Multiple Human Face Detection Based on Local Genetic Algorithm Clustering Using Mobile iPhone Platform

Indra Adji Sulistijono
Dept. of Mechatronics Eng., Electronics Engineering Polytechnic Institute of Surabaya (EEPIS)
Kampus PENS, Jalan Raya ITS Sukolilo, Surabaya 60111, Indonesia
Tel: +62 (31) 594-7280 Ext. 4186; Fax: +62 (31) 594-6114; Email: indra@eepis-its.edu

Yuichiro Toda, Naoyuki Kubota
Graduate School of System Design, Tokyo Metropolitan University
6-6 Asahigaoka, Hino, Tokyo 191-0065, Japan
Tel&Fax:+81 (42) 585-8441; Email: kubota@tmu.ac.jp

Abstract

This paper discusses the role of evolutionary computation in visual perception for partner robots. The search of evolutionary computation has many analogies with human visual search. First of all, we discuss the analogies between the evolutionary search and human visual search. Next, we propose the concept of evolutionary robot vision, and multiple human face detection method based on local genetic algorithm clustering. Finally, we show experimental results of the multiple human face detection using mobile platform iPhone platform to discuss the effectiveness of our proposed method.

1. INTRODUCTION

Vision plays a significant role for the interaction of organisms, and therefore we need it to construct autonomous and intelligent robots that usually seek face recognition, expression understanding, motion detection, anticipations, etc.

Evolutionary computation (EC) is a field of simulating evolution on a computer [13]. 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. EC has been applied to optimization problems in dynamic environments, because the population can maintain the genetic diversity to adapt to the dynamic environment. Visual system can be discussed from the dynamic point of view since the visual image is changing over time. The visual systems including image processing can be divided into passive vision and active vision. In general, the passive vision is used for the focused information extraction toward a specific direction, while the active vision is used for the information extraction by updating the sensing direction and range in the vast area where the range of visual scene is restricted. In fact, the active vision is one of the main methods for the information extraction of an intelligent robot in order to extract perceptual information from the environment. To realize a visual perception system for a robot, we should take recent works of psychology into account, especially, sensation, perception, and attention.

We apply a local genetic algorithm based on clustering (LGAC) to detect for the robot vision like human visual perception.

This paper is organized as follows. Section 2 discusses the analogy between human visual search and evolutionary search. Section 3 and 4 explain the detail of the local genetic algorithm based on clustering and multiple human face detection for partner robots and iPhone operating system. Section 5 shows preliminary simulation results of the proposed method in a dynamic environment, and experimental results of multiple human face detection.

2. MULTIPLE HUMAN FACE DETECTION

2.1. Image Processing

Various types of intelligent methods [11-23] have been applied for image processing in real world applications thanks to the development of cheap and small digital cameras and signal processing boards with low energy consumption. Especially, information extraction by image processing plays the essential role in monitoring systems, surveillance systems, and automatic control.

Various technologies for image processing are required for realizing the robot vision, *e.g.*, color processing, target detection, template matching, shape recognition, motion extraction, and optical flow. We have applied spiking neural networks, cellular neural networks, self-organizing map, and others for human detection, motion extraction, and shape recognition [24-30]. In this paper, we focus on multiple human face detection. The detection problem of multiple human face or objects is significantly harder than that of a single person or object. The multiple human face detection problem includes two problems of multiple human face detection problem in each image and a detection problem of detected multiple human face over time. In previous works, multiple human face detection problems have been mainly solved by appearance-based methods. Kalman filter, particle filters, genetic algorithms, particle swarm optimization, and others [11-22] have been applied in appearance-based methods. Furthermore, dynamic model of human movement is also applied to improve the accuracy of multiple human face detection. 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 multiple human face detection. Therefore, we use only colors corresponding to human face and hair, and propose a local genetic algorithm based on clustering to realize the fast coarse search for multiple human face detection.

2.2 Local Genetic Algorithm in Dynamic Environment

The previous section discussed the features of EC in dynamic environment. Next, we discuss a method for the optimization and adaptation in a dynamic environment. 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 (Fig. 3).

