Finger Identification in Hand Gesture Based Human-Robot Interaction

Xiaoming Yin and Ming Xie

School of Mechanical and Production Engineering
Nanyang Technological University
Singapore 639798
p145567279@ntu.edu.sg, mmxie@ntu.edu.sg

Abstract. Natural and friendly interface is critical for the development of service robots. Gesture-based interface offers a way to enable untrained users to interact with robots more easily and efficiently. In this paper, we present a gesture recognition system implemented on our humanoid service robot HARO-1. The system applies RCE neural network based color segmentation algorithm to separate hand images from the complex background. The topological features of the hand are extracted from the binary image of the segmented hand region. Based on the analysis of these simple but distinctive features, hand postures are identified accurately. Experimental results on gesture-based robot programming demonstrated the effectiveness and robustness of the system.

1 Introduction

With the massive influx of computers in society and the increasing importance of service sectors in many of industrialized nations, the market for robots in conventional applications of manufacturing automation is reaching saturation, and the research on robotics is rapidly proliferating in the field of service industries [1] [2]. Service robots are intelligent machines that provide service for human beings and machines themselves. They operate in dynamic and unstructured environment and interact with people who are not necessarily skilled in communicating with robots [3]. Friendly and cooperative interface is thus critical for the development of service robots [4] [5]. Gesture-based interface holds the promise of making human-robot interaction more natural and efficient.

Gesture-based interaction was firstly proposed by M. W. Krueger as a new form of human-computer interaction in the middle of the seventies [6], and there has been a growing interest in it recently. As a special case of human-computer interaction, human-robot interaction is imposed by several constraints [7]: the background is complex and dynamic; the lighting condition is variable; the shape of the human hand is deformable; the implementation is required to be executed in real time and the system is expected to be user and device independent. Numerous techniques on gesture-based interaction have been proposed, but hardly any published work fulfills all the requirements stated above.

R. Kjeldsen and J. Kender [8] presented a realtime gesture system which is used in place of the mouse to move and resize windows. In this system, the hand is segmented from the background using skin color and the hand's pose is classified using a neural net. A drawback of the system is that its hand tracking has to be specifically adapted for each user. The Perseus system developed by R. E. Kahn [9] was used to recognize the pointing gesture. In the system, a variety of features, such as intensity, edge, motion, disparity and color has been used for gesture recognition. This system is implemented only in a restricted indoor environment. In the gesture-based human-robot interaction system of J. Triesch and C. Ven Der Malsburg [7], the combination of motion, color and stereo cues was used to track and locate the human hand, and the hand posture recognition was based on elastic graph matching. This system is person independent and can work in the presence of complex backgrounds in real time. But it is prone to noise and sensitive to the change of the illumination because its skin color detection was based on a defined prototypical skin color point in the HS plane.

This paper present a simple, fast and robust system that segment and recognize hand gestures for human-robot interaction. In the system, a novel color segmentation algorithm developed on the basis of Restricted Coulomb Energy (RCE) neural network is applied to segment hand images. This method uses the skin color prototype to describe the skin color. With the abundant skin color prototypes that are derived from the training procedure of the RCE network, the system is capable to characterize the distribution region of skin colors accurately in the color space and segment various hand images efficiently from the complex background. The topological features of the hand are extracted from the binary image of the segmented hand region, and the recognition of hand postures is based on the analysis of these features.

The rest of the paper is organized as follows. The problem of hand image segmentation is addressed in the next section. The proposed algorithms for hand feature extraction and posture recognition are then presented in

Section 3. Section 4 introduces our humanoid service robot HARO-1 and discusses experimental results conducted on gesture-based robot programming for HARO-1. Finally, conclusions and future work are given in Section 5.

2 Hand Image Segmentation

Hand image segmentation separates the hand image from the background. It is the first important step in every hand gesture recognition system, and all subsequent stages heavily rely on the quality of the segmentation. Two types of cues, color cues and motion cues, are often applied for hand image segmentation [10]. Motion cues are used in conjunction with certain assumptions [11] [12]. For example, the gesturer is stationary with respect to the background that is also stationary. Such assumption restraints its application at the occasion when the background is not stationary, which is the usual case for service robots. The characteristic color of human skin makes color a stable basis for skin segmentation. In our work, a novel color segmentation approach based on RCE neural network has been developed for hand segmentation.

