Trajectories Tracing for a Pitching Robot based on Human Recognition

Osamu Yuuki, *Member, IEEE*, Kunihiro Yamada and Naoyuki Kubota, *Member, IEEE*

***Abstract*— In this study, we discussed human recognition method for the trajectory tracking. Visual perception is very important to realize the feature extraction from the time series of images, but it is very difficult to perform the object tracking. We classified errors occurring in the throwing; errors caused by internal factors and errors caused by external factors. At first, we propose a theoretical method of calculating the trajectory of a ball from the set-values. Second, we propose the method of calculating the trajectory from parts positions. These positions are recognized from the image captured by CCD camera. In this calculation, the pattern recognition were used to find parts positions. These positions are the released position of a ball and the position of the fulcrum. By this way, e.g., we can find the released position error of a ball and the speed error of a ball caused by internal factors. We used parabola equations in these calculations of trajectories. Third, we propose the method of extracting the trajectory from the time series of images captured by CCD camera. The specifications of camera are "480 by 360 pixels", "RGB color" and "29 frames per second". We propose the method of recognizing a position of the flying ball from images of the movie, directly. The robot plots the trajectory of a flying ball. By using this method, we can find the errors caused by external factors, e.g., we can suppose the influence of the air resistance working to the ball. Finally, we performed the experiment of the trajectories tracing for a pitching robot based on human recognition.**

I. Introduction

Human intelligence is useful to create robot intelligence. Especially, human-like visual intelligence is very important to realize the robot intelligence. The researches on vision can be divided into two categories of passive vision and active vision. "Active Vision" contains its own source of radiation as opposed to "Passive Vision". And, "Active Vision" links between perception and action [1]. In our previous works, we discussed the effectiveness of evolutionary robot visions and active vision based on perceiving-acting cycle. In the study on evolutionary robot vision, we discussed the analogy between human visual search and evolutionary search, and the effectiveness of the distributed evolutionary search.

However, we have not yet discussed actual high-level human recognition mechanism until now.

Therefore, in this study, we focus on visual tracking based on human recognition mechanism. We analyzed behaviors of human throwing as theoretical approach. And, we conducted experiment to detect the trajectory of a thrown ball by a robot.

If the self-feedback control will be carried out using the above-mentioned methods in the robot, the throwing will become almost human-like. The control method by the pattern recognitions has been studied [2]. Furthermore, the fuzzy control system had been studied [3][4]. And, the robot learning [5][6][7] system will be needed, too. By this study and prior studies, the robot motions will be more intelligent motions.

Manuscript received September 5, 2009.

This work was supported in part by Japan Science and technology Agency.

O. Yuuki is with the Professional Graduate School Embedded Technology, Tokai University, Tokyo 108-8619, Japan, (corresponding author to provide phone: +81-35475-5508; fax: +81-3475-5507; e-mail: 8aejp017@mail.tokai-u.jp).

K. Yamada is with the School of Information Technology and Science, Tokai University, Kanagawa 259-1292, Japan, (e-mail: yamadaku@tokai.ac.jp).

N. Kubota is with the Graduate School of System Design, Tokyo Metropolitan University, Tokyo 191-0065, Japan, (e-mail: kubota@tmu.ac.jp).

II. Theory

A. The Throwing Behavior

When a human throws a ball, the human imagines the trajectory of a thrown ball beforehand, and regulates the angle and speed of the arm to muscles based on cooperative structures of muscles. After the human throws a ball, the human predicts and traces the trajectory of the ball by one's eyes. The check process can use the released position and the arm speed as indirect information for the prediction of the trajectory of the thrown ball. This kind of additional information, not only the state of the ball itself, is very useful to predict the trajectory of the ball. And, the human will evaluate whether or not the ball reaches the target position. The image of the actual trajectory of the ball should be used in the evaluating process. Human can use such important and useful information. This kind of perception can be done in the high level of human recognition based on experience knowledge. We discuss the following things; (1) how to predict the trajectory of the throwing behavior, (2) physical phenomena on the parabola trajectory of the thrown ball, (3) the reason why the difference of trajectories between theoretical calculation of a ball and experimental results.