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 [14,15]. Furthermore, we use the elitist crossover in the following updating rule;

$$x_{ij} \leftarrow x_{ij} + \alpha_1 r_1 (x^S_{i,j} - x_{i,j}) + \alpha_2 r_2 (x^S_{L,j} - x_{i,j}) + \alpha_N N(0,1) \quad (1)$$

where $x^S_{i,j}$ is the self-best solution; $x^S_{L,j}$ 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;

$$\alpha_N = \beta_1 \cdot \frac{f_i - f_{\min}}{f_{\max} - f_{\min}} + \beta_2 \quad (2)$$

where $f_{\max}$ and $f_{\min}$ are the maximal value and minimal value of fitness values in a local cluster or the population; β_1 and β_2, are coefficient and offset, respectively. 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 [23] as a clustering method. The *k*-means algorithm is one of the most popular iterative descent clustering methods. The inputs to *K*-means algorithm are $(x_{i,1}, x_{i,2}, \ldots, x_{i,m})$ of the *i*th candidate solution. The number of clusters is *K*. When the reference vector of the *k*th cluster is represented by $r_k = (r_{k,1}, r_{k,2}, \ldots, r_{k,2})$, the Euclidian distance between the *i*th input vector $u_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,m})$ and the *k*th reference vector is defined as

$$d_{i,k} = \| u_i, r_k \| \quad (3)$$

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

$$c_i = \arg\min_k \{ \| u_i, r_k \| \} \quad (4)$$

where c_i is the cluster number which the *i*th input belongs to.

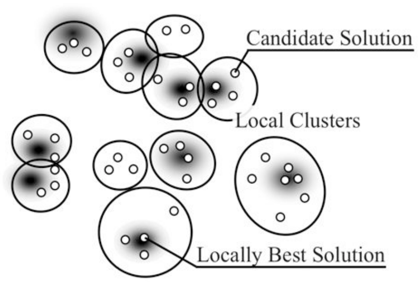

Fig. 3. Search of LGAC in a dynamic environment; fitness landscape is depicted as monochrome gradation

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. Figure 4 shows an example of the search of the proposed method in a dynamic environment using Gaussian functions. As shown in Fig.

3, the number of local clusters should be larger than that of modalities in the dynamic environment. However, the change of fitness landscape is unknown beforehand. Therefore, we prepare the sufficient number of clusters. In this way, the genetic search is mainly conducted within a local cluster.

2.3. Human Detection

Human detection is one of the most important tasks for partner robots. Pattern matching has been performed by various methods such as template matching, cellular neural network, neocognitron, and dynamic programming (DP) matching. In general, pattern matching is composed of two steps of target detection and target recognition. The aim of target detection is to extract a target from an image, and the aim of the target recognition is to identify the target from classification candidates. 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 detection.

First, 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 human detection result. The differential extraction calculates the difference of the number of pixels based on color difference between the previous and current images. 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 for the fast human detection.

The area corresponding to human skin and hair colors is extracted by the proposed method, LGAC, based on template matching. Figure 4 shows a candidate solution of a template used for detecting a target. 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. Fitness value is calculated by the following equation,

$$F_i = C_{Skin} + C_{Hair} + \eta_1 \cdot C_{Skin} \cdot C_{Hair} - \eta_2 \cdot C_{Other} \qquad (5)$$

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 $(\eta_1, \eta_2 > 0)$.

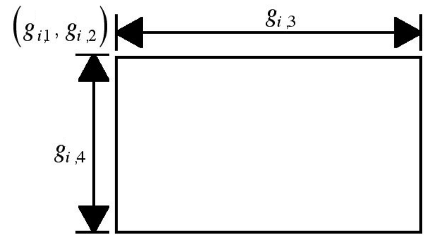

Fig. 4. A candidate solution of a template used for human detection

3. IPHONE OPERATING SYSTEM (IOS)

3.1 iPhone OS Event-Handling System

Since introducing in January 2007 on Apple Event: Macworld at San Fransisco, iPhone become one of smartphone that people want to try to develope the application on iPhone. One of interesting application is based on camera on iPhone. In this first research on iPhone camera, we use camera for image capturing.

The iPhone OS event-handling system is responsible for detection touch and motion events and delivering them to your application. All events are delivered to the application through the UIApplication object, which manages the queue of incoming events and distributes them to other parts of the application. For most applications, touches are the most significant type of event you can receive. Other types of events may also be generated and delivered, but touches reflect direct interactions with your application's views.

When it launches an application, the system creates both a process and a single thread for the application. This initial thread becomes the application's main thread. This is where the UIApplication object sets up the main run loop and configures its event-handling code, as shown in Figure 5. As touch events come into the application, they are queued until the application is ready to process them. The application processes events in the main run loop to ensure that they are handled sequentially as they arrive. The actual handling of a given event usually occurs in other objects, such as your application's views and view controllers.