2.1 Skin Color Distribution

Color segmentation techniques rely on not only the segmentation algorithms, but also the color spaces used. RGB, HSI, and L*a*b* are the most commonly used color spaces in computer vision, and have all been applied in numerous proposed color segmentation techniques. After exploring the algorithm in these three color spaces respectively, we found L*a*b* color space is the most suitable for our hand segmentation algorithm.

L*a*b* color space is the uniform color space defined by the CIE (Commission International de l'Eclairage) in 1979. It maps equal Euclidean distance in the color space to equal perceived color difference. The transformation from RGB to L*a*b* color space is defined as follows [13]:

$$L^* = \begin{cases} 116\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - 16 & \text{if } \frac{Y}{Y_n} > 0.008856 \\ 903.3\left(\frac{Y}{Y_n}\right) - 16 & \text{if } \frac{Y}{Y_n} \leq 0.008856 \end{cases} \tag{1}$$

$$a^* = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right] \tag{2}$$

$$b^* = 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right] \tag{3}$$

$$\text{where } f(t) = \begin{cases} t^{\frac{1}{3}} & \text{if } t > 0.008856 \\ 7.787 * t + 16/116 & \text{if } t \leq 0.008856 \end{cases} \tag{4}$$

X, Y and Z are tristimulus values of the specimen, and calculated from the values of R, G and B as follows:

$$X = 0.607R + 0.174G + 0.201B \tag{5}$$

$$Y = 0.299R + 0.587G + 0.114B \tag{6}$$

$$Z = 0.000R + 0.066G + 1.117B \tag{7}$$

X_n, Y_n and Z_n are tristimulus values of a perfect reflecting diffuser, which are selected to be 250.410, 255.000 and 301.655 respectively.

A common belief is that different people have different skin colors, but some studies show that such a difference lies largely in intensity than color itself [14]. We quantitatively investigated the skin color distribution of different human hand under different lighting conditions. It is found that skin colors have the specific distribution in the color space shown in Fig. 1, which has the following properties:

1. Skin colors distribute in a small region in the color space.
2. Skin colors do not fall into a random region, but form clusters at specific points.
3. Skin color clusters have more difference in intensity than in color.
4. Skin color distribution translates along the "Lightness" axis in the color space with the change of lighting conditions.

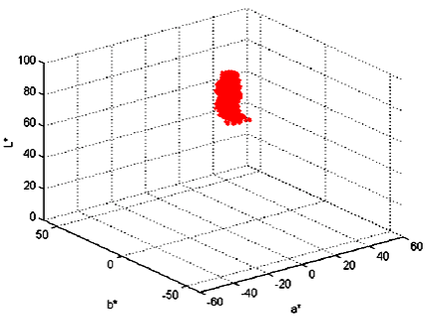


Fig. 1. Skin color distribution in L*a*b* color space

2.2 Skin Color Modeling

Skin colors cluster in a specific small region in the color space, but the shape of the skin color distribution region is complicated and irregular. Common color segmentation techniques based on histogram are not effective enough to segment the hand image for the complex and dynamic background due to the difficulty to properly select threshold.

In this paper, a new color segmentation algorithm based on RCE neural network is proposed. RCE neural network was designed as a general-purpose, adaptive pattern classification engine [15]. It consists of three layers of neuron cells, with a full set of connections between the first and second layers, and a partial set of connections between the second and third layers. Fig. 2(a) shows the network structure used for hand segmentation. Three cells on the input layer of the network are designed to represent the L*a*b* color values of a pixel in the image. The middle layer cells are called prototype cells, and each cell contains color information about an example of the skin color class that occurred in the training data. The cell on the output layer corresponds to the skin color class.