However, it is very difficult to measure human throwing behaviors directly. Therefore, in order to simplify the problem, we develop a robot.

In case of the robot, after the robot calculated the set-values from the target points, the robot sets the set-values to an actuator as arm angle to throw and the arm speed. Then the robot calculates the target trajectory from the set-value. The equation (1) and (2) are used in the calculations of a parabola. Next, an image processing method is used for recognizing the released position and the fulcrum position. The arm angle and the arm length are calculated using these positions. The equation (3) is used for calculating the angle to throw a ball. The equation (4) is used for calculating the arm length. The robot calculates the trajectory of a ball from the arm angle, the arm speed and the arm length. In other words, the robot can calculate the trajectory based on a physical model of a robot by a mathematical approach using these parameters. After a ball was released from a grip, the robot takes a movie of the flying ball. The method of visualization is used for tracking the actual trajectory of the flying ball from images of the movie. The flying ball is recognized by the pattern recognition [8] from images of the movie.

Furthermore, In order to create the ideal trajectory, we will apply feedback control for a robot.

$$x = v_0 \times \cos\Theta \times t \tag{1}$$

$$y = v_0 \times \sin\Theta \times t - \frac{g}{2} \times t^2 \tag{2}$$

x : Distance of a ball

y : Height of a ball

v_0 : Initial velocity of a ball

Θ : The angle to throw a ball

g : Gravitation

t : Time

$$\Theta = ATAN \frac{(x_1 - x_2)}{(y_2 - y_1)} \tag{3}$$

$$L = k \times \sqrt{(x_1 - x_2)^2 + (y_2 - y_1)^2} \tag{4}$$

Θ : The angle to throw a ball

L : The arm length

Pixel position (x_1, y_1): Point subtracted the grip length from the position of the ball just released from the grip

Pixel position (x_2, y_2): Fulcrum point decided from the servomotor position.

k : Scale factor for converting to the actual scale.

B. Learning Algorithm

Various types of machine learning algorithms have been proposed. An artificial neural network is one of promising methods to realize machine learning with or without feedback error functions. Neural networks have been applied to nonlinear control [9]. The neural network realizes a nonlinear discriminated function.

However, the piecewise linear discriminant function can approximate the nonlinear discriminant function with arbitrary precision. Therefore, the neural network is proved to equivalent in the piecewise linear discriminant function [10].

If the computer has the enough memory capacity and we have the enough processing time, the learning vector quantization or the complete storage method is enough to be used practical. Therefore, we should choose an appropriate method depending on the problem.

The system consists of three kind means for the robot motions. The concept of the system is shown in Fig. 1. In this study, we experimented without parts of a dotted line.

In case of the means 1, the set-values are calculated from the target positions. The set-values are used for controlling the robot. The calculated data are treated as basic data for the feedback controlling. Target trajectory is calculated from the set-values.

In case of the means 2, parts positions of the system such as a ball position, a grip position and servomotor position are recognized by the pattern recognition. These recognitions are used to find the released position of the ball and the position of the fulcrum. Using these positions, we can calculate the thrown angle of a ball. And, these values are used to calculate the trajectory of the ball by using the equation of a parabola. The difference between the "calculated trajectory from the released position of the ball and the position of the fulcrum" and the "calculated trajectory from the set-values" corresponds with errors caused by internal factors. Because the errors should be corrected, the means has tables to relate with inputs and outputs. The output data from the relation table is used for the feedback controlling (5), (6). We explain the example of the feedback controlling as follows. Because it is a purpose to explain only the technique, the case of the angle control is omitted.

$$\Delta S = S_{Parts\_position} - S_{target} \tag{5}$$

$$y_{speed} = f(S_{target} + \Delta S) \tag{6}$$

ΔS : Error of the arm speed (Internal factors)

