Lighting Control System using an Actor - Critic type Learning Algorithm

Tomoyuki Hiroyasu
Faculty of Life and Medical Sciences
Doshisha University
Kyoto, Japan
tomo@is.doshisha.ac.jp

Akiyuki Nakamura
Graduate School of Engineering
Doshisha University
Kyoto, Japan
anakamura@mikilab.doshisha.ac.jp

Masato Yoshimi
Department of Science and Engineering
Doshisha University
Kyoto, Japan
myoshimi@mail.doshisha.ac.jp

Mitsunori Miki
Department of Science and Engineering
Doshisha University
Kyoto, Japan
mmiki@mail.doshisha.ac.jp

Hisatake Yokouchi
Faculty of Life and Medical Sciences
Doshisha University
Kyoto, Japan
hyokouch@mail.doshisha.ac.jp

***Abstract*—A novel lighting control system using the Actor - Critic algorithm was developed, in which users can set the brightness of the system through sensory operation, such as "much brighter" or "slightly darker". During development, this system must learn two states, i.e., the demands of the user and the brightness around the user. The Actor - Critic algorithm was applied for this purpose, and a simplified algorithm was developed. The effectiveness and usefulness of the proposed algorithm are discussed here through numerical simulations.**

***Keywords*-Lighting; Sensory scale; Reinforcement learning; Actor-Critic**

I. Introduction

Traditional lighting systems control multiple banks of lights at once. In the near future, new devices such as light-emitting diodes (LEDs) and organic light-emitting diodes (OLED) will change lighting environments, and the number of lights to be controlled will increase dramatically. At present, it is difficult to control lights in such numbers using conventional control systems. The development of methods to control these new devices on an individual basis will allow the system to perform intelligent actions and achieve various lighting environments[1][2]. It will be necessary to change the lighting system user interface (UI) to address this increase in number of lights. Adjusting the brightness of many lights on an individual basis places a large burden on the user[3][4]. The development of a UI capable of interpreting users' sensory indications, such as "much brighter" or "slightly darker", will be very convenient for users[5][6]. However, the definitions of the sensory scale, such as "much" or "slightly", differ for each user[7]. To address this issue, we have developed a learning mechanism consisting of an Actor - Critic algorithm[8], which is a reinforcement learning method, to allow the system to learn the users' sensory lighting requirements. This system changes the illuminance according to two states: the user's sensory indications and the degree of change in illuminance. The amount of brightness is changed corresponding to the users' sensory indications. However, the amount of brightness change can be varied relative to the present level even with the same sensory indication.

This is because the users' perceptions differ with variation in the brightness of the surrounding environment[9][10]. Thus, the Actor - Critic algorithm should learn two states: the user's sensory indications and the degree of change in brightness. When there are m choices for sensory indication and n choices for the degree of change in brightness, the total number of conditions is mn. The conventional Actor - Critic algorithm should learn all of these states and conditions[11][12]. However, the target is a lighting system, which we assume will have many possible conditions. This increases the number of possible states and results in huge computational costs. The time required for learning should be as short as possible, and an efficient learning algorithm is necessary for this system. To make learning of the users' sensory scale more efficient, we propose a Two-Actor - Critic algorithm in this paper, which is an Actor - Critic algorithm with two types of Actor applying to the two types of state.

II. Lighting Control System using Sensory Operation

As described in the previous section, it is efficient that users can operate a lighting system by sensory order, such as "much brighter" or "slightly darker"[5][6]. In this section, the overview and the requirements of the lighting control system using sensory operation are described.

A. Overview

An overview of the proposed system is shown in Fig. 1.