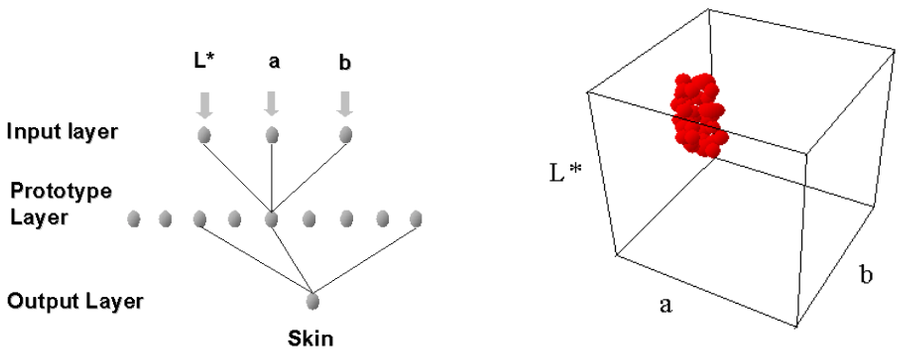


Fig. 2. (a) Architecture of RCE neural network for hand segmentation, (b) Distribution region of skin colors in L*a*b* color space.

During training procedure, the RCE network allocates the positions of prototype cells and modifies the sizes of their corresponding spherical influence fields, so as to cover the arbitrarily complex distribution region of skin colors in the color space. Fig 2(b) shows the distribution region of skin colors constructed by skin color prototype cells and their spherical influence fields in L*a*b* color space. During running, REC responds to input color signals in the fast response mode. If an input color signal falls into the distribution region of skin colors, this input color signal belongs to the skin color class, and the pixel represented by this color signal are identified as skin texture in the image.

2.3 Hand Segmentation

In our system, the training procedure of the REC network can be taken either offline or online. In offline training, the RCE network is trained using the training data saved in a file before segmentation. In online training, parts of

the hand section in the video image are selected as the training set of the network directly by the mouse of the computer. The online train set can also be saved and used for offline training at next time.

Fig. 3. Hand segmentation results

During running, the RCE network identifies all the skin-tone pixels in the image. There are occasions that other skin-tone objects such as faces are segmented, or some non-skin pixels are falsely detected due to the effects of lighting conditions. We assume the hand is the largest skin-tone object in the image, and use the technique of grouping by connectivity of primitive pixels to further identify the region of the hand. With numerous skin color prototype cells together with their different spherical influence fields, the RCE network is capable to segment various hand images under variable lighting conditions from the complex background after trained properly. Fig. 3 shows some segmentation results, in that the regions of the hand are separated perfectly from the complex background.

3 Hand Posture Recognition

3.1 Feature Selection

Hand segmentation is followed by feature extraction. Contour is the commonly used feature for accurate recognition of hand gestures, and can be extracted easily from the binary image of the segmented hand region. In the study, we found it is difficult to extract the smooth and continuous contour of the hand because the segmented hand region is irregular, especially when the REC neural network is not trained sufficiently. In Fig. 4, (a) shows the segmentation of hand image, (b) shows the binary image of the segmented hand section, (c) shows the hand edge detected by Deriche edge detection algorithm, and (d) shows the hand boundary of 8-connection.

Segen and Kumar [16] extracted the points along the boundary where the curvature reaches a local extremum as “local features”, and used those features which are labeled “peaks” or “valleys” to classify hand postures. However, if the boundary is not smooth and continuous, it is difficult to identify peaks and valleys correctly.

The topological features of the hand, such as the number and positions of fingers, are other distinctive features of hand gestures. In this paper, we proposed a new method for accurate recognition of hand postures, which extract topological features of the hand from the binary image of the segmented hand region, and recognize hand postures on the basis of the analysis of these topological features.