$S_{Parts\_position}$: Calculated arm speed from parts position

S_{Target} : Arm speed (Initial set-value)

y_{speed} : Correction data to actuator (Internal factors)

In case of the means 3, positions of the flying ball are recognized by the pattern recognition. These positions are memorized to the memory device. By this way, the actual trajectory of the ball is extracted from the images of the movie. The difference between the "actual trajectory of a ball from the image by the video camera" and the "calculated trajectory from the released position of the ball and the position of the fulcrum" corresponds with errors caused by external factors. Because the "height of the vertex" and the “distance to landing“ relates to the “arm angle to throw“ and the “arm speed“, these data is used to estimate by the fuzzy logic. The input data of the fuzzy reasoning part are the errors of the height of the vertex and distance to the landing. And, the output data of the fuzzy reasoning part is the arm angle to throw and the arm speed. The fuzzy reasoning uses the membership function. Using an equation (7), (8) and (9), we explain the example of the fuzzy controlling. Because it is a purpose to explain only the technique, the case of the angle control is omitted.

$$y_{\Delta MH} = y_{MHA} - y_{MHP} \tag{7}$$

$$x_{\Delta X} = x_{XA} - x_{XP} \tag{8}$$

$y_{\Delta MH}$: Error of the height of the vertex (External factors)

y_{MHA} : Height of the vertex of actual trajectory

y_{MHP} : Calculated vertex height

$x_{\Delta X}$: Error of distance to the landing (External factors)

x_{XA} : Actual distance to the landing

x_{XP} : Calculated distance to the landing

Fuzzy rule is shown as follows.

$if\ \ y_{\Delta MH}\ \ is\ Under\ and\ x_{\Delta X}\ \ is\ Under\ then\ F\ is\ SJ$

$if\ \ y_{\Delta MH}\ \ is\ Under\ and\ x_{\Delta X}\ \ is\ Just\ then\ F\ is\ SJ$

$if\ \ y_{\Delta MH}\ \ is\ Under\ and\ x_{\Delta X}\ \ is\ Over\ then\ F\ is\ JJ$

$if\ \ y_{\Delta MH}\ \ is\ Just\ and\ x_{\Delta X}\ \ is\ Under\ then\ F\ is\ SJ$

$if\ \ y_{\Delta MH}\ \ is\ Just\ and\ x_{\Delta X}\ \ is\ Just\ then\ F\ is\ JJ$

$if\ \ y_{\Delta MH}\ \ is\ Just\ and\ x_{\Delta X}\ \ is\ Over\ then\ F\ is\ JO$

$if\ \ y_{\Delta MH}\ \ is\ Over\ and\ x_{\Delta X}\ \ is\ Under\ then\ F\ is\ JJ$

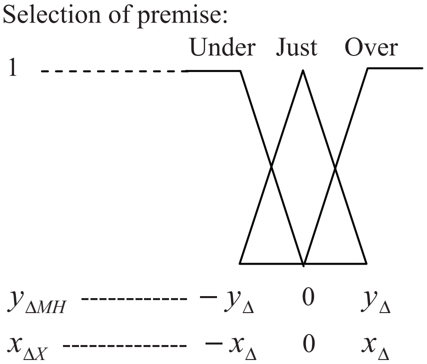
$if\ \ y_{\Delta MH}\ \ is\ Over\ and\ x_{\Delta X}\ \ is\ Just\ then\ F\ is\ JO$

$if\ \ y_{\Delta MH}\ \ is\ Over\ and\ x_{\Delta X}\ \ is\ Over\ then\ F\ is\ JO$

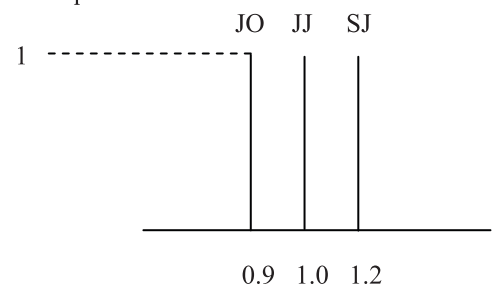
F : Fuzzy reasoning

Membership function is shown as follows.

