 then

$$d_j = \begin{cases} 0 & (\text{GET}(\text{AI}, t) > \text{GET}(\text{OC}, t)) \\ -\Delta/5 & (\text{GET}(\text{AI}, t) \leq \text{GET}(\text{OC}, t)) \end{cases}$$

ここでflagは弱体フラグを指し、thresholdは弱体化の判定を行う閾値、SCORE(x, t)はt時点でのxの累計スコアとする。

NNへの入力は既存手法の入力に弱体フラグを加えた7種、入力ノードも同じ7個とする。中間層および出力層、行動決定の方法や移動行動の学習については既存手法と同じとする。

6. 実験

今回は提案手法の有効性を示すために

- ・ 提案手法によるAI
- ・ 既存手法によるAI
- ・ ランダム行動を行うAI

の3種をシミュレータ上でパターン行動を繰り返すOCと対戦させることで比較検証を行った。提案手法と既存手法によるAIは前述した仕様で作成した。ランダム行動を行うAIは行動決定を完全にランダムで決定するものである。

OCの行動はChoら[2]の行った実験では、表2に示す4種のパターンの内1つのパターンの繰り返しとなっている。これは人間の行動を1つのパターンの繰り返しとして実験を行っているためである。本実験ではOCの動きをより人間らしくするためにこれら4種のパターンをランダムで実行し続けるようにして実験を行った。なお、表2の行動パターンは表1の行動名の頭文字を用いて短縮表記している。

実験の際に用いるパラメータは、戦闘時のキャラクター間の初期距離を3、距離の最大値を7、NNの中間ノード数を30とし、学習率は0.1で常に固定した。学習率以外のパラメータは[2]に基づいている。

各結果は1試行を100,000クロックとして100試行繰り返したデータの統計を取る。性能指標としてはスコア比を用いる。これはAIの累計スコアを分子、OCの累計スコアを分母として算出しており、比が1に近いほど対等な戦いが出来ていると言える。スコア比は1000クロック毎に記録し、1つ目なら100試行分の1つ目のデー

