Speaker Detection and Recognition for a Welfare Robot

Masanori Sugisaka, and Xinjian Fan

Department of Electrical and Electronic Engineering, Oita University, Japan
(Tel : +81-97-554-7831; E-mail: {msugi, fxinjian}@cc.oita-u.ac.jp)

Abstract: Computer vision and natural-language dialogue play an important role in friendly human-machine interfaces for service robots. In this paper we describe an integrated face detection and face recognition system for a welfare robot, which has also been combined with the robot's speech interface. Our approach to face detection is to combine neural network (NN) and genetic algorithm (GA): ANN serves as a face filter while GA is used to search the image efficiently. When the face is detected, embedded Hidden Markov Model (EMM) is used to determine its identity. A real-time system has been created by combining the face detection and recognition techniques. When motivated by the speaker's voice commands, it takes an image from the camera, finds the face inside the image and recognizes it. Experiments on an indoor environment with complex backgrounds showed that a recognition rate of more than 88% can be achieved.

Keywords: Welfare robot, Face detection, Face recognition, NN, GAs, EMM

1. INTRODUCTION

In service robots, many different research areas are involved, e.g., sensor design, control theory, manufacturing science, artificial intelligence, and also computer vision and natural-language dialogue. The latter two are especially important, since service robots should serve as personal assistants. As a consequence, service robots differ from other mobile robotic systems, mainly by their intensive interaction with people in natural environments. In typical environments for service robots, such as hospitals or day-care facilities for elderly people, the demands on the interface between robot and humans exceed the capabilities of standard robotic sensors, such as sonar, laser, and infrared sensors. Thus, in many cases, computer vision as well as natural-language dialogue components become essential parts of such a system.

In this paper, we mainly concentrate on the following two aspects of a service robot: computer vision and natural-language dialogue. Specially, we are mainly interested in the integration of vision and speech to achieve a friendly human-robot interface. The task can be described as follows: when a person (user) comes to the front of the robot, and says, "Hello, robot", the robot will try to find and recognize the person and response to the speaker, "Hello, dear user" (Figure 1). For speech recognition and speech synthesis, we use IBM ViaVoice text-to-speech and speech recognition toolkits. So the main part of this paper is about face detection and face recognition techniques. The system described in this paper has been applied to a real welfare robot (called *Liferobot*), which are being developed in our lab [1]. We first give a brief description of *Liferobot* in Section 2. Then face detection and face recognition techniques are presented in Section 3 and Section 4 respectively. Finally, Section 5 contains our conclusion.

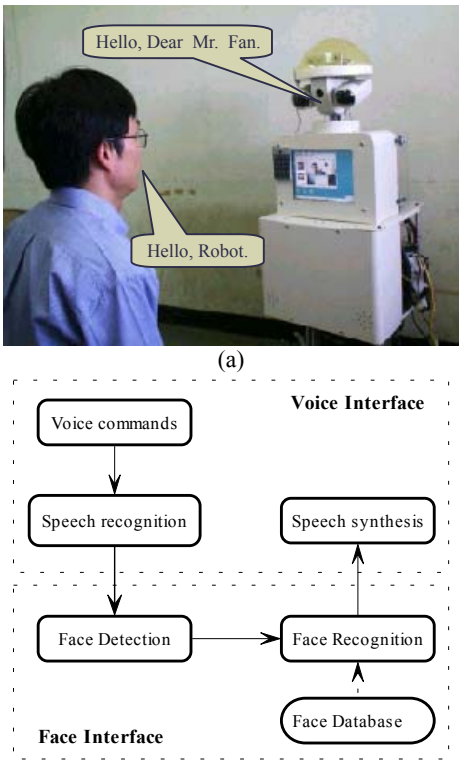

Fig. 1 System overview.

2. LIFEROBOT DESCRIPTION

The Welfare Liferobot (Figure 1.a) is an intelligent autonomous mobile robot with its own control system on-board and the set of sensors to perceive an environment.

The robot control system consists of three networked computers: two PCs (on board) and a notebook. Two PCs are used for robot control (PC-1) and vision processing (PC-2). The notebook with sound card and microphone is used for voice processing. All PCs use Windows 98 SE Japanese version as their operating system.