Selection of premise:



Consequence:



Deffuzify is shown as follows.

$$Z_o = \frac{JO \times h_1 + JJ \times h_2 + SJ \times h_3}{h_1 + h_2 + h_3}$$

h_1, h_2, h_3 : Grade

Z_o : Fussy output (External factors)

The output data is used for the feedback controlling. The arbitrator calculates equation (9). And, the output data is used for the feedback controlling.

$$y_{actuator} = y_{speed} \times Z_o \tag{9}$$

$y_{Actuator}$: Output-data to actuator

y_{speed} : Correction data to actuator (Internal factors)

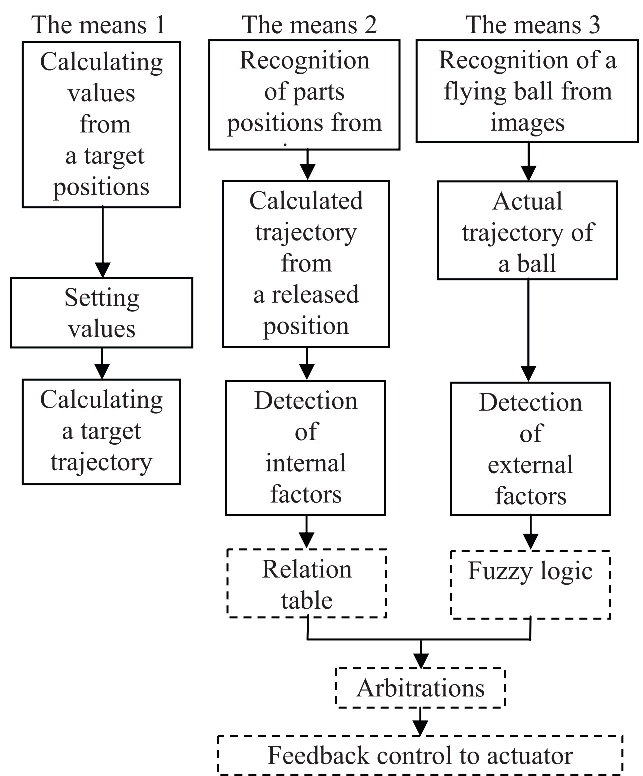


Fig. 1. The concept of the system

C. *Pattern Recognition Algorithm*

We apply AdaBoost (adaptive boosting) algorism to enhance the learning capability. AdaBoost algorithm [11][12] detects the positions of the released position of the ball and the position of the fulcrum.

Given example images

$(x_1, y_1), \ldots, (x_n, y_n)$

Where

$y_i = 0,1$ for negative and positive examples respectively.

Initialize weight

$w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0,1$ respectively,

Where m and l are the number of negatives and positives respectively.

For $t = 1, \ldots, T$:

1. Normalize the weights,

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$

So that w_t is a probability distribution.

2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to

$$w_t, \varepsilon_j = \sum_i w_i \mid h_j(x_i) - y_i \mid .$$

3. Choose the classifier, h_t, with the lowest error ε_t.

4. Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

Where $e_i = 0$ if example x_i is classified correctly,

$e_i = 1$ otherwise, and $\beta_t = \frac{\varepsilon_t}{1 - \varepsilon_t}$.

The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & Otherwise \end{cases}$$

Where $\alpha_t = \log \frac{1}{\beta_t}$

III. EXPERIMENT

We used a pitching robot for this experiment. The servomotor and the controller control the arm angle and the speed of the arm. After we calculated the set-values from the target points, we set the set-values to an actuator as arm angle to throw and the arm speed. And, we calculated the target trajectory from the set-values.