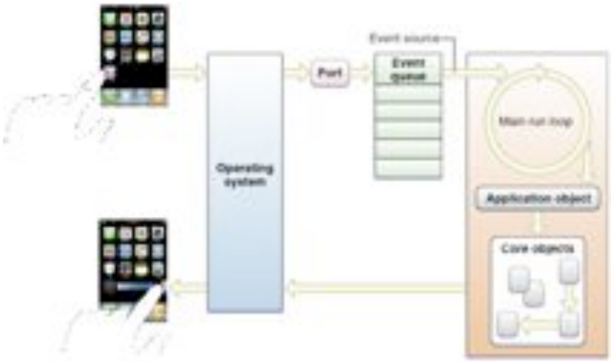

Fig. 5. Processing events in the main run loop (source: *iPhone Application Programming Guide Book*, Apple Developer, 2010-06-14, pp.41)

A run loop monitors sources of input for a given thread of execution. The application's event queue represents one of these input sources. When an event arrives, the run loop wakes up the thread and dispatches control to the handler for the event queue, which in this case is the UIApplication object. When the handler finishes, control passes back to the run loop, which then processes another event, processes other input sources, or puts the thread to sleep if there is nothing more to do.

3.2 View Coordinates in iPhone

Coordinates in UIKit are based on a coordinate system whose origin is in the top-left corner and whose coordinate axes extend down and to the right from that point. Coordinate values are represented using floating-point numbers, which allow for precise layout and positioning of content and allow for resolution independence. Figure 5 shows this coordinate system relative to the screen, but this coordinate system is also used by the UIWindow and UIView classes. This particular orientation was chosen to make it easier to lay out controls and content in user interfaces, even though it differs from the default coordinate systems in use by Quartz and Mac OS X.

Fig. 6. A candidate solution of a template used for human detection
(source: *View Programming Guide for iOS*, Apple Developer, 2010-06-14, pp.18)

Every window and view object maintains its own local coordinate system. All drawing in a view occurs relative to the view's local coordinate system. The frame rectangle for each view, however, is specified using the coordinate system of its parent view, and coordinates delivered as part of an event object are specified relative to the coordinate system of the enclosing window. For convenience, the UIWindow and UIView classes each provide methods to convert back and forth between the coordinate systems of different objects.

4. EXPERIMENTAL RESULTS

This section shows experimental results of multiple human face detection of a partner robot by the proposed method where the maximal number of humans is 4. We used three kinds of images as shown in Figure 7.

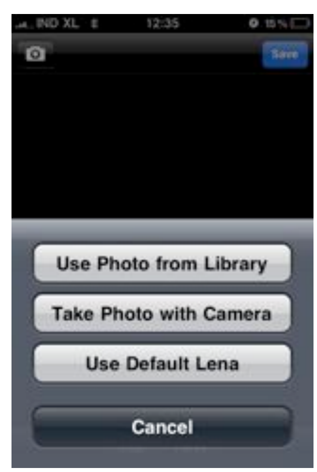

Fig. 7. A snapshot of iPhone screen input of still images: photo from library, photo with camera and Lena default

Figure 8 shows multiple human face detection experimental results on iPhone human detection results by blue boxes corresponding to the best solutions in the local clusters. Such a misdetection sometimes occurs, but we can improve this problem by using human facial landmark extraction in our previous research [26]. However, since the aim of the partner robots is not to track all multiple human face in the human detection, but to track the near multiple human face for the communication with them. Of course, the precise multiple human face detection is also very important, but the partner robots should perform various functions such as voice recognition and gesture recognition simultaneously. Therefore, the partner robots should perform the human detection with less computational time and cost.

5. CONCLUSIONS

In this paper, we proposed a local genetic algorithm based on clustering in order to improve the human detection performance. The proposed method sufficiently performs the detection of several human face. However, the proposed method sometimes misdetects objects similar to skin and hair colors. In order to improve the performance of the human detection, we will incorporate an extraction method of the human facial landmarks into the human detection method [30].

As another future work, we intend to develope on-line human face detection on iPhone platform for facial expression recognition and focus to .

References

[1] W. James, "*Principles of Psychology*", Holt, NewYork, 1890.