3.2 Feature Points Extraction

In order to find the number and positions of fingers, the edge points of fingers are the most useful features. We extract these points using the following proposed algorithm:

1. Calculate the center of mass of the hand from the binary image of the segmented hand region, in that pixel value 0 represents the background and 1 represents the hand image;
2. Draw the search circle with the radius r at the position of the center of mass;
3. Find all the points $\mathbf{E} = \{P_i, i = 0, 1, 2, ..., n\}$ that have the transition either from pixel value 0 to 1, or 1 to 0 along the circle;
4. If the distance between two conjoint points $D = |P_i P_{i-1}| < \text{threshold } \lambda_d$, delete P_i and P_{i-1};

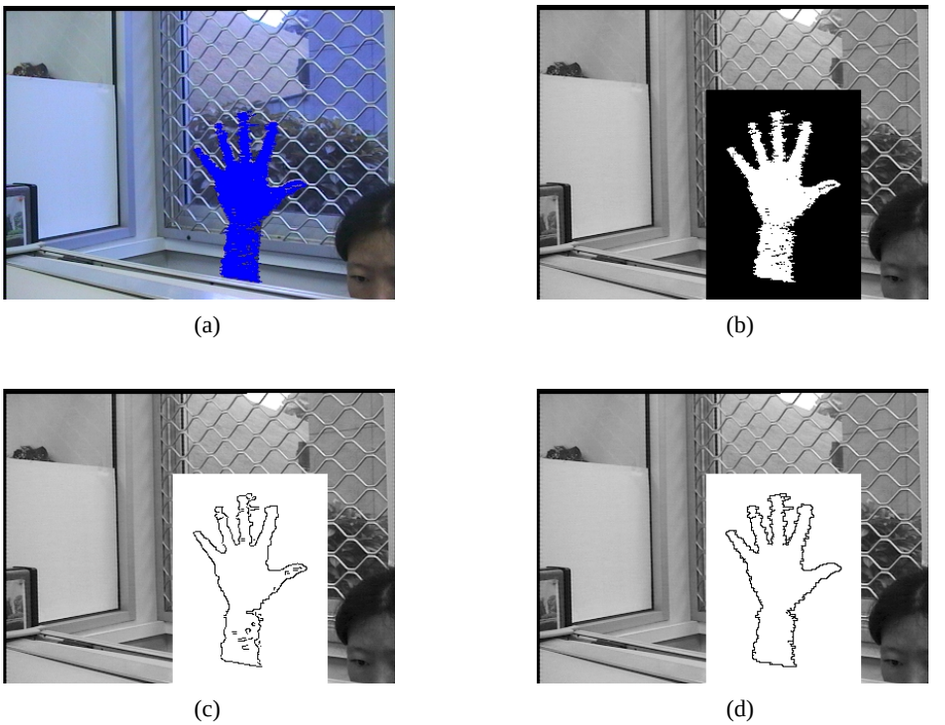


Fig. 4. (a) Segmentation of hand image, (b) Binary image of the segmented hand region, (c) Hand edge detected by Deriche, (d) Hand boundary of 8-connection.

5. Increment the radius r and iterate Step 2 to 4, until $r > \frac{1}{2}$(the width of the hand region).

The purpose of Step 4 is to remove the falsely detected edge points resulted from imperfect segmentation, and the distance threshold λ_d is selected to be 4 pixels in this paper. For example, in Fig. 5(a), Point $P_{i-1}(0,1)$ has the transition from pixel value 0 to 1, and $P_i(1,0)$ has the transition from 0 to 1. $(P_{i-1}(0,1), P_i(1,0))$ indicates the twig and $(P_{j-1}(1,0), P_j(0,1))$ indicates the hole in the image. $P_{i-1}(0,1)$, $P_i(1,0)$, $P_{j-1}(1,0)$ and $P_j(0,1)$ can all be detected at Step 3, but they are not the actual edge points of fingers. Since the distances between them are less than the threshold, they will be deleted at Step 4. This step can removal most of falsely detected edge points, but there are occasions that one finger is divided into several branches if there are bigger holes in the image, or several fingers are merged into one branch if these fingers are too close. So we define the branch as follows:

Definition 1. *The branch is the segment between $P_{i-1}(0,1)$ and $P_i(1,0)$. Where $P_{i-1}(0,1)$ and $P_i(1,0)$ are the two conjoint feature points detected on the search circle. $P_{i-1}(0,1)$ has the transition from pixel value 0 to 1, and $P_i(1,0)$ has the transition from 1 to 0.*

A branch indicates the possible presence of a finger. Then the extracted feature points accurately characterize the edge points of branches of the hand, including fingers and elbow. Fig. 5(b) shows the part of Fig. 4(b) with the scale of 200%, in that the green circles represent the search circles and the red points represent the extracted feature points.