We developed a visualization method in order to discuss the difference of the trajectories between the theoretical approach and experimental results. The arm, the ball and the grip are recorded with camera for the pattern recognition. We developed the means to calculate a parabola by recognized parts positions from images of the movie. Furthermore, we developed the means to recognize the position of the flying ball from the movie image. The recognized result was used for evaluating whether a ball can reach to the target position.

A. The Hardware of the Pitching Robot

We developed the pitching robot with one DOF. It is made of some parts of the commercial robot kit named Bioloid. We show specifications of Bioloid in table 1. We prepared the software to operate an actuator via UART from the PC. Fig.2 is the photo of a servomotor that we used for an arm drive.

Table 1. Specifications of Bioloid kit

	Size	370mm
Body	Weight	2.2kg
	Voltage	12V5A
	Battery	Ni-MH 9.6V
	Type	AX-12
	Rotate angle	300 deg.
Servomotor	Torque	16.5kg.cm
	Speed	0.196sec/60°
	Weight	55g
	Signal	TTL
Software	Feature	Leaning

Fig. 2. Photo of a servomotor

The pitching robot has a transparent plastic arm with a rigid grip [13]. And, a servomotor put on a fulcrum to fix it. We made the arm of the hollow plastic. Its inertia moment can be small. But the hardness is not enough. The grip is made of the paper cup that a ball is smoothly released considerably. Because we want to find errors, we use these parts as is. We show this pitching robot in Fig. 3.

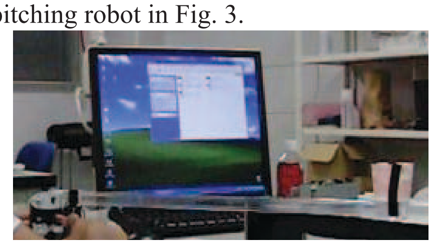


Fig. 3. The pitching robot

B. Image Recognition System based on AdaBoost

At first, we made 4,000 pieces of correct patterns by affine transform from the image. We got feature vectors from correct images for learning. Next, we prepared 19 pieces of non-correct images for learning. And, we performed the object learning. Looking for the position of the pattern, we input the movie image from a camera and processed the pattern recognition.

IV. EXPERIMENTAL RESULTS

A. Recognition Result of a Released Position

We performed the experiment to throw a ball by the robot. We set the arm angle to 60 degrees. And, we set the time of a round to 1 second.

But, the robot seemed to wave the arm too much in spite of the setting of the stop at 60 degrees.

We recorded the image of the arm motion and the thrown out ball with a camera. And, we saved it on a computer as an AVI file. We extracted the correct image of the ball just released from the grip. We show this correct image in Fig. 4. We created the learning pattern from it. And, we got feature vectors from these patterns. We classified the pattern in AVI file. We show this result in Fig. 5. (The right -side photo shows the robot, before the ball is released from the grip. The mid-position photo shows the ball just released from the grip. The left-side photo shows the robot, after the ball was released from the grip.) The rectangle of a red line is drawn around a recognized pattern. The pitching robot recognized only an image of the ball just released from the grip. It was correct recognition.

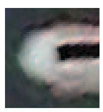


Fig. 4. Pattern of the ball just released from the grip

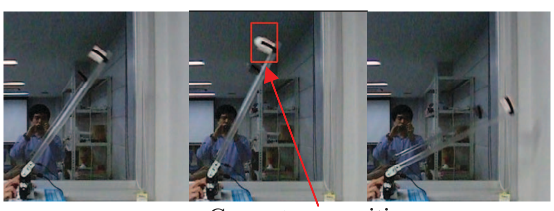


Fig. 5. Recognition of the ball just released from the grip

B. Recognition Result of a Fulcrum (a Servomotor)

We extracted the correct image of the servomotor corresponding to a fulcrum. We show this correct image in Fig. 6. We created learning pattern from it. And, we got feature vectors from these patterns. We classified the pattern in AVI file. We show this result in Fig. 7. The result was correct recognition.