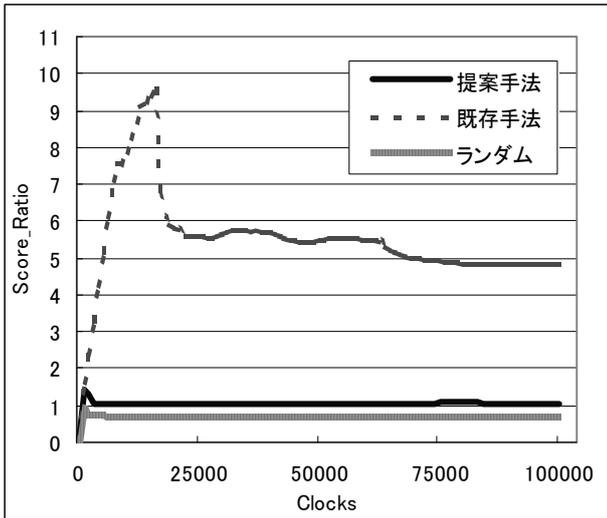


図2 累計スコア比の平均

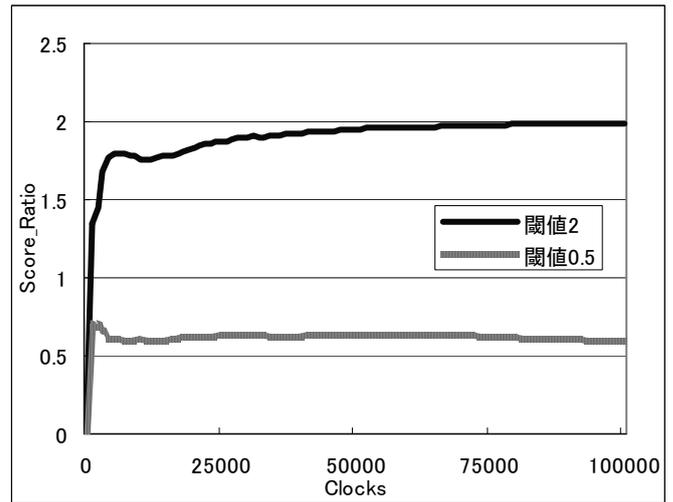


図4 閾値変更：累計スコア比の平均

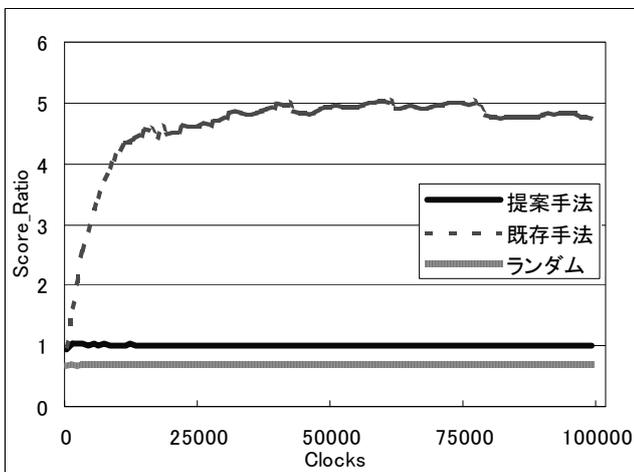


図3 累計スコア比の中央値

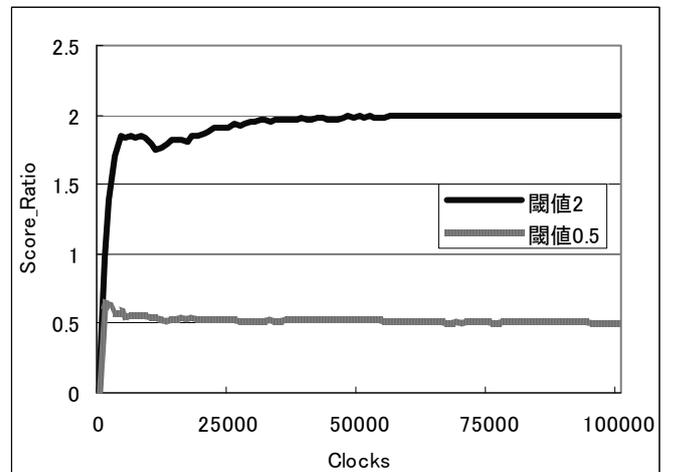


図5 閾値変更：累計スコア比の中央値

タを平均，または中央値を算出して出力している．すべてのノードの重みは各試行の開始時に $[-0.1, 0.1]$ の範囲の乱数で初期化する．

以上の条件を共通条件として3種の実験を行った．実験毎に変更するパラメータはその都度記載する．

6.1 累計スコア比の比較

累計スコア比を用いて3種の手法の比較実験を行った．本実験ではOCと同等の性能を持つAIを実現できるかを検証するために，提案手法におけるthresholdは1に設定した．

累計スコア比の平均及び中央値を図2，図3に示す．図2と図3から既存手法のAIはOCに対して高いスコア比で勝利している．一方提案手法のAIはスコア比を1に保ち続けている．このことからOCに対して動的に難易度を調整できていることがわかる．ランダムで行動するAIは負け続けた結果，スコア比が1を下回っている．この事から提案手法はOCのパターン行動に対応した上で，AIを一定の強さに保つことが出来ている事がわかる．

6.2 弱体化閾値の影響の検証

提案手法による弱体化学習の性能を検証するために，弱体化の判定を行う閾値thresholdを2と0.5に変更して実験を行った．

累計スコア比の平均及び中央値を図4，図5に示す．図4と図5から比がそれぞれの閾値に向かって収束していることがわかる．このことから提案手法は柔軟な難易度調整が可能であると言える．例えばユーザに対して強めの難易度を提示する場合には閾値を大きくする，弱めの難易度の時には閾値を小さくするといったことが可能である．

図4と図5において閾値を0.5にした時，比が完全に0.5へ収束しないのは，シミュレータの性質上確実に負ける行動が存在しないため，スコアを取得してしまうことがあるためであると考えられる．