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

[3] A. Mori, S. Uchida, R. Kurazume, R. Taniguchi, T. Hasegawa, and H. Sakoe, "Early Recognition and Prediction of Gestures", *Proc. Int. Conf. on Pattern Recognition*, Hong Kong, pp. 560-563, 2006.

[4] R.C.Gonzalez and R.E.Woods, *Digital Image Processing*, Addison Wesley, 1992.

[5] D.A.Forsyth and J.Ponce, *Computer Vision*, A Modern Approach, Prentice Hall, 2003.

[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 MIT Press Ed., Massachusetts: The MIT Press, 1992.

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

[9] R. C. Eberhart, J. Kennedy, and Y. Shi, "*Swarm Intelligence*", San Francisco: Morgan Kaufmann Publ., 2001.

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

[11] G. G. Syswerda, "A Study of Reproduction in Generational and Steady-state Genetic Algorithms", *Genetic Algorithms*, San Mateo: Morgan Kaufmann Publ. Inc., 1991

[12] S. Watanabe, T. Hiroyasu and M. Miki, "LCGA: Local Cultivation Genetic Algorithm for Multi-Objective Optimization Problems", *Proc. Genetic and Evolutionary Comp. Conf.,* pp. 762-763, 2002.

[13] E. Bonabeau, M. Dorigo and G. Theraulaz, "*Swarm Intelligence: From Natural to Artificial Systems*", New York: Oxford University Press, 1999.

[14] M. Dorigo and T. Stutzle, "*Ant Colony Optimization*", Bradford Books, The MIT Press, 2004.

[15] J. Branke, "Evolutionary Optimization in Dynamic Environments", *Vol. 3 of the Book Series on Genetic Algorithms and Evolutionary Computation*, Kluwer Academic Publ. 2001.

[16] J. Huidong, L. K. Sak, and W. M. Leung, "Genetic-guided Model-based Clustering Algorithms" *Proc. 2001 Int. Conf. on Art. Intelligence*, vol. 2, pp. 653-659, 2001.

[17] M. Lozano, F. Herrera, N. Krasnogor, and D. Molina. "Real-Coded Memetic Algorithms with Crossover Hill-Climbing", *Evolutionary Comp.*, The MIT Press, Cambridge, Massachusetts, USA, vol. 12, no. 3, pp. 273-302, 2004.

[18] T. Hastie, R. Tibshirani, J. Friedman, "*The Elements of Statistical Learning: Data Mining, Inference, and Prediction*", New York: Springer-Verlag, 2001.

[19] N. Kubota and K. Nishida, "The Role of Spiking Neurons for Visual Perception of A Partner Robot", *Proc. 2006 IEEE World Cong. Comp. Intelligence*, pp. 530-537, 2006.

[20] N. Kubota and K. Nishida, "Cooperative Perceptual Systems for Partner Robots Based on Sensor Network", *Int. J. Comp. Sci. and Netw. Security,* Vol. 6, No. 11, pp. 19-28, 2006.

[21] N. Kubota, "Visual Perception and Reproduction for Imitative Learning of A Partner Robot", *WSEAS Trans. Signal Processing*, vol. 2, no.5, pp. 726-731, 2006.

[22] N. Kubota, "Computational Intelligence for Structured Learning of A Partner Robot Based on Imitation", *Info. Sci.,* vol. 171, no. 4, pp. 403-429, 2005.

[23] I. A. Sulistijono and N. Kubota, "Particle Swarm Intelligence Robot Vision for Multiple Human Tracking of A Partner Robot", *Proc. Society of Instrument and Control Eng. Annual Conf.*, pp. 604-608, 2007.

[24] I. A. Sulistijono and N. Kubota, "Evolutionary Robot Vision and Particle Swarm Optimization for Multiple Human Heads Tracking of A Partner Robot", *Proc. IEEE Cong. Evolutionary Comp.*, pp. 1535-1541, 2007.

[25] N.Kubota, T.Ohinata, S.Wakisaka, H.Liu, Extraction of Facial Landmarks for a Partner Robot Based on Evolutionary Computation, *Proc. Comp. Intelligence, Robotics and Autonomous Sys.*, Massey University, Palmerston North, New Zealand, November 28-30, pp. 44-49, 2007.

[26] Naoyuki Kubota, Indra Adji Sulistijono, Evolutionary Robot Vision for People Tracking Based on Local Clustering, 2008 World Automation Congress (WAC2008), Proceeding (CD ROM) of the 6th International Forum on Multimedia and Image Processing (IFMIP2008), Hawaii, USA, Sep 28 - Oct 2, 2008.