The arm angle and length are calculated from the released position and the fulcrum position. In order to calculate the real scale, the actual arm length is compared with that of movie. We converted the image scale by the real scale. By this way, we could calculate the arm angle that the ball was just released from the grip. The equation (3) is used for calculating the thrown angle of a ball. The equation (4) is used for calculating the arm length.

We transmitted these parameters to Excel program with a CSV file. Excel program calculate a parabola from the “thrown angle of a ball“, the “length of an arm“, and the “time of a round“. The equation (1) and (2) are used in the calculations.

Fig. 6. Pattern of a servomotor

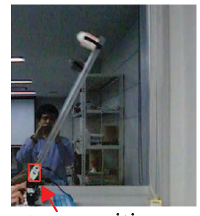


Fig. 7. Recognition of a servomotor

C. Recognition Result of a Ball (Actual Trajectory)

We extracted the correct image of a ball. We show this correct image in Fig. 8. We created learning pattern from it. And, we got feature vectors from these patterns. We classified the pattern in AVI file. We show this result in Fig. 9. As a recognition result, the correct recognition of the ball was performed. Furthermore, the pitching robot did not recognize the ball before the ball is released from the grip.

Fig. 8. Pattern of a ball

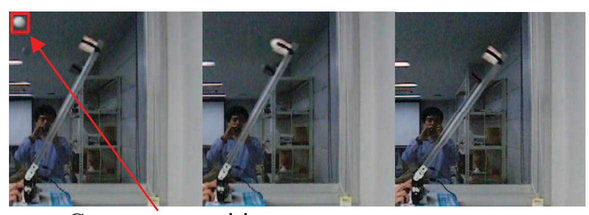


Fig. 9. Recognition of the ball

D. Results of Traced Trajectories

We set the “arm angle to throw the ball” and the “arm speed”. The set-values are decided by the calculation from the target positions. And, we make a graph of the “calculated parabolas“ and the “actual trajectory of the ball“.

This graph is shown in Fig. 10. In this graph,

○ marks are calculated data from the set-values.

△ marks are calculated data from parts position (the position of the ball just released from the grip).

× marks are data of the actual ball positions after flying.

As the results, we could find the difference among the ”calculated trajectory from the set-values”, the “calculated trajectory from the released position of the ball and the position of the fulcrum” and the “trajectory of the ball from images by the video camera". We investigated the difference.

(1) An arm angle to throw a ball was set at 60 degrees, but the angle went more over to 62 degrees in the recognition with the camera. The servomotor has 10 bits resolution. It is enough to resolve the one-degree. But, the inertia moment caused the over-turn of arm. This error was caused by internal factors. If the angle to throw is made the same, the trajectory of ○ marks becomes same as the trajectory of △ marks.

(2) If the influence of the air resistance is added to parabola equations (1) and (2), the trajectory of △ marks becomes nearly the trajectory of × marks. This error was caused by external factors. We can estimate the air resistance influence, using following equations (10) and (11).

$$k = \frac{1}{2} \times \rho \times \frac{\pi}{4} \times d^2 \times C_D \tag{10}$$

k : Air resistance coefficient

ρ : Density of the air

d : Diameter of the ball

C_D : Drag coefficient

The C_D can be referred from correspondence list with the Re.

Here,

$$\mathrm{Re} = \rho \times V \times \frac{d}{\mu} \tag{11}$$

Re : Reynolds number

ρ : Density of the air

V : Velocity

d : Diameter of the ball

μ : Viscosity coefficient of the air

As a result of calculation, air resistance coefficient: k is 0.000409. The external error is probably caused by the influence of the air resistance. Even if the other external errors were included a little, it is difficult to analyze all of errors. In such the case, the proposed fuzzy controller will work to correct the errors. It will be just like a human adjusting the arm motion by the sense.

