IC-MED

Vol. 3, No. 2, Page 119-129

Fuzzy-Based Evolutionary Robot Vision for People Tracking

Akihiro Yorita*, Naoyuki Kubota
Tokyo Metropolitan University, Graduate School of System Design, Tokyo, Japan.

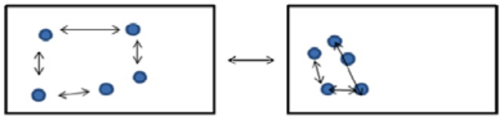
Received Date: April 20, 2009; Accepted Date: June 12, 2009

Abstract
This paper deals with evolutionary robot vision based on a genetic algorithm and fuzzy evaluation in order to realize people tracking. Active robot vision is an important research topic, and we must improve the performance of the visual perception. First, we discuss the concept of evolutionary robot vision in dynamic environments. Next, we apply growing neural gas for preprocessing as a bottom-up processing, and a local genetic algorithm based on clustering for template matching in human face recognition as a top-down processing. Furthermore, in order to improve the performance of the human face detection, we use fuzzy evaluation for evaluating the degree of human face. Finally, we show several experimental results of the proposed method and discuss the effectiveness of the proposed method.

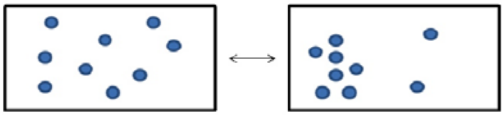
Keywords
People Tracking, Evolutionary Computation, Robot Vision, Fuzzy Theory, Partner Robots

have discussed the applicability of the evolutionary computation in robot vision [26].

Evolutionary computation (EC) is a field of simulating evolution on a computer. Evolutionary optimization methods are fundamentally iterative generation and alternation processes of candidate solutions. The optimization is done by the multi-point search operating on a set of individuals, which is called a population.



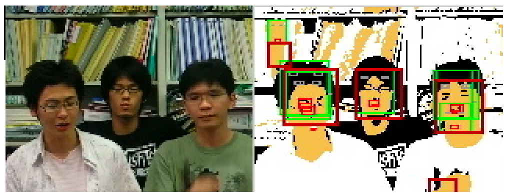
(a) Movement of attentive points in the visual search



(b) Distribution of candidate solutions in evolutionary search

Figure 1. Comparison of visual search and evolutionary search

Figure 1 shows the comparison between the visual search and evolutionary search. The left and right figures show the distribution of search points in the distributed search and the focused search, respectively. The visual search controls the searching area based on the fast movement of attentive point. The region of interest (ROI) for the information extraction is deeply related with the search of geometrical features included in the visual target. The search results are reflected to the next visual search. On the other hand, the evolutionary search controls the searching area based on the selection pressure to candidate solutions. If the selection pressure is high, the candidate solutions are centralized toward the better candidate solutions. Otherwise, candidate solutions are globally distributed in the search space. The next search points in the evolutionary search are generated by crossover and mutation. The search in ROI is mainly performed by mutation and local search, while a new search point for ROI is generated by crossover. The degree of interest is calculated by the fitness value. If the fitness value is high, the focused search should be performed. We applied a steady-state genetic algorithm (SSGA) to realize the continuous and real-time search for the robot vision like human visual perception in a dynamic environment [21]. Furthermore, we proposed a simple method of people tracking based on the combination of skin color and hair color [22], but we have a problem of misdetection of people by objects with similar color combination in the background image (Figure 2). Therefore, we propose a method for detecting a human face based on a local genetic algorithm with based on clustering (LGAC) and fuzzy evaluation in order to improve the performance of people tracking. The both of evolutionary computation and fuzzy theory are useful and practical in the search under the environment including noise [24,25].



Original Image **Human Tracking**

Figure 2. Pre-experiment of people tracking

The paper is organized as follow. In the Section 2, we explain a growing neural gas for color extraction and a local genetic algorithm based on clustering with fuzzy evaluation for people tracking. Section 3 shows several experimental results and discusses the effectiveness of the proposed method.