This system consists of a control computer, control devices, lights, and illuminance sensors. Each light can be controlled by the control computer on an individual basis. The

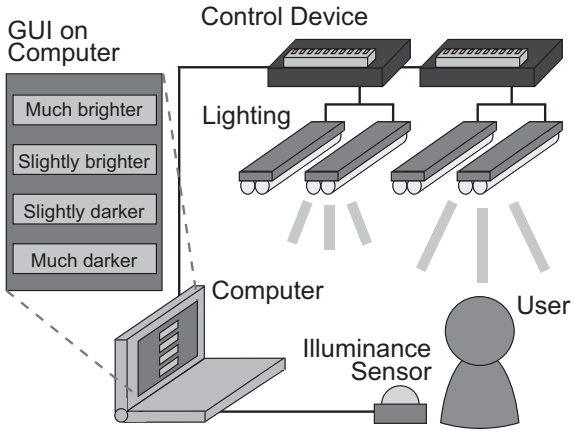

Fig. 1. Overview of the Lighting Control System through Sensory Operation

user conveys the required amount of change in brightness to the system via the Graphical User Interface (GUI) on the computer, as shown in Fig. 1, or using voice input, etc. The user then selects the required change from the eight decision branches presented in the GUI: "very much brighter", "much brighter", "brighter", "slightly brighter", "slightly darker", "darker", "much darker", and "very much darker". The system changes brightness in the room according to the user input. This process is repeated until the user is satisfies with the ambient lighting conditions.

B. Requirements

The requirements of this system are as follows:

- Control of Illuminance
  Conventionally, users used to adjust the luminance [i.e., the index of the brightness illuminating an object in units of candela per square meter (cd/m^2)] of lighting by a switch. However, users want a change of the illuminance [i.e., the index of the brightness emitted by the light in units of lux (lx)] around them. Therefore, this system operates brightness in the room based not on luminance, but on illuminance. The system achieves the illuminance control by using a illuminance sensor, because it can control only the luminance of lighting directly. The control algorithm of the system described here is the same as the references [13][14].
- Learning of the Sensory Scale
  As the sensory scale, such as the definitions of "very" and "slightly", differs for each user, the system learns the sensory scale for each user[7]. Even if a user gives the same instructions to the system, the changes in illuminance by the system will vary in relation to the illuminance around the user[9][10]. Therefore, the system should learn the amount of change in illuminance according to both the user's instructions and also the illuminance around the user. The number of learning cycles of the system is equal to the number of user demands on this system. Therefore, the system requires an efficient algorithm to reduce the time for learning by the system.

The precise learning of the sensory scale is described in the following section.

III. Learning of the Sensory Scale

This section describes reinforcement learning and the Actor - Critic algorithm. The proposed method, i.e., the Two-Actor - Critic Algorithm, is also described.

A. Reinforcement Learning

Reinforcement learning is a method to learn action policy with adaptation to the various states of an environment by trial and error[15][16]. Reinforcement learning includes the concept of the environment, which is the target problem on learning and has several states, and the concept of an agent, which plays a learning role and decides on the appropriate action in relation to the state of the environment based on its own policy. Because the agent needs a guidepost to learn, the environment provides the agent with a reward as an evaluation of the action. In this problem, users are included in the environment, and reward is determined according to changing of users' demand. The agent updates the policy to maximize the eventual sum of rewards. In reinforcement learning, the Markov Decision Process (MDP) is used to model the dynamics of the environment. MDP is the probability process model in which the probability that an event will occur in the future is determined according not to the past states but is based only on the current state. Therefore, the sum R_t of rewards is eventually expressed as (1)[16].

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{1}$$

r_t is the reward in discrete time *t*. γ is the discount rate, which fulfills $0 \leq \gamma \leq 1$. The value $V^{\pi}(s)$ of the state *s* is formulated as (2)[16].

$$V^{\pi}(s) = E_{\pi}\{R_t | s_t = s\} = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\} \tag{2}$$

s_t is the state in discrete time *t*, and π is the policy. $E_{\pi}\{\}$ is the expected value given when the agent follows the policy π. The purpose of learning is the acquisition of the policy that maximizes the value $V^{\pi}(s)$ for every state *s*. In the problem addressed by this system, two types of state, i.e., the illuminance around the user and the demands of the user, are decided from within discrete spaces (the state of the illuminance around the user is delimited with some intervals). The action is decided from within a consecutive space, which is the amount of change in illuminance.