Two color CCD cameras (EVI-G20) mounted on the rotating head. The cameras support focus, white balance, flicker removal, and back-lighting features which may be used to reduce image noise, blur, skew and warp. Each camera is capable of independent pan and tilt movement. Zoom, pan and tilt values of these cameras are set from the PC serial port using the SONY VISCA protocol. The FDMPCI capture card can provide data in RGB, or YUV formats, and support (depending on model) brightness and contrast enhancement, as well as image scaling functions. For head positioning a stepping motor is used and four LEDs to display a status.

3. FACE DETECTION

The face detection module is motivated by the speaker's voice commands. Its aim is to take images from the

environment and return the location and size which corresponds to a human face. There have been a large body of work on computerized face detection [2]. Our particular system is based on neural network (NN) and genetic algorithms (GA): NN serves as a face filter while GA is used to search the image efficiently. The GA searches the image with a group of subwindows extracted from the input image. The subwindows are evaluated by the NN-based face filter. A face is indicated only when the best individual's filter response is above a given threshold value. Experimental results show that to find a face, only a small number of subwindows need to be evaluated and thus greatly speed the detection process. The main procedures are shown in Figure 2. In the following, we will discuss the procedures in detail.

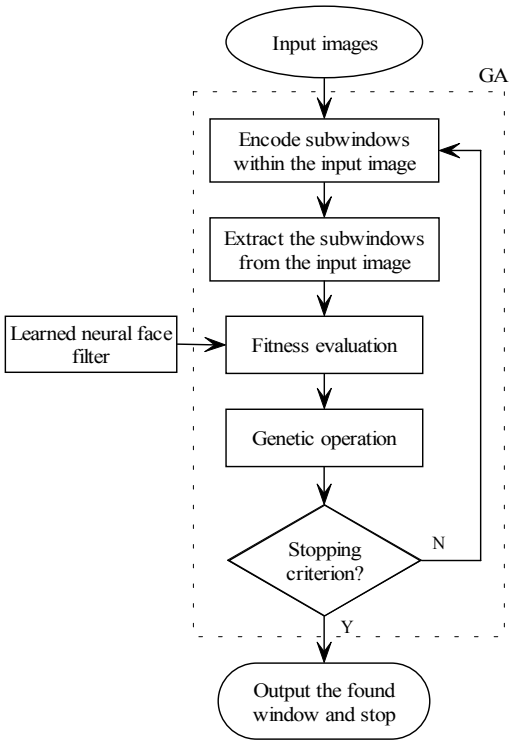

Fig. 2 Main procedures of face detection.

3.1 Neural network-based face filter

The purpose of the face filter is to evaluate a subwindow of size 20×20 pixels extracted from an image, to see how much it likes a face. We use a retinally connected neural network proposed by Rowley [3] to serve as the face filter. The network takes as input a 400-length vector, each corresponding to the gray value of a pixel in an extracted 20×20 subwindow. The subwindow is pre-processed through lighting correction (a best fit linear function is subtracted) and histogram equalization. There are two copies of a hidden layer with 26 units, where 4 connected to 10×10 pixel subregions, 16 connected to 5×5 subregions, and 6 units connected to 20×5 pixels overlapping horizontal stripes. Compared to the fully connected neural networks used in [4][5], the retinal connection type has the advantage of 1) Such a configuration allows the hidden units to detect local features (i.e. mouths or eye pairs) that might be important for face detection; 2) It computes more efficiently than the fully connected networks since it has much fewer weights than the latter.

It results a real value form -1.0 to 1.0, indicating how much the subwindow resembles a face.

The neural filter is trained using standard back-propagation. The face training set is composed of 1000 frontal faces (positive examples), which were normalized into 20×20 pixels. Fifteen additional face examples are generated from each original face image by randomly rotating it (up to 10º), scaling (90% to 110%), translating (up to half a pixel), and mirroring. 6000 random patches chosen from images containing no faces serve as initial non-face training set (negative examples). Additional non-faces were introduced by applying the bootstrap algorithm.