2. PEOPLE TRACKING BASED ON EVOLUTIONARY ROBOT VISION

Partner robots and Image Processing

We developed two types of partner robots; a mobile PC called MOBiMac and a human-like robot called Hubot in order to realize the social communication with a human [20,23] (Figure 3). Each robot has two CPUs and many sensors such as CCD camera, microphone, and ultrasonic sensors. Therefore, the robots can con-

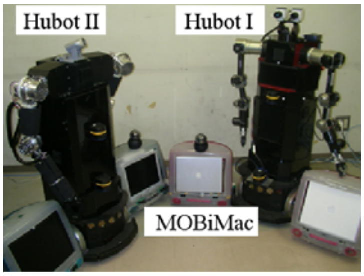


Figure 3. Partner robots; MOBiMac and Hubot

duct image processing, voice recognition, target tracing, collision avoidance, map building, imitative learning, and others.

In this paper, we focus on people tracking. The tracking problem of people or objects is significantly harder than that of a single person or object. The people tracking problem includes two problems of people detection problem as candidate detection in each image and a tracking problem of detected people as target recognition over time. In previous works, people tracking problems have been mainly solved by appearance-based methods. Kalman filter, particle filters, genetic algorithms, particle swarm optimization, and others [8-17]. Furthermore, dynamic model of human movement is also applied to improve the accuracy of people tracking. These methods try to detect the features of human appearance, and to trace them over time, but there are problems on variability of appearance features and computational cost in the real-time people tracking.

To realize visual perception for a robot, we should take recent works of psychology into account, especially, sensation, perception, and attention [2-7]. Visual perception is organized into a central object called *figure* and its blurred surroundings called *ground*. Our visual system operates in a flexible and adaptive manner to perceive the environment by using bottom-up and top-down processes. Bottom-up processing depends directly on external stimuli, while top-down processing is influenced by expectations, stored knowledge, context, and so on. The candidate detection is considered as the bottom-up processing based on the color distribution in segments of an image, while the target recognition is considered as top-down processing based on the similarity between the extracted target candidate and classification templates. The synthetic combination of bottom-up processing and top-down processing realizes the efficient and effective search. In this paper, we apply growing neural gas and LGAC for bottom-up processing and top-down processing, respectively.

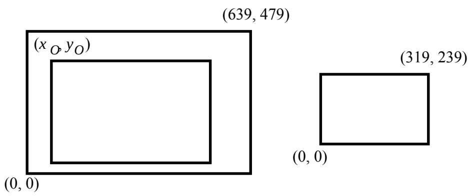
Preprocessing in Image Processing

Since the image processing takes much computational time and cost, the full size of image processing to every image is not practical. Therefore, we use the reduced size of image to detect a moving object for the fast human candidate detection.

First, an image of RGB color space is taken by a CCD camera equipped with the partner robot. Next, the robot calculates the center of gravity (COG) of the pixels different from the previous image as the differential extraction. The size of image used in the differential extraction is updated according to the previous result of human detection. Here the area generated by the differential extraction is called an attention range. If the robot does not move, the COG of the difference represents the location of the moving object. Therefore, the main search area for the human detection can be formed according to the COG in the attention range for the fast human detection. In this paper, the original size of an image is 640×480, and the size of this image is reduced into 320×240 as an attention range according to the reduction level ($1.0 \leq RL \leq 2.0$) and the origin (x_O, y_O) of the attention range (Figure4). If the reduction level is set at 1, the same resolution of the image is cut off from the original image. Otherwise, each pixel on the attention range is interpolated according to the four surrounding pixels based on the reduction level.

Growing Neural Gas for Color Extraction

An image is a set of pixels with color information. The segmentation into target candi-



(a) Original Image **(b) Attention range**

Figure 4. The generation of attention range

dates based on color distribution is very important to reduce the computational cost of direct template matching. Therefore, we apply an unsupervised clustering method for the segmentation based on color distribution in an image as bottom-up processing. Growing neural gas (GNG) is a competitive learning network as one of variants of SOM [14] used as a clustering method.

The learning algorithm of GNG is shown as follows.

Step 0. Generate two units at random position, w_{c1}, w_{c2} in $\mathbf{R}^n$ where w_i is the nth dimensional vector of a node ($w_i \in \mathbf{R}^n$). Initialize the connection set.