B. Actor - Critic Algorithm

In the target system, states of actions are defined as consecutive values. Thus, this system uses an Actor - Critic algorithm, which is a type of reinforcement learning and is suitable for learning under such conditions[8][16]. In this algorithm, the agent consists of an Actor that selects the action and a Critic that evaluates the action. The Actor decides the action a according to the probabilistic policy $\pi(s)$. The Critic has the state value function $V(s)$, which indicates by how much each state is a good state. An overview of the Actor - Critic algorithm is shown in Fig. 2.

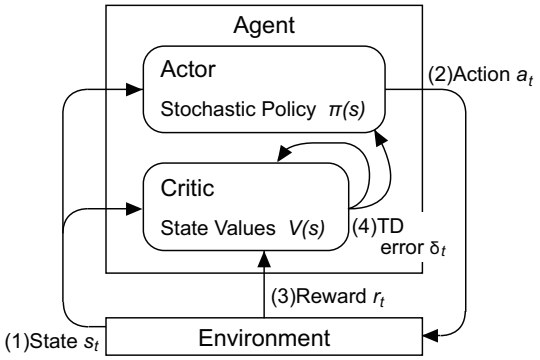

Fig. 2. Overview of Actor - Critic Algorithm

The processes shown in Fig. 2 can be explained as follows:

(1) State Observation
The agent observes the state of the environment s_t.

(2) Action
The Actor decides on the action a_t according to the probabilistic policy $\pi(s_t)$.

(3) Reward
The state of the environment changes to the state s_{t+1} by the action a_t. Then, the environment gives the reward r_t to Critic as the evaluation of the action.

(4) Reinforcement Signal
The Critic observes the reward r_t and the next state of the environment s_{t+1}. The Critic then calculates the Temporal Difference error (TD error) δ_t as the reinforcement signal for the Critic and the Actor. TD error δ_t is expressed as (3)[16].

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \tag{3}$$

γ is the discount rate, which fulfills $0 \leq \gamma \leq 1$. According to TD error δ_t, the Critic updates the state value function $V(s_t)$ and the Actor updates the probabilistic policy $\pi(s_t)$. Updating of the state value function $V(s_t)$ is expressed as (4)[16].

$$V(s_t) \leftarrow V(s_t) + \alpha\delta_t \tag{4}$$

α is the learning rate, which fulfills $0 \leq \alpha \leq 1$. At the same time, the probabilistic policy $\pi(s_t)$ updates its parameters so that the selection probability of a_t is increased if TD error δ_t is positive. In addition, if TD error δ_t is negative, it is updated such that the selection probability of a_t is decreased. Then, if the system uses a normal distribution as the probabilistic policy, the mean and the standard deviation of the probabilistic policy $\pi(s_t)$ are updated.

C. Two-Actor - Critic Algorithm

In this system, there are two types of discrete state, i.e., the illuminance around user and the demands of the user. In a conventional Actor - Critic algorithm, the system must learn all of the states and conditions[11][12]. However, as discussed in the previous section, efficient learning is required in this system to reduce the user operation burden. Therefore, we discuss the Two-Actor - Critic algorithm, which learns two types of state separately with two types of Actor, i.e., the Actor on illuminance around the user and the Actor on user demands. The amount of illuminance change perceived by the user is proportional to the height between lightings and users[9][10]. Therefore, in this system, the Actor on illuminance around the user plays a role in deciding the parameters for tuning the scale of the Actor on user demands. The relation between the two types of Actor is shown in Fig. 3. This system uses a normal distribution as the probabilistic policy of the Actors.

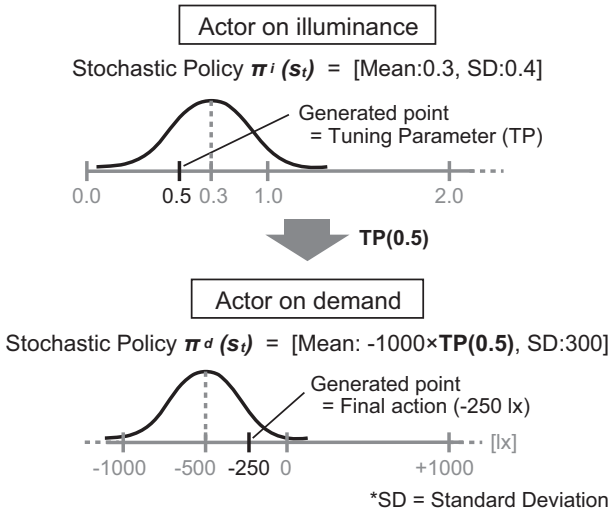

Fig. 3. Relation between the two types of Actor

As shown in Fig. 3, the Actor on illuminance around the user first decides the action, which is the parameter