6.3 行動多様性の比較

提案手法による行動多様性の向上を，行動分布図と行動分布のエントロピーを用いて検証した．

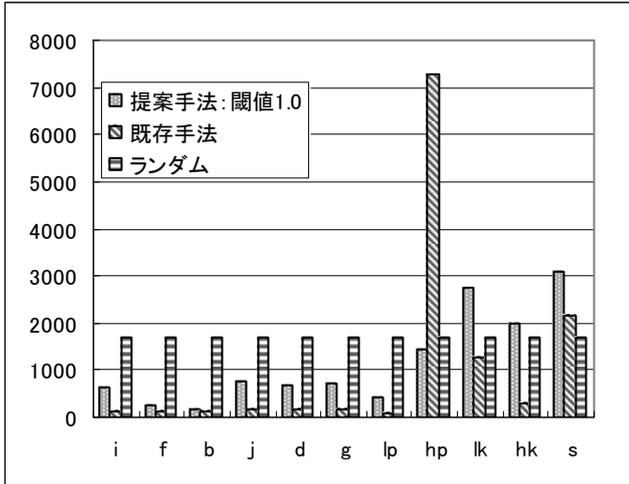


図6 手法別の行動分布図

全試行通しての手法別の行動分布図を図6、閾値別の提案手法の行動分布図を図7に、それらの行動分布のエントロピーを表3に示す。図6、図7は行動名を表2と同じく表1に示した行動名の頭文字を用いて表記している。

図6から既存手法の行動分布は攻撃行動に大きく偏っていることが解る。一方、提案手法の行動分布は既存手法に比べて分散しており、行動の多様性が向上していることがわかる。表3のエントロピーを見ても同じことが確認できる。これは従来手法の行動分布における攻撃行動に偏る性質が、弱体化によって抑えられているからである。

表3から提案手法の情報量はどの閾値でも既存手法に比べて高い事から、提案手法は多くの状況において行動多様性の向上に効果があるということがわかる。

図7から閾値を変更すると行動分布の傾向が変化していることがわかる。閾値を0.5に下げると行動が大幅に偏るのは、この行動は待機時間が非常に長い行動であることから、弱体化の結果として相手に当たりにくい行動を選択するようになったからであると考えられる。

7. おわりに

本論文では既存研究であるNNを用いた格闘ゲームAIに着目し、その問題点を述べ、改良するための手法を提案した。そして提案手法について既存手法との比較検証を行った。その結果として提案手法は動的にAIの強さを調整でき、行動の多様性の向上も確認された。また、閾値を変更してもその値にスコア比を収束させることが出来ることを示した。