Step 1. Generate at random an input data v according to the probability $p(v)$ where v is composed of the position (x, y) and color information (R, G, B).

Step 2. Determine the nearest unit s_1 and the second-nearest unit s_2 (Figure 5 (a))

$$s_1 = \arg\min_{i \in A} \|v - w_i\|$$

$$s_2 = \arg\min_{i \in A \setminus \{s_1\}} \|v - w_i\| \tag{1}$$

Step 3. If a connection between s_1 and s_2 does not yet exist, create it. Set the age of the connection between s_1 and s_2 to zero.

$$c_{s_1, j} = 1, \quad a_{s_1, s_2} = 0 \tag{2}$$

Step 4. Add the squared distance between the input signal and the winner to a local error variable E_{s_1} (see Figure 5 (b)).

$$E_{s_1} \leftarrow E_{s_1} + \|v - w_{s_1}\|^2 \tag{3}$$

Step 5. Adapt the reference vectors of the winner and its direct topological neighbors by the learning rate η_1^G and η_2^G, respectively.

$$w_{s_1} \leftarrow w_{s_1} + \eta_1^G \cdot (v - w_{s_1}) \tag{4}$$

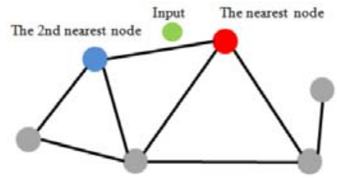
$$w_j \leftarrow w_j + \eta_2^G \cdot (v - w_j) \quad if\ c_{s_1, j} = 1$$

Step 6. Increment the age of all edges emanating from s_1.

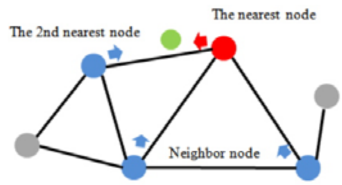
$$a_{s_1, j} \leftarrow a_{s_1, j} + 1 \quad if\ c_{s_1, j} = 1 \tag{5}$$

Step 7. Remove edges with the age larger than $a_{\max}$. If units have no more emanating edges after this, remove those units (see Figure 5(c)).

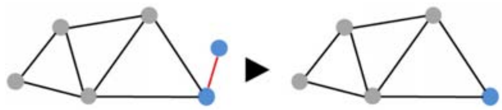
Step 8. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows (see Figure 5(d)).



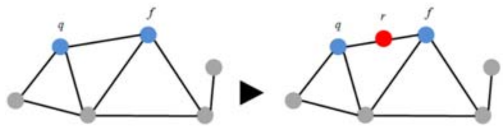
(a) Node selection



(b) Learning of nodes



(c) Edge deletion



(d) Node addition

Figure 5. How to learn GNG nodes and edge

Step 8-1. Determine the unit q with the maximum accumulated error.

$$q = \arg\max_{i \in A} E_i \tag{6}$$

where A is a set of nodes

Step 8-2. Determine the unit f with the maximum accumulated error among the neighbors of q.

$$f = \arg\max_{c \in N_q} E_c \tag{7}$$

where N_i is a set of nodes connected to the ith node

Step 8-3. Add a new unit r to the network and interpolate its reference vector from q and f.

$$w_r = 0.5 \cdot \left(w_q + w_f\right) \tag{8}$$

Step 8-4. Insert edges connecting the new unit r with units q and f, and remove the original edge between q and f.

Step 8-5. Decrease the error variables of q and f by the discount rate α.

$$E_q \leftarrow (1-\alpha)E_q \ , \ E_f \leftarrow (1-\alpha)E_f \tag{9}$$

Step 8-6. Interpolate the error variable of r from q and f

$$E_r = 0.5 \cdot \left(E_q + E_f\right) \tag{10}$$

Step 9 Decrease the error variables of all units

$$E_c \leftarrow (1-\beta)E_c \ , \ c \in A \tag{11}$$

Step 10 If a termination condition (e.g., some performance measure) is not yet fulfilled continue with step 2.

Figure 6 shows an example of the learning of GNG. In this way, the color distribution can be extracted from the image by using GNG.