to tune the Actor on user demands, by the probabilistic policy $\pi^i(s_t)$. The tuning parameter should be larger when the illuminance around the user is high. Second, the Actor on user demands fixes the mean of the normal distribution by the tuning parameter and the foundation value of the mean, and decides the amount of change in illuminance as the action according to the probabilistic policy $\pi^d(s_t)$. In this way, the system uses the learning algorithm with two types of Actor. The overview of this Two-Actor - Critic algorithm is shown in Fig. 4.

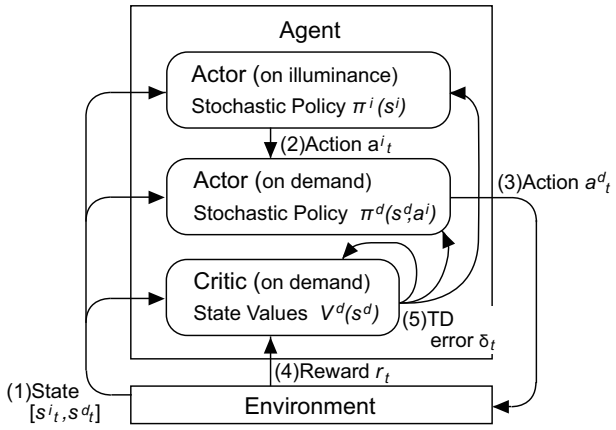

Fig. 4. Overview of Two-Actor - Critic Algorithm

As shown in Fig. 4, although there are two types of Actor (the Actor on illuminance around the user and the Actor on user demands), there is only one type of Critic; the Critic on user demands. This is because the role of the Critic is to evaluate the action of the Actor and the evaluation guide is only how much the action brings the user demand close to the satisfactory state. The processes shown in Fig. 4 are explained below.

(1) State Observation
The agent observes the state of the environment $[s^i{}_t, s^d{}_t]$.

(2) Action of Actor on Illuminance around the User
The Actor on illuminance around the user decides the action $a^i{}_t$ according to the probabilistic policy $\pi^i(s^i{}_t)$.

(3) Action of Actor on user demands
The Actor on user demands decides the action $a^d{}_t$ according to the probabilistic policy $\pi^d(s^d{}_t, a^i{}_t)$. The function of $a^i{}_t$ on $\pi^d(s^d{}_t, a^i{}_t)$ is shown in Fig. 3.

(4) Reward
The state of the environment changes to the state $[s^i{}_{t+1}, s^d{}_{t+1}]$ by the action $a^d{}_t$, the environment gives the reward r_t to Critic as the evaluation of the action.

(5) Reinforcement Signal
The state value function $V^d(s^d{}_t)$ of Critic and the probabilistic policy $\pi^i(s^i{}_t)$ and $\pi^d(s^d{}_t, a^i{}_t)$ of Actors are updated according to TD error δ_t which are also updated. These updating steps were described in paragraph III-B. As the scales of values in the two probabilistic policies differ, the learning rates of each policy should be set separately.

IV. Numerical Experiment

This section describes the numerical experiments performed to verify the effectiveness of the Two-Actor - Critic algorithm.

A. Target Problem

To confirm the basic characteristics of the proposed method, a virtual user was constructed and applied in this experiment instead of a human user. An example of the virtual user's sensory scale is shown in Fig. 5.