3.2 Genetic search

3.2.1 Representation We choose the center (C_x, C_y) and the size S to define a subwindow. To evaluate subwindows of different sizes using the neural network, we should scale them down to the size of 20×20 (the input size of the neural network). However, if this computation is done on every size of subwindows, it will be very time-consuming. To avoid it, we first build an image pyramid by a scale factor q. The top level (level L) should have a size more than 20×20:

$$INT(\frac{MIN(W, H)}{q^L}) \geq 20 \text{ , gives}$$

$$L = INT\left(q\sqrt{\frac{MIN(W, H)}{20}}\right) \quad (1)$$

where W and H are the width and height of the input image respectively.

Then we let S to be chosen among the following series:

$$20, 20\times q, 20\times q^2, \text{L} , 20\times q^s \text{ L} , 20\times q^L \quad (2)$$

In our experiment, $q = 1.2$, resulting a 14-level size of search windows for a 320×240 input image. This implies that we can detect faces with sizes ranging from 20×20 pixels to 212×212 pixels.

For a subwindow $\mathbf{F}_w = (C_x, C_y, 20\times q^s)^T$, we only need to find its mapped 20×20 window $\mathbf{F}'_w = (C'_x, C'_y, 20)^T$ in level s of the pyramid by:

$$C'_x = INT(\frac{C_x}{q^s}),\ C'_y = INT(\frac{C_y}{q^s}) \quad (3)$$

So the chromosome of an individual consists of 3 genes, represented by the data structure shown in Figure 3.

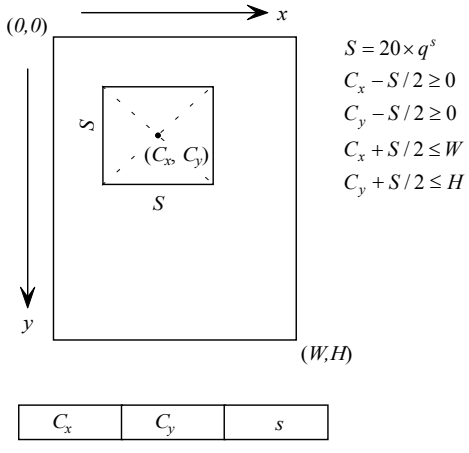

Fig. 3 Chromosome representation and constraints.

3.2.2 Genetic design Many researchers have pointed that

real-coded GAs outperforms conventional binary-coded GAs [7][8]. So a real-coded GA (RGA) is used here.

(1) Crossover operator We applied BLX-α crossover [7][8]: $\mathbf{P}_1$ and $\mathbf{P}_2$ are two chromosomes that have been selected to apply the crossover to them. Two offspring $\mathbf{P}_1$ and $\mathbf{P}_2$ are generated by

$$\mathbf{P}_1' = \lambda\mathbf{P}_1 + (1-\lambda)\mathbf{P}_2$$
$$\mathbf{P}_2' = (1-\lambda)\mathbf{P}_1 + \lambda\mathbf{P}_2 \quad (4)$$

where λ is a uniform random number $\in [-\alpha, 1+\alpha]$. In our experiment, a = 0.3.

(2) Mutation operator Let us suppose that $\mathbf{C} = (c_1, \mathrm{L}, c_2, \mathrm{L}, c_n)$ is a chromosome and $c_i \in [a_i, b_i]$ a gene to be mutated. The gene c_i' is resulted from the application of mutation operator [9]:

$$c_i' = c_i + \gamma \cdot dev_i \quad (5)$$

where dev_i defines the mutation range and it is normally set to $0.1\times(b_i - a_i)$, γ is randomly generated from [-1, 1].

(3) Other operators and parameters for RGA were set as summarized in Table 1.

Table 1. Settings for RGA

Selection	Linear rank selection with elitism
Size of population	100
Maximum generation	100
Probability of crossover	0.90
Probability of mutation	0.15
Stop criterion	A "face" is found or the maximum generation is met.