Original Image **GNG for Human shape recognition**

Figure 6. An example of GNG

Local Genetic Algorithm based on Clustering for People Tracking

Generally, it is very difficult to realize both optimization and adaptation in a real world problem. In order to perform the optimization, it takes much computational time and cost, but the environmental condition might change much. Therefore, the real-time adaptation should be done, but all the population should trace the local minima as much as possible in real-time adaptation, because the current best solution is not guaranteed as the best solution in future. We propose a local genetic algorithm based on clustering (LGAC) as a distributed search method based on local hill- climbing of clustered individuals (Figure 7).

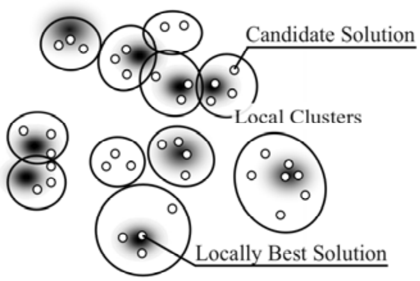


Figure 7. Search of LGAC in a Dynamic Environment; fitness landscape is depicted as monochrome gradation

Basically, each individual is composed of diploid, i.e., the self-best solution and candidate solution. If the fitness value of the candidate solution is larger than that of the self-best solution, the candidate solution is replaced with the self-best solution. In this way, each individual performs the elitist selection. We use the term of personal best or self-best that inspired from particle swarm optimization (PSO) invented by Eberhart and Kennedy [19]. Furthermore, we use the elitist crossover in the following updating rule;

$$\begin{aligned} g_{i,j} \leftarrow g_{i,j} &+ \alpha_1 r_1 (g^S_{i,j} - g_{i,j}) \\ &+ \alpha_2 r_2 (g^S_{L,j} - g_{i,j}) + \alpha_N N(0,1) \end{aligned} \tag{12}$$

where $g_{i,j}{}^S$ is the self-best solution; $g_{L,j}{}^S$ is the locally best solution in a cluster; r_1 and r_2, are uniform random value between 0 and 1.0; $N(0,1)$ is a normal random value with average of 0 and 1.0, and α_1, α_2, and α_N are coefficients. Furthermore, we can use adaptive mutation as follows;

$$\alpha_N = \beta_1 \cdot \frac{f_{\max} - f_i}{f_{\max} - f_{\min}} + \beta_2 \tag{13}$$

where $f_{\max}$ and $f_{\min}$ are the maximal value and minimal value of fitness values in a local cluster or the population; and β_1 and β_2 are coefficient and offset, respectively. In the adaptive mutation, the variance of the normal random number is relatively changed according to the fitness values of the population in case of maximization problems. The proposed method is similar to PSO, but in this paper, we explicitly use a mutation factor in order to trace local minima in a dynamic environment.

We use the *k*-means algorithm [18] as a clustering method. The *k*-means algorithm is one of the most popular iterative descent clustering methods. The number of clusters is *K*. When the reference vector of the *k*th cluster is represented by $r_k = (r_{k,1}, r_{k,2}, \dots, r_{k,2})$, the Euclidian distance between the *i*th input vector $u_i = (g_{i,1}, g_{i,2}, \dots, g_{i,m})$ and the *k*th reference vector is defined as

$$d_{i,k} = \|u_i - r_k\| \tag{14}$$

Next, the reference vector minimizing the distance $d_{i,k}$ is selected by

$$c_i = \arg\min_k \{\|u_i - r_k\|\} \tag{15}$$

where c_i is the cluster number which the *i*th input belongs to. After selecting the nearest reference vector to each input, the *k*th reference vector is updated by the average of the inputs belonging to the *k*th cluster. If the update is not performed at the clustering process, this updating process is finished.

The robot must recognize a human face from complex background speedily. The human face candidate positions based on the colors of facial landmarks are extracted by LGAC with template matching. Figure 8 shows a candidate solution of a template used for detecting a human face. A template is composed of numerical parameters of $g_{i,1}$, $g_{i,2}$, $g_{i,3}$, and $g_{i,4}$. The number of individuals is *G*. The initial population of LGAC for human detection at the discrete time step *t* is updated by using the reference vector of GNG in addition to the candidate solutions obtained at the previous time step *t*-1. The evaluation method of each candidate solution is described in the next subsection. The iteration of LGAC is repeated until the termination condition is satisfied.