3.3 Branch Number and Phase Determination

For each branch, two edge points can be found on the search circle, so half of the feature points found on the search circle just indicate the branch number of the hand posture. But the feature points on the different search circles are varied, how to determine the correct branch number is critical. In this paper, we define the following function to determine the probability p_i for each branch number:

$$p_i = a_i * \frac{C_i}{N}, i = 0, 1, 2, ..., 6 \tag{8}$$

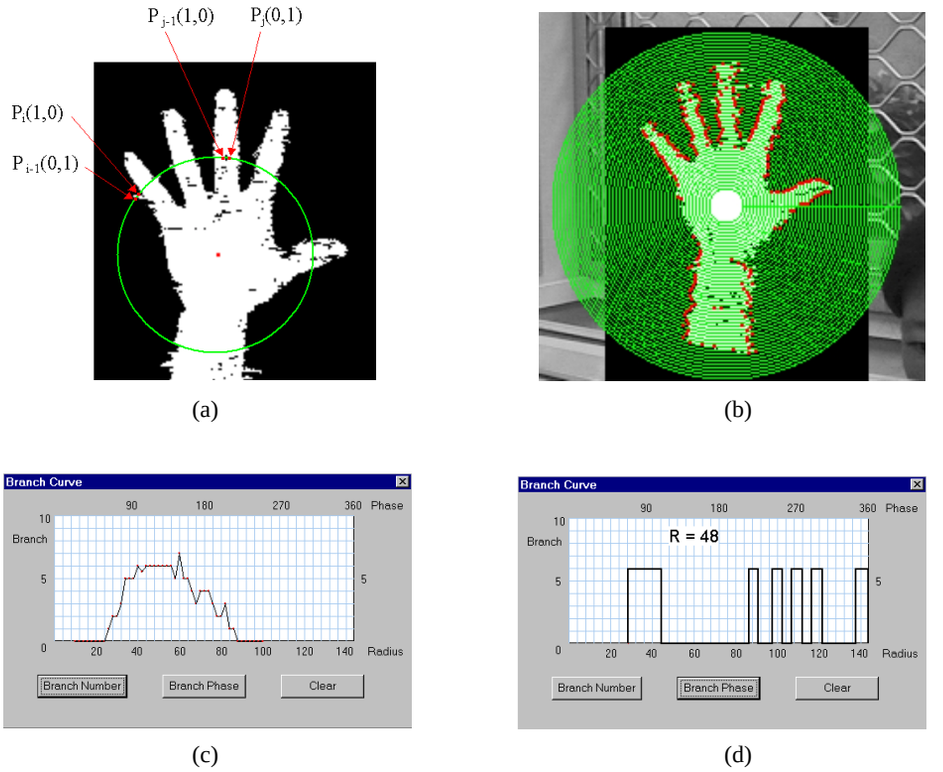


Fig. 5. (a) Falsely detected edge points, (b) Feature points extracted from the binary image of the segmented hand region, (c) Plot of branch number of the hand vs the radius of the search circle, (d) Plor of branch phase of the hand on the selected search circle.

Where C_i is the number of the search circles on that there are i branches; N is the total number of the search circles; a_i is the weight coefficient. We have $a_1 < a_2 < ... < a_i < ... < a_6$, because the number of the branches may decrease when the search circle is beyond the thumb or little finger. Then the branch number with the biggest probability is selected as the most possible branch number BN.

In practice, the branch number BN can also be determined as follows:

1. Find all the branch numbers **K** (a set) whose occurrences are bigger than a threshold λ_n.
2. Among the munbers in **K**, choose the biggest one as the branch number BN.