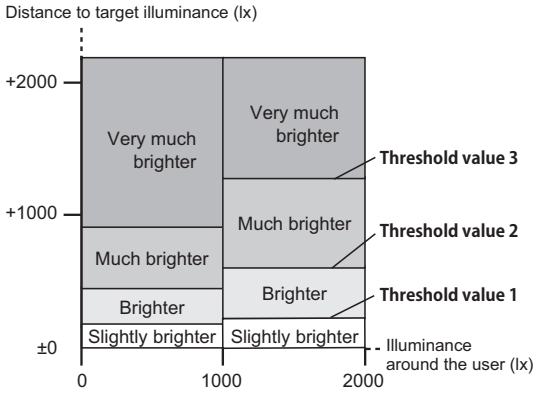

Fig. 5. Example of the Sensory Scale of the Virtual User

The horizontal axis in Fig. 5 shows the illuminance around the user. The virtual user has a target illuminance, and the vertical axis shows the sensory amount of illuminance to change to the target illuminance. In conclusion, the demands of the user, such as "Very much brighter" or "Slightly darker", are decided by the illuminance around the user and the difference between the illuminance around the user and the target illuminance. For the virtual users, threshold values (lx) were specified for the borders of each user demand in Fig. 5 as the sensory scale. The threshold values of Virtual User 1 used in this experiment are shown in Table I, where the border between "Slightly brighter" and "Brighter" is given a threshold value of 1, that between "Brighter" and "Much brighter" has a threshold value of 2, and that between "Much brighter" and "Very much brighter" has a threshold value of 3.

Table I
THRESHOLD VALUES OF VIRTUAL USER 1

Threshold value / Illuminance around user	Threshold value 1	Threshold value 2	Threshold value 3
0-1000 lx	+20 lx	+40 lx	+80 lx
1000-2000 lx	+80 lx	+160 lx	+320 lx
2000-3000 lx	+320 lx	+640 lx	+1280 lx
3000-∞ lx	+1280 lx	+2560 lx	+5120 lx

Table Ishows only the ”Brighter” user demands. However, if the illuminance obtained by subtracting the illuminance around the user from the target illuminance is negative, the threshold values in Table I also become negative. Virtual user 1 was designed such that the difference between learning accuracies in two cases was large. The first is where the conventional Actor - Critic algorithm is not applicable to the two types of state. Application of the Actor - Critic algorithm to one type of state judges the amount of change in illuminance according only to the user demands, and learns only eight states related to user demands. The second case is where the Actor - Critic algorithm is applied to two types of state but these two states are converted into one state.. The Actor - Critic algorithm applied to two types of state judges the amount of change in illuminance according to the user demands and the illuminance around the user, and learns 32 states (8 states $\times$ 4 states; 0 - 1000 lx, 1000 - 2000 lx, 2000 - 3000 lx, 3000 - ∞ lx).

Fig. 6 shows the results learned for Virtual User 1 by Actor - Critic algorithms applied to one and two types of state. In this experiment, the discount rate was 0.95 and the learning rates for both Critic and Actor were 0.5. As the reward, the environment gave a score of 800 if the user demand changed to ”satisfaction” and a score of -50 if it changed to other state.

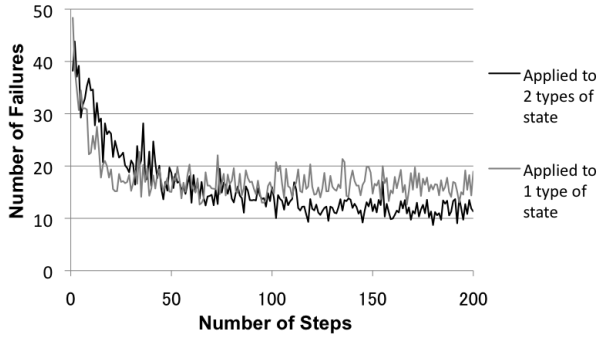

Fig. 6. Comparison of Actor - Critic Algorithms Applied to 1 and 2 Types of State

One step represents the period between the time that the illuminance around the user was set to the default value and that at which the user demand reached the state ”satisfaction”. The number of failures is the number of times that the user reentered the input during the simulations. The graphs in Fig. 6 show the median values of 30 trials. As shown in Fig. 6, the learning accuracy of the Actor - Critic algorithm applied to two types of state was higher than that applied to only one type of state. Therefore, in learning for Virtual User 1, the results indicated that applying the algorithm to two types of state yielded better results.