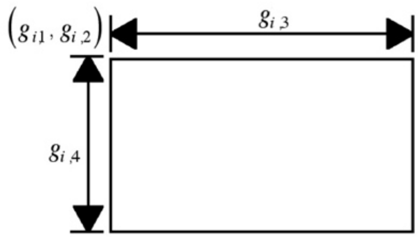


Figure 8. A template used for human detection in LGAC

The human tracking is performed according to the time series position of the *i*th human candidate ($g_{i,1}$, $g_{i,2}$) obtained by LGAC. The position of the *j*th human candidate in the human tracking ($X_{k,1}$, $X_{k,2}$) is updated by the nearest human candidate position within the tracking range. In addition, the width and height of the human candidate for the human tracking ($X_{k,3}$, $X_{k,4}$) are updated by the size of the detected human ($g_{i,3}$, $g_{i,4}$). The update is performed as follows (*j*=1,2,3,4);

$$X_{k,j} = (1-\lambda)X_{k,j} + \lambda \cdot g_{i,j} \tag{16}$$

Furthermore, the time counter for the reliability of human tracking is used. If the human candidate position in the human tracking is performed, the time counter is incremented. Otherwise, the time counter is decremented. If the time counter is larger than the threshold (*HT*), the human count is started. Sometimes, several human candidates are close each other, because several human candidates in a single human can be generated by the human detection. Therefore, the removal processing is performed when human candidates are coexisting within the tracking range.

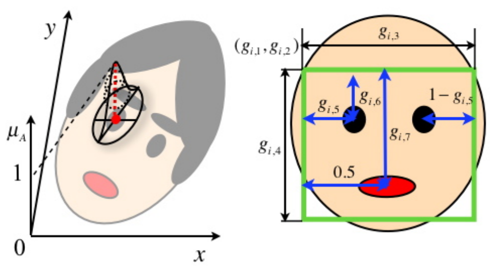
Fuzzy Evaluation for Human Face Detection

The human face candidate positions based on human skin and hair colors are extracted by LGAC with template matching. The fitness value is calculated by the following equation,

$$f_{Pixel} = C_{Skin} + C_{Hair} + \eta_1 \cdot C_{Skin} \cdot C_{Hair} - \eta_2 \cdot C_{Other} \tag{17}$$

where C_{Skin}, C_{Hair} and C_{Other} indicate the numbers of pixels of the colors corresponding to human skin, human hair, and other colors, respectively; η_1 and η_2 are the coefficients (η_1, η_2>0). The human detection based on color distribution sometime extracts some object with similar color distribution. In order to improve the performance of human face detection, we can use the color information of facial landmarks.

We apply fuzzy evaluation for face detection based on the position of facial landmarks (Figure 9 (a)). We use the position of eyes and mouth. We use



(a) Fuzzy Evaluation based on Gaussian membership **(b) The position of facial landmarks**

Figure 9. A template used for face recognition

$$\mu_{A(h,i,j)} = \begin{cases} \exp\left(-\dfrac{(x-a_{h,1})^2}{2b_{h,1}^2} - \dfrac{(y-a_{h,2})^2}{2b_{h,2}^2} \right) & if\ p(i,j) = c_h \\ 0 & otherwise \end{cases} \tag{18}$$

where (x,y) is the normalized position of the pixel (i,j) on the image; $p(i,j)$ is the color ID of a the pixel (i,j) on the image; c_h is the color ID of the hth facial landmark; $(a_{h,1},a_{h,2})$ is the center of a facial landmark; $(a_{h,1},a_{h,2})$ and $(b_{h,1},b_{h,2})$ are the normalized position and size of the hth facial landmark in the template candidate extracted by LGAC. Therefore, this value is high if the color pixel corresponding to the facial landmark is near with the center of the facial landmark. We can evaluate the degree of existing each facial landmark as follows;