3.2.3 Fitness evaluation As stated above, each subwindow is evaluated by how well it matches the NN-based face filter. The larger its detection value (*DV*), the more the subwindow resembles a face. The fitness function $F(\mathbf{F}_w)$ is given as

$$F(\mathbf{F}_w) = 0.5\times(1 + DV(\mathbf{F}_w)) \quad \mathbf{F}_w \in \mathbf{T} \quad (6)$$

where $\mathbf{F}_w$ is a subwindow, $DV(\mathbf{F}_w) \in [-1, 1]$.

3.2.4 Face detection experiments We have tested the above algorithm on many scenes of an office environment with different illumination conditions and increasing complexity background. All the scenes in our experiments are 320×240 pixels large, taken by the camera mounted on the the robot's head. In these experiments, the GA can find the face in the scenes most of the time, and obtained only a few false results. Table 2 shows the result tested on 150 images. Figure 4 shows an example and GA's performance plot.

Table 2. Average performance of the face detector

Processing time	Success rate	False detect rate
0.44s	92%	0.7%

4. FACE RECOGNITION

When a face is detected, the system attempts to recognize it. We use the embedded Hidden Markov Model (EMM) for face recognition. This choice was made based on a desire to reproduce the excellent performance reported by Nefian [10]. It should be noted that the detected faces should be properly enlarged because they are generally smaller than their actual sizes.

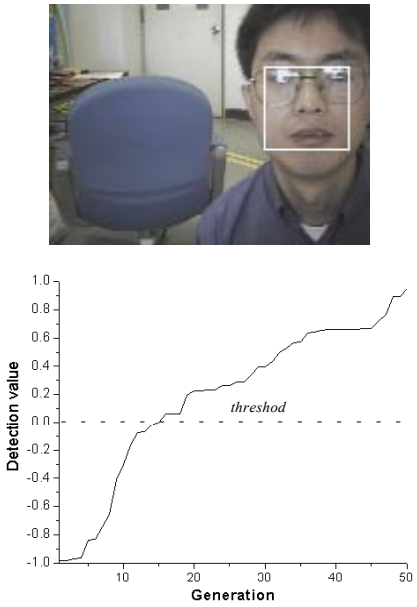

Fig. 4 A Scene and GA's performance plot (average of 30 runs).

4.1 Embedded HMM (EMM)

The EMM consists of a set of super states along with a set of embedded states. The super states may then be used to model two-dimensional data along one direction with the embedded HMM modeling the data along the other direction. The elements of an embedded HMM are:

- The number of super states, N_0, and the set of super states, $S_0 = \{S_{0,i}\}$ $1 \le i \le N_0$
- The initial super state distribution, $\Pi_0 = \{\pi_{0,i}\}$, where $\pi_{0,i}$ are the probabilities of being in super state i at time zero.
- The super state transition probability matrix,
  $$A_0 = \{a_{0,ij}\}$$
  where $a_{0,ij}$ is the probability of transitioning from super state i to super state j.
- The parameters of the embedded HMMs Λ, which include
  – The number of embedded states in the *k*th super state, $N_1^{(k)}$, and the set of embedded states,
  $$S_1^{(k)} = \{s_{1,i}^{(k)}\}$$
  – The initial state distribution, $\Pi_1^{(k)} = \{\pi_{1,i}^{(k)}\}$, where $\pi_{1,i}^{(k)}$ are the probabilities of being in state *i* of super state *k* at time zero.
  – The state transition probability matrix,
  $$A_1^{(k)} = \{a_{1,jk}^{(k)}\}$$
  that species the probability of transitioning from state *k* to state *j*.

- Finally there is the state probability matrix,

$$B^{(k)} = \{b_i^{(k)}(O_{t_0,t_1})\}$$

for the set of observations where O_{t_0,t_1} represent the observation vector at row t_0 and column t_1.

Let $\Lambda^{(k)} = \{\Pi_1^{(k)}, A_1^{(k)}, B^{(k)}\}$ be the set of parameters that define the kth super state. Using a shorthand notation an embedded HMM is defined as the triplet

$$\lambda = (\Pi_0, A_0, \Lambda) \tag{7}$$

where $\Lambda = \{\Lambda^{(1)}, \Lambda^{(2)}, \Lambda\ , \Lambda^{(N_0)}\}$.