[27] Indra Adji Sulistijono, Zaqiatud Darojah, Abdurahman Dwijotomo, Dadet Pramadihanto and Naoyuki Kubota, Facial Expression Recognition Using Back Propagation, Proceeding (CD ROM) of The 2010 International Symposium on Intelligent Systems (iFAN 2010), Tokyo, Japan, Sep 25-26, 2010, paperID:0569.

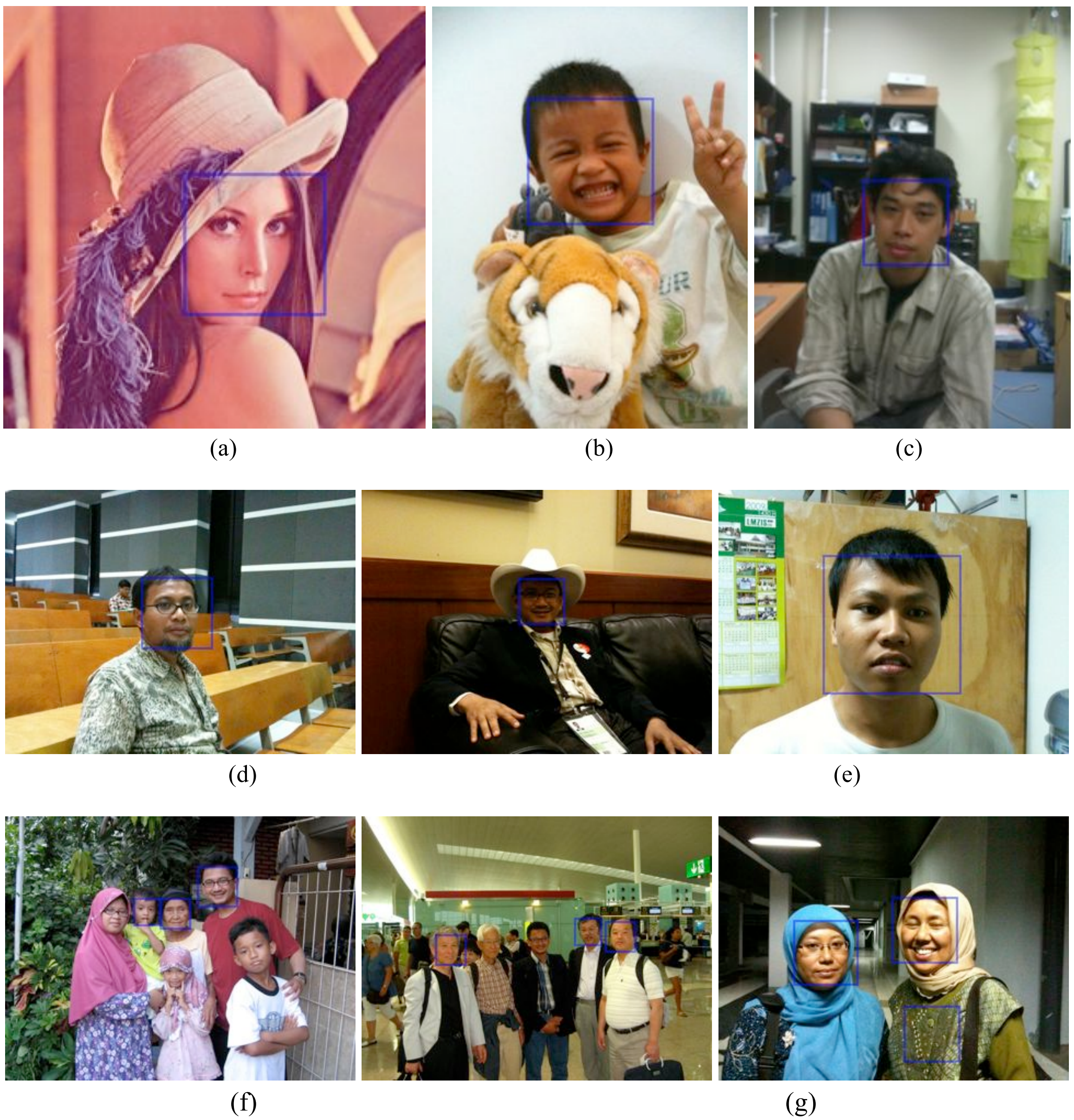

Fig. 8. Multiple human face detection experimental results on iPhone