今後の課題としては、学習速度のさらなる高速化や、相手のレベルが変化していく場合への対応等が挙げられる。また今回は元となった手法との比較のみにとど

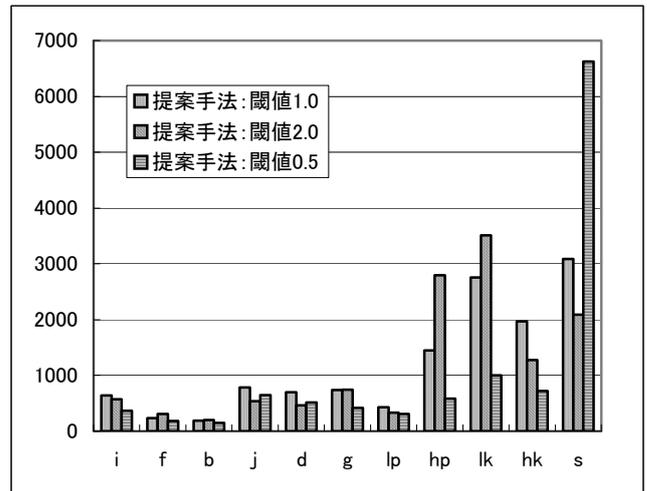


図7 閾値別の行動分布図

表3 行動分布のエントロピー

既存手法
1.86
ランダム
3.46
提案手法: 閾値 1.0
3.01
提案手法: 閾値 2.0
2.91
提案手法: 閾値 0.5
2.32

まったが、[10, 11]などの他の手法との性能比較も課題として挙げられる。そして最終的には人間を相手にした評価実験を行いたいと考えている。

参考文献

- [1] Jenova Chen: "Flow in Game", Communications of the ACM, Vol. 50, No. 4, pp. 31-34, 2007.
- [2] Byeong Heon Cho, Sung Hoon Jung, Yeong Rak Seong, Ha Ryoung Oh: "Exploiting Intelligence in Fighting Action Games Using Neural Networks", IEICE Trans. INF. & SYST., Vol. E89-D, No. 3, pp. 1249-1256, 2006.
- [3] 三宅 陽一郎: "デジタルゲームにおける人工知能技術の応用", 人工知能学会誌, 23 巻, 1 号, pp. 44-51, 2008.
- [4] 対馬 勝英: "ゲーム, アミューズメントの教育とAI", 人工知能学会誌, 19 巻, 1 号, pp. 35-42, 2004.
- [5] Aaron Khoo and Robert Zubek: "Applying

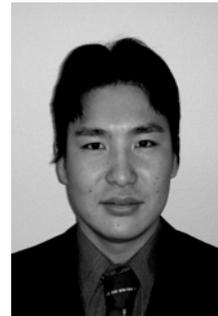
Inexpensive AI Techniques to Computer Games", IEEE Intelligent Systems, Vol. 17, No. 4, pp. 48-53, 2002.

- [6] 馬場 則夫, 北 富夫, 高河原 裕介, 小田 和広: "ニューラルネットならびに遺伝的アルゴリズムを活用したコンピュータゲーミングシステム", 計測と制御, 第36巻, 第6号, pp. 434-448, 1997.
- [7] 伊藤 寛隆, 田中 敏光, 杉江 昇: "人間の戦術を予測する機械学習システムの構築", 人工知能学会論文誌, 18巻, 3号, pp. 161-164, 2003.
- [8] Simon M. Lucas, Graham Kendall: "Evolutionary Computation and Games", IEEE Computational Intelligence Magazine, pp.10-18, February 2006.
- [9] Richard S.Sutton and Andrew G.Barto: Reinforcement Learning, A Bradford Book, MIT Press, cambridge, MA,(1998), (訳書:強化学習, 三上貞芳, 皆川雅章共訳, 森北出版,(2000)).
- [10] Bing-qiang Huang, Guang-yi Cao, Min Guo: "Reinforcement learning neural network to the problem of autonomous mobile robot obstacle avoidance," Proc. of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 pp. 85-89, August 2005.
- [11] Caihong Li, Jingyuan Zhang, Yibin Li : "Application of Artificial Neural Network Based on Q-learning for Mobile Robot Path Planning," Proc. of the 2006 IEEE International Conference on Information Acquisition, pp. 978-982, 2006.
- [12] R.P. Lippmann: "An introduction to computing with neural nets", IEEE ASSP Mag., Vol. 4, No. 2, pp. 4-22, 1987.



柴崎 智哉

2007年3月立命館大学情報学科卒業。現在、同大学大学院理工学研究科情報理工学専攻博士課程前期課程に在籍。ゲームAIの研究に従事。



逢坂 翔太

2006年3月立命館大学工学部情報学科卒業。2008年3月同大学大学院理工学研究科情報システム学専攻博士課程前期課程修了。同年4月より株式会社コナミデジタルエンタテインメントに勤務。在学中、ゲームAIの研究に従事。



Ruck Thawonmas

1987年タイ国Chulalongkorn大学工学部電気工学科卒業。1990年茨城大学大学院工学研究科情報工学専攻修士課程修了。1994年東北大学大学院工学研究科情報工学専攻博士課程修了。博士(工学)。日立製作所, 理化学研究所, 会津大学, 高知工科大学を経て, 2004年4月より立命館大学情報理工学部知能情報学科教授。ゲームAI, オンラインゲームにおける知的獲得, 双方向物語の自動生成に興味をもつ。IEEE (Senior Member), ACM, IGDA, 電子情報通信学会, 情報処理学会, ゲーム学会などの学会の各会員。

著者紹介



中川 明紀

2008年3月立命館大学知能情報学科卒業。現在、同大学大学院理工学研究科情報理工学専攻博士課程前期課程に在籍。ゲームAIの研究に従事。

2008年4月19日 論文受理

2008年10月15日 採録決定