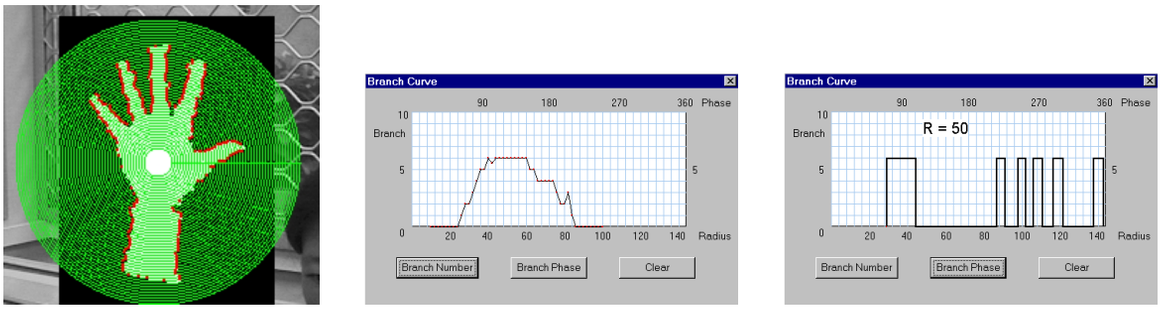
The biggest number in **K** is selected as BN, because the biggest number may not have the most occurrence when the search circle is beyond the fingers, but when its occurrence is bigger than the threshold, it should be the most possible branch number. This method is easier to implement, and is very effective and reliable with the threshold λ_n selected to be 6 in our system.

After the branch number BN is determined, the branch phase can be obtained easily. Here we define the branch phase as follows:

Definition 2. *The branch phase is the positions of the detected branches on the search circle, described by angle.*

In this paper, we selected the middle one of the search circles on that there are BN branches to obtain the branch phase. Fig. 5(c) shows the relationship between the branch number and the radius of the search circle, and (d) shows the radius of the selected search circle, and the branch phase on this circle.

Some morphological operations, such as dilation and erosion, are useful for improvement of the binary image of the segmented hand region, but the branch number and phase obtained from the image after improvement is the same as above (see Fig. 6). It indicates that the feature extraction algorithm has good robustness to noise, and can extract the correct branch number and phase reliably from the segmented hand image even though the segmentation is not very good.



(a) (b) (c)

Fig. 6. (a) Feature points extracted from the binary image after morphological operations, (b) Plot of the branch number of the hand vs the radius of the search circle, (c) Plot of the branch phase of the hand on the selected search circle.

3.4 Posture Recognition

After the branch phase is determined, the width of each branch BW_i can be obtained easily from the branch phase. The widest branch must be the elbow, and is used as the base branch B_0. Then the distance from other branch B_i to B_0 can be calculated, that is just the distance between the finger and the elbow BD_i. Using these parameters mentioned above: the branch number BN, the width of the branch BW_i, the distance between the finger and the elbow BD_i, the hand posture can be recognized accurately.

Such parameters are all very simple and easy to estimate in real time, and they are also distinctive enough to differentiate those hand postures defined explicitly. The recognition algorithm also possesses the property of rotational invariance and user independence because the topological features of human hands are quite similar and stable.

4 Implementation

4.1 Humanoid Service Robot HARO-1

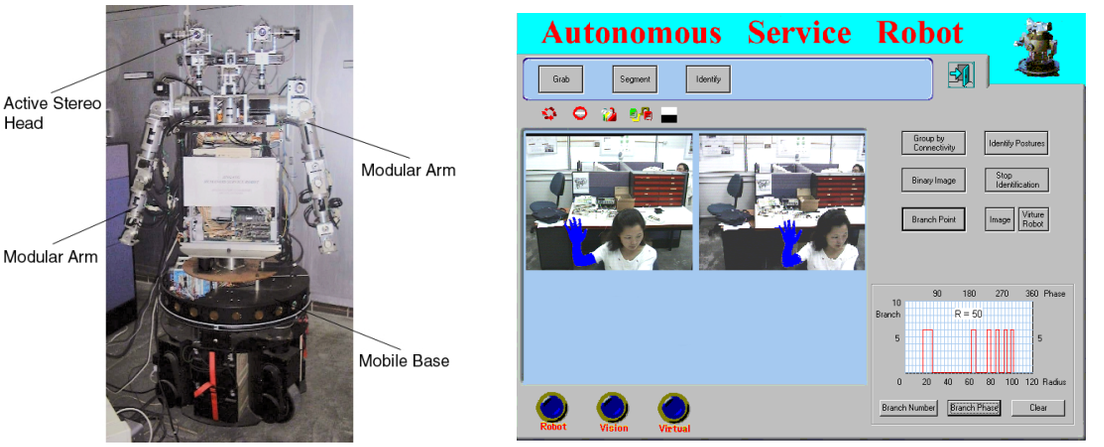
Our research on hand gesture recognition is a part of the project of Hybrid Service Robot System, in which we will integrate various technologies, such as real robot control, virtual robot simulation, hand gesture recognition, gesture-based robot programming etc., to build a multi-modal and intelligent human-robot interface.