$$f_{Land,h} = \sum_{(i,j)\in g_k} \mu_{A(h,i,j)} \tag{19}$$

where g_k is the template of the kth candidate solution in LGAC; h is facial the landmark ID. Furthermore, we can evaluate the degree of face as follows;

$$f_{Face} = \prod_{h=1}^{H} f_{Land,h} / p_{no,h} \tag{20}$$

where H is the number of facial landmarks; $p_{no,h}$ is the number of pixels of the hth facial landmark. We use right eye (h=1), left eye (h=2) and mouth (h=3) for the evaluation (H=3). Therefore, we use the following total evaluation function;

$$f_{Total} = f_{Pixel} + \eta_3 f_{Face} \tag{21}$$

where η_3 is a coefficient (η_3>0). As a result, this problem results in the maximization problem.

Figure 9 (b) shows the positions of facial landmarks where $(g_{i,5}, g_{i,6})$ is the position of the right eye and $(0.5, g_{i,7})$ is the position of the mouth. Therefore, $(a_{1,1}, a_{1,2}) = (g_{i,5}, g_{i,6})$, $(a_{2,1}, a_{2,2}) = (1-g_{i,5}, g_{i,6})$, and $(a_{3,1}, a_{3,2}) = (0.5, g_{i,7})$. However, the position of each facial landmark is peculiar to a person. Therefore, the position of membership function corresponding to each facial landmark should be updated according to the detected person. As a result, the total number of parameters for human detection by LGAC is 7.

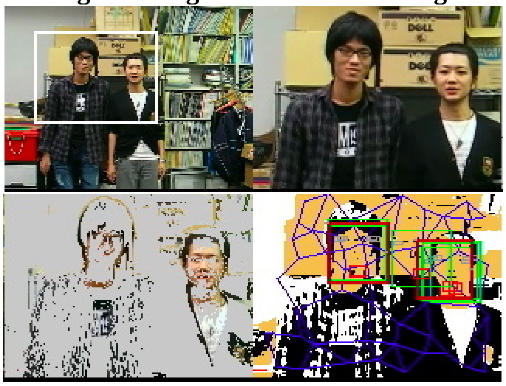
3. EXPERIMENTAL RESULTS

This section shows experimental results of people tracking of a partner robot. The maximal number of nodes in GNG is 50. The population size of LGAC is 50. The number of generations in each frame is 100 including the initial evaluations after the frame of image is updated. This value is relatively small comparing with that of the search by standard GA, but the search by LGAC is a time series of continuous search in a dynamic environment including a small change.

Figure 10 shows experimental results of the proposed method using attention range. The rightmost figure is the original image where the white box indicates the attention range generated according to the result of differential extraction. The next is the generated image based on the attention range. The next is the result of differential extraction. According to the center of gravity and the number of the pixels with temporary high difference, the center of attention range and the resolution of the attention range are updated sequentially. The rightmost figure indicates the result of the color extraction by GNG (blue line), the human face detection (green boxes), and people tracking (red boxes). Figure 11 shows the change of resolution of the attention range. These results show that the attention range is updated according to the motion of people, and the proposed method successfully performs people tracking.

Next, we conduct an experiment on the people tracking where the number of people is increasing. Figure 12 shows snapshots of the people tracking, and Figure 13 shows the number of the people tracked by the proposed method. At first, there is nobody in the image, but the number of people is increasing. The experimental results show that the proposed method can perform people tracking with allowable error in order that the robot can perform the communication with people.

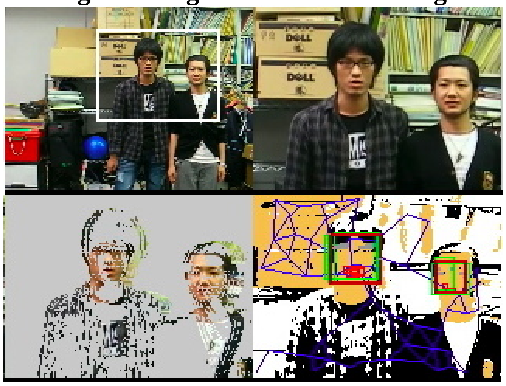
Original Image **Attention Range**



Differential Extraction **Human Tracking**

(a) Initial position (*t*=1)