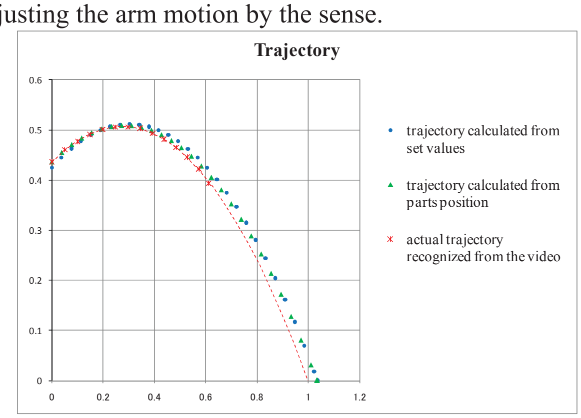


Fig. 10. Calculated trajectories and Actual trajectory

V. CONCLUSION

We performed the experiment of trajectories tracing for a robot based on human recognition. We could find the difference among the calculated trajectories and the actual trajectory by using the robot. We were able to demonstrate a series of the human-like recognition.

But, the misrecognition occurs yet. Therefore, the recognition should be improved in the system. If the system uses nonlinear discriminant function, the recognition may be improved. However, more processing time is necessary.

In future theme, the robot will memorize the relations with the trajectories, the arm angle to throw a ball and the servomotor speed. The robot will throw a ball to target position by the self-feedback control.

ACKNOWLEDGMENT

The authors would like to thank Prof. S. Oohara, S. Ohki, S. Nishio, N.Ohsawa, S. Yoneki and all the participants for their support in the development of the project. The JST is also appreciated. We performed this study as a part of the 2009 SSP (Science Partnership Project) The SSP was supported by JST (Japan Science and technology Agency).

REFERENCES

[1] Edited by A.Blake ans A.Yuille, "Active Vision", Massachusetts Institute of Technology, 1992.
[2] T.Nakamura and M.Asada, "Motion Sketch. Acquisition of Visual Motion Guided Behaviors", In Proc. Of IJCA-95, pp.126-132, 1995.
[3] L. A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes", IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-3, No.1, Jan. 1997.
[4] K.Hirota, Y.Arai and S.Hachisu, "Robot-Arm-System Playing 2 Dimensional Ping-Pong Game using Image Processing Technique", Research Reports, Department of Engineering, Hosei Univ. Mar. 1988.
[5] J.H.Connel and S.Mahadevan, editors. "Robot Learning", Kluwer Academic Publishers, 1993.
[6] M.Asada, S.Noda and K.Hosoda, "Action-Based State Space Construction for Robot Learning", Proc. of IEEE/RSJ Int. Conf. On Intelligent Robots and System 1996 (IROS'96), Vol.3, pp. 1502-1509, 1996.
[7] P.Whaite and F.P.Ferrie, "Autonomous exploration. Driven by uncertainty. ", IEEE Trans. Pattern Analysis and Machine Intelligence, 19, 3, pp.193-205, 1997.
[8] C.H.Chen, L.F.Pau, and P.SP.Wang, "Handbook of Pattern Recognition & Computer Vision", World Scientific Publication Co. Pte. Ltdtable 1993
[9] M.Satomi, H.Masuta, and N.Kubota, "Hierarchical Growing Neural Gas for Information Structure Space", IEEE Symposium Series on Computational Intelligence 2009, 2009.
[10] N. J. Nilsson, "Learning Machines", McGraw-Hill, 1965.
[11] Paul Viola and Michael J.Jones. "Rapid Object Detection using a Boosted Cascade of Simple Features", IEEE CVPR, 2001.
[12] Rainer Lienhart and Jochen Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection", IEEE ICIP Vol. pp.900-903, Sep. 2002.
[13] Sato, Asaji, "Theory about the energy-saving orbit of the ma-nipulator", A doctoral dissertation of Miyazaki University graduate school engineering graduate school, Mar.2009.