B. Experimental Results of Two-Actor - Critic Algorithm

Experiments were performed to verify the effectiveness of the Two-Actor - Critic algorithm. Fig. 7 shows that the Two-Actor - Critic algorithm learned the sensory scale of Virtual User 1, and presents the results obtained by applying the normal Actor - Critic algorithm to two types of state. In the Two-Actor - Critic algorithm, the discount rate was 0.95, the learning rates of the Critic and the Actor on user demands were 0.5 and 0.1, respectively, and the learning rate of the Actor on illuminance around the user was 0.0005. As the reward, the environment gave a score of 300 if the user demand changed to ”satisfaction” and a score of -50 if it changed to other state.

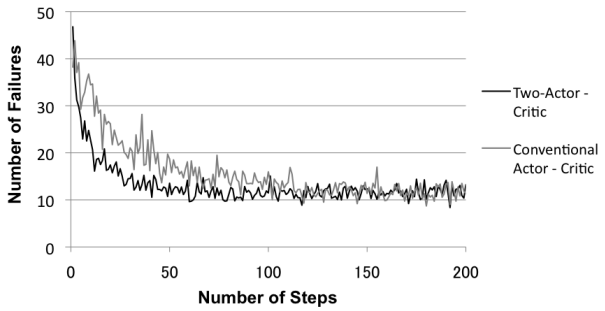

Fig. 7. Comparison of Two-Actor - Critic Algorithm and Conventional Actor - Critic Algorithm for Virtual User 1

As shown in Fig. 7, the learning accuracies of the two algorithms were equivalent at the time of convergence, and the Two-Actor - Critic algorithm yielded better results than the conventional Actor - Critic algorithm with regard to the speed of convergence.

C. Considerations

In the case where there were 8 states for the user demands and 4 states for illuminance around the user, the conventional Actor - Critic algorithm had to optimize 32 (8 $\times$ 4) normal distributions. On the other hand, the Two-Actor - Critic algorithm had to optimize only 12 (8 + 4) normal distributions. Thus, the Two-Actor - Critic algorithm was better than the conventional Actor - Critic algorithm with regard to the speed of convergence. To verify this consideration, we carried out an experiment with a virtual user, called ”Virtual

User 2", in which we increased the number of states for illuminance around the user. Although the threshold values of Virtual User 1 were delimited at intervals of 1000 lx, Virtual User 2 had threshold values delimited at intervals of 500 lx (0 - 500 lx, 500 - 1000 lx, 1000 - 1500 lx, 1500 - 2000 lx, 2000 - 2500 lx, 2500 - 3000 lx, 3000 - 3500 lx, 3500 - ∞ lx). The experimental results are shown in Fig. 8, and the parameters of each algorithm were the same as those in the experiments described in sections IV-A and IV-B.

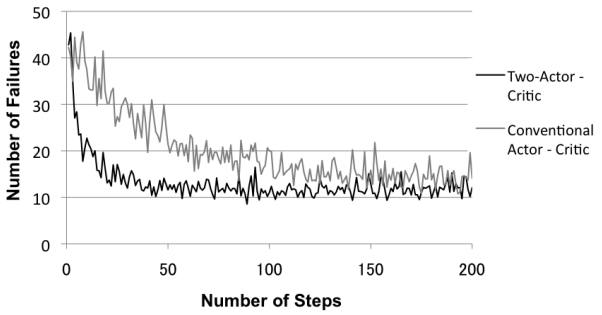

Fig. 8. Comparison of Two-Actor - Critic Algorithm and Conventional Actor - Critic Algorithm for Virtual User 2

As shown in Fig. 8, the difference in speed of convergence between the Two-Actor - Critic algorithm and the conventional Actor - Critic algorithms was larger than that in Fig. 7. The results of this experiment indicated that learning efficiency can be improved by applying two types of Actor to two types of state in the Two-Actor - Critic algorithm.