Original Image **Attention Range**



Differential Extraction **Human Tracking**

(b) Position after moving toward the left (*t*=120)

Figure 10. Experimental results of people tracking

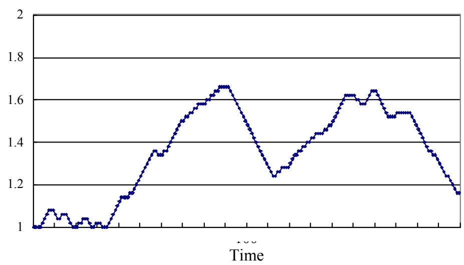


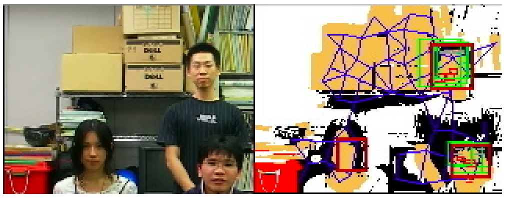
Figure 11. Attention range of people tracking

Original Image **Human Tracking**

(a) *t*=20

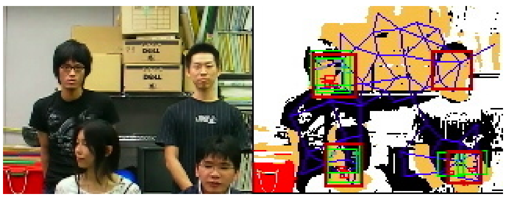
Original Image **Human Tracking**

(b) *t*=40



Original Image **Human Tracking**

(c) *t*=80



Original Image **Human Tracking**

(d) *t*=120

Figure 12. Experimental results of people tracking

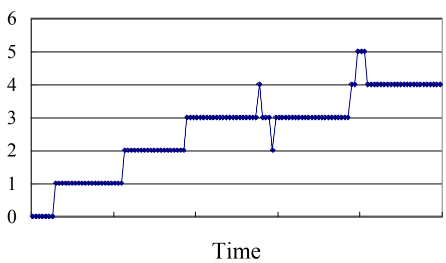


Figure 13. Experimental results of people tracking

4. SUMMARY

This paper proposed a method of people tracking based on the color extraction by growing neural gas, the template matching with a local genetic algorithm based on clustering (LGAC), and fuzzy evaluation for recognizing a human face. The experimental results show the effectiveness of the proposed method. The membership functions are very useful to evaluate candidate solutions including noisy data. The essence of the proposed method is in the flexibility of the search by combining local genetic algorithm and fuzzy evaluation.

As a future work, we will develop a method of human face detection in case of rotation of human face. Furthermore, we apply the proposed method to the associative learning between the perceptual information and symbolic information peculiar with the interacting person.

REFERENCES

[1] J. Cassell, Embodied Conversational Agents: Representation and Intelligence in User Interface, AI Magazine, Winter Vol. 22, No. 3, pp. 67-83, 2001.

[2] J.A. Anderson and E. Rosenfeld,. *Neurocomputing*, The MIT Press, Cambridge, Massachusetts, US, 1988.

[3] David Marr, Vision, W. H. Freeman, San Francisco, 1982.

[4] C.-H. Chen, L.-F. Pau, P.-S-P. Wang, *Handbook of Pattern Recognition & Computer Vision*, World Scientific Publication Co. Pte. Ltd., 1993.

[5] H.H. Bulthoff, S.W. Lee, T.A. Poggio, C. Wallraven, *Biologically Motivated Computer Vision*, Springer-Verlag, 2002.

[6] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Massachusetts: Addison Wesley Publishing Company Inc, 1989.

[7] J.H. Holland, Adaptation in Natural and Artificial Systems, First edition, Massachusetts, The MIT Press, 1992.

[8] D. Fogel, Evolutionary Computation, New York: IEEE Press, 1995.

[9] M. Lozano, F. Herrera, N. Krasnogor, D. Molina, Real-Coded Memetic Algorithms with Crossover Hill-Climbing, Evolutionary Computation, Vol. 12, No. 3, Pages 273-302, MIT Press, Cambridge, Massachusetts, US, 2004.