For face recognition a separate EMM is built for each face in the database. The candidate face is first converted into a 2D observation sequence by extracting blocks from the face image left to right and then top to bottom. Six 2D-DCT coefficients from each block are used as observation vectors [10]. Then he observation sequence is matched against each of the models in the database:

$$P_r[O \mid \lambda^{(k)}], \quad 1 \le k \le F \tag{8}$$

where F is the total number of different faces in the database.

The highest match is chosen and the person corresponding to the chosen model reveals the identity of the unknown face.

4.2 Face recognition experiment

In the beginning our database contains 10 persons. Each person was trained with 5 face images. We have tested our face recognition system on these 10 people. Each person was tested on 12 images that are acquired autonomously by the robot's vision system. In these runs, the robot recognized the person about 88 percent of the time. Note that the system as a whole has a higher failure rate than the EMM itself, because the robot system can fail at other points along the recognition process. The speed is about 1.2s (0.4s for face detection and 0.8s for face recognition) per frame. Figure 5 shows the interface of the system.

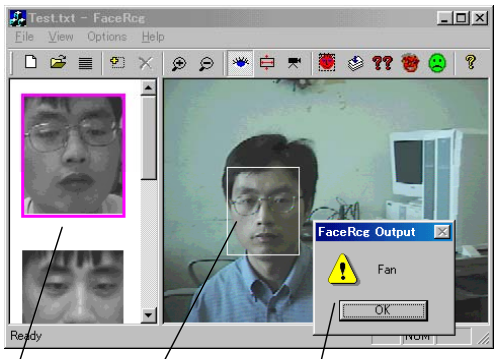

Fig. 5 The interface of the system.

5. CONCLUSION

In order for service robots to interact effectively with people they will have to be able to find and recognize the user. The process described in this paper is a first step in that direction. Our main contributions are:

- Successfully incorporate two techniques which have been tested to be successful – neural network for face detection and embedded HMM for face recognition respectively, into a full face recognition system;
- To avoid the heavy computational cost caused by the face detection process, we propose to use GAs to search the image efficiently. Experiment results have shown that a great speedup has been achieved by using this approach compared to using the exhaustive search.

All of these are implemented on an actual welfare robot – *liferobot*. In the future we hope to add additional human interaction skills, including gesture and pose recognition.

REFERENCES

[1] Masanori Sugisaka and Takuya Adachi, "The controlling of the welfare robot prototype", *Proceedings of the Sixth International Symposium on Artificial Life and Robotics*, pp.309-312, Tokyo, Jan. 2001.

[2] Ming-Hsuan Yang, David Kriegman and Narendra Ahuja, "Detecting Faces in Images: A Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol.24, No.1, pp.34-58, 2002.

[3] Henry A. Rowley , "Neural Network-Based Face Detection", *Thesis submitted for the degree of Doctor of Philosophy*, School of Computer Science, Carnegie Mellon University, 1999.

[4] B. Fasel, "Fast multi-scale face detection", *Technical Report COM-98-04*, IDIAP, 1998.

[5] K. K. Sung and T. Poggio, "Example-based Learning for View-based Human Face Detection", *IEEE Trans. Pattern Anal. & Mach. Intell.*, Vol.20, pp.39-50, 1998.

[6] Goldberg D. , "Genetic algorithms in search, optimization and machine learning", Addison-Wesley, MA, 1989.

[7] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval schemata", *Foundations of Genetic Algorithms 2*, Morgan Kaufmann Publishers, 1993.

[8] F. Herrera, M. Lonzano, and J. L. Verdegay, "Tackling real-coded genetic algorithms: Operators and tools for behaviorial analysis", *Artificial Intelligence Review*, 12(4), 1998.

[9] H. Muhlenbein, and D. Schlierkamp-Voosen, "Predictive Models for the Breeder Genetic Algorithm I", *Evol. Comp.*, Vol.1:1, pp.25-50, 1993.

[10] Ara V. Nefian and Monson H. Hayes III , " Face ecognition using an embedded HMM " , *IEEE International onference Audio Video iometric based Person Authentication*, pp.19-24, 1999.

[11] Intel's open source computer vision library *OpenCV*: http://www.intel.com/research/mrl/research/opencv/