V. Conclusions

Here, we proposed a lighting control system through sensory operation to minimize the operation burden associated with a lighting system. To realize sensory operation, this system learns the users' sensory scale, such as "very" or "slightly", using an Actor - Critic algorithm. This system must learn efficiently to decrease user burden. There are two types of state in the target environment of this system, and the conventional Actor - Critic algorithm has to learn all states combined with two types of state. To improve learning efficiency, a learning algorithm involving the application of two types of Actor to two types of state was proposed, i.e., a Two-Actor - Critic algorithm. We verified the effectiveness of this algorithm by experiments using virtual users. The results indicated that this algorithm has learning accuracy equivalent to that of the conventional Actor - Critic algorithm. In addition, the proposed algorithm was shown to have a faster learning speed than the conventional Actor - Critic algorithm. Further studies are required to verify the effectiveness under different environments and experiments with real human users should also be performed.

References

[1] M Ashibe, M Miki, T Hiroyasu, : Distributed Optimization Algorithm for Lighting Color Control using Chroma Sensors, 2008 IEEE International Conference on Systems,Man and Cybernetics, pp.174-178, 2008.

[2] Vipul Singhvi, Andreas Krause, Carlos GuestrinJames H. Garrett Jr, H. Scott Matthews : Intelligent light control using sensor networks, Proceedings of the 3rd international conference on Embedded networked sensor systems, pp.218-229, 2005.

[3] Barry Brumitt, JJ Cadiz : "Let There Be Light" Examining Interfaces for Homes of the Future, Proceedings of Interact'01, 2001.

[4] Krzysztof Gajos, Daniel S. Weld : SUPPLE -Automatically Generating User Interfaces-, Proceedings of the 9th international conference on Intelligent user interfaces, 2004.

[5] Liz C. Throop : Field of play: sensual interface, Proceedings of the 2003 international conference on Designing pleasurable products and interfaces, pp.82-86, 2003.

[6] K Tsukada, M Yasumura : Ubi-Finger: Gesture Input Device for Mobile Use, Proceedings of APCHI 2002, Vol.1, pp.388-400, 2002.

[7] Stevens SS : On the psychophysical law, Psychological Review, Vol.64(3), pp.153-181, 1957.

[8] Andrew G. Barto, Richard S. Sutton, Charles W. Anderson : Neuronlike adaptive elements that can solve difficult learning control problems, Neurocomputing: foundations of research, pp.535-549, 1988.

[9] Tomoaki Sikakura, Hiroyuki Morikawa, Yoshiki Nakamura : Perception of Lighting Fluctuation in Office Lighting Environment, Journal of Light and Visual Environment, Vol.27, No.2, pp.75-82, 2003.

[10] D. H. Kelly : Visual Responses to Time-Dependent Stimuli. I. Amplitude Sensitivity Measurements, Journal of the Optical Society of America, pp.422-429, 1961.

[11] Jun Morimoto, Kenji Doya : Acquisition of Stand-up Behavior by a Real Robot using Hierarchical Reinforcement Learning, Proceedings of the Seventeenth International Conference on Machine Learning, pp.623-630, 2000.

[12] G. R. Gajjar, S. A. Khaparde, P. Nagaraju, S. A. Soman : Application of actor-critic learning algorithm for optimal bidding problem of a GenCo, IEEE Transactions on Power Engineering Review, Vol.18, No.1, pp.11-18, 2003.

[13] M Miki, T Hiroyasu, K Imazato, M Yonezawa : Intelligent Lighting Control using Correlation Coefficient between Luminance and Illuminance, Proc IASTED Intelligent Systems and Control, Vol.497, No.078, pp.31-36, 2005.

[14] M Miki, E Asayama, T Hiroyasu : Intelligent Lighting System using Visible-Light Communication Technology, 2006 IEEE Conference on Cybernetics and Intelligent Systems,pp.1-6, 2006.

[15] T Mitchell, B Buchanan, G DeJong, T Dietterich, P Rosenbloom, A Waibel : Machine Learning, Annual Review of Computer Science, Vol.4, No.1, pp.417-433, 1990.

[16] R. S. Sutton, A Barto : Reinforcement Learning -An Introduction-, The MIT Press, 1998.