[10] J. Branke, Evolutionary Optimization in Dynamic Environments, Volume 3 of the Book Series on Genetic Algorithms and Evolutionary Computation, Kluwer Academic Pub Published, 2001.

[11] R. C. Eberhart, J. Kennedy, Y. Shi, Swarm Intelligence, Morgan Kaufmann Publisher, San Francisco, 2001.

[12] Indra A. Sulistijono and N. Kubota, Human Head Tracking Based on Particle Swarm Optimization and Genetic Algorithm, Journal of Advanced Computational Intelligence and intelligent Informatics, Vol. 11 No. 6, Jul. 2007, pp. 681-687, 2007.

[13] B. Fritzke. A growing neural gas network learns topologies. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, Advances in Neural Information Processing Systems 7, pages 625-632. MIT Press, Cambridge MA, 1995.

[14] M. Hynar, M. Burda, J. Sarmanova: Unsupervised clustering with growing self-organizing neural network -- a comparison with non-neural approach. DATESO pp. 58-68, 2005.

[15] Y. Wu, T. Yu, "A Field Model for Human Detection and Tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, No. 5, pp. 753-765, 2006.

[16] S.M. Khan, M. Shah, "Tracking Multiple Occluding People by Localizing on Multiple Scene Planes", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31, No. 3, pp. 505-519, 2009.

[17] M.W. Lee, R. Nevatia, Human Pose Tracking in Monocular Sequence Using Multilevel Structured Models, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31, No. 1, pp. 27-38, 2009.

[18] B. Fritzke. Growing cell structures - a self-organizing network for unsupervised and supervised learning. Technical Report ICSTR-93-026, International Computer Science Institute, Berkeley, May 93.

[19] J. Kennedy and R. Eberhart, "Particle Swarm Optimization" *Proc. IEEE Int. Conf. Neural Netw.*, Perth, Australia, pp. 1942-1945, 1995.

[20] N. Kubota, K. Nishida, Cooperative Perceptual Systems for Partner Robots Based on Sensor Network, International Journal of Computer Science and Network Security, Volume 6, Number 11, pp. 19-28, 2006.

[21] N. Kubota, Visual Perception and Reproduction for Imitative Learning of A Partner Robot, WSEAS Transaction on Signal Procesing, Issue 5, Vol. 2, pp. 726-731, 2006.

[22] N. Kubota, J. Pei, A. Yorita, H. Kojima, Human Tracking and Natural Communication of Partner Robots, Proc (CD-ROM) of International Symposium on Advanced Intelligent Systems (ISIS2007), pp. 205-210, September 5-8, 2007.

[23] N. Kubota, Computational Intelligence for Structured Learning of A Partner Robot Based on Imitation, *Information Science*, No. 171, pp.403-429, 2005.

[24] F. Herrera, F. Hoffmann, L. Magdalena, O. Cordon, Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases, World Scientific Pub Co Inc, 2001.

[25] W. Pedrycz, Fuzzy Evolutionary Computation, Kluwer Academic Pub, 1997.

[26] N. Kubota, T. Ohinata, S. Wakisaka, H. Liu, Extraction of Facial Landmarks for a Partner Robot Based on Evolutionary Computation, Proc. of Computation Intelligence, Robotics and Autonomous Systems (CIRAS 2007), pp. 44-49, 2007.

AUTHOR INFORMATION

Akihiro Yorita graduated from Saitama University in 2007. He received M.E. degree in System Design from Tokyo Metropolitan University in 2009. He has been a doctor course student of System Design at Tokyo Metropolitan University.

Naoyuki Kubota graduated from Osaka Kyoiku University in 1992, received the M.E. degree from Hokkaido University in 1994, and received the D.E. degree from Nagoya University, Japan in 1997. He joined Osaka Institute of Technology, in 1997. He joined the Department of Human and Artificial Intelligence Systems, Fukui University as an associate professor in 2000. He joined the Department of Mechanical Engineering, Tokyo Metropolitan University in 2004, and has been currently an associate professor of the Department of System Design, Tokyo Metropolitan University since 2005.