Service robots are programmable, sensor-based, freely moving appliances that fully or semi-automatically accomplish service tasks, such as maintenance, assistance, security, housekeeping, entertainment and so on. Humanoid service robots are gaining more attentions. They have more advantages for service applications because they possess the appearance similar to human, can imitate human-like behaviors, and can interact with people in a more natural way. Fig. 7(a) shows the HARO-1 human-alike service robot at our lab. It is designed and developed by ourselves, and mainly consists of an active stereo vision head on modular neck, two modular arms with active links, omnidirectional mobile base, dextrous hands under development and computer system. Each modular arm has 3 serially connected active links with 6 axes. Fig. 7(b) shows the graphic user interface developed using Visual C++.

4.2 Gesture-Based Robot Programming

In order to carry out a useful task, the robot has to be programmed. Robot programming is the act of specifying actions or goals for the robot to perform or achieve. The usual methods of robot programming are based on the keyboard, mouse and teach-pendant. But service robots necessitate new programming techniques because they operate in everyday environment, and have to interact with people that are not necessarily skilled in communicating with robots. Gesture-based programming offers a way to enable untrained users to instruct service robots easily and efficiently.

Based on hand gesture recognition, we have proposed a task programming method for our service robot. In this method, we define task gestures and corresponding motion gestures respectively, and associate them during the training procedure, so that the robot will perform all the motions associated with a task if that task gesture is presented to the robot by the user. Then, the user can interact with the robot and guide the behavior of the robot by using various task gestures easily and efficently.



(a) (b)

Fig. 7. (a) HARO-1 humanoid service robot, (b) Graphic user interface.

The postures shown in Fig. 8 all have distinctive topological features to be recognized. Posture a to f represents the six axes of the robot arm respectively, Posture g means “move’, and Posture h means “stop”. We use them as motion gestures to control the movements of the six axes of one robot arm.

As shown in Fig. 7(b), live images with the size of 384×288 are captured through two CCD video cameras (EVID31, SONY) in our system. At the end of each video field the system processes the pair of images, and output the detected hand infomation. The processing is divided into two phases: hand tracking phase and posture recognition phase. At the beginning, we have to segment the whole image to locate the hand, because we have no any information on the position of the hand. After the initial search, we do not need to segment the whole image, but a smaller region surrounding the hand, since we can assume continuity of the position of the hand during the tracking. At the tracking phase, the hand is segmented using the approach described in Section 2 from a low resolution sampling of the image , and can be tracked reliably at 3-5Hz on a normal 450MHz PC. The system also detects the motion features of the hand such as pauses during the tracking phase. Once a pause is confirmed, the system stops the tracking, crops a high resolution image tightly around the hand and performs a more accurate segmentation based on the same techniques. Then the topological features of the hand is extracted from the segmented hand image and the hand posture is classified based on the analysis of these features as described in Section 3. If the segmented hand image is recognized correctly as one of the postures defined in Fig. 8, the robot will respond by moving the corresponding axis of its arm. if the segmented image can not be recognized because of the present of noises, the robot will not output any response. The time spent on the segmentation of the high resolution image is less than 1 second, and the whole recognition phase can be accomplished within 1.5 seconds. After the posture recognition phase is finished, the system continues to track the hand until another pause is detected.

Preliminary trials were conducted with users of different age, gender and skin color. The robot successfully recognized each of gestures with the accuracy of more than 95% after the RCE network was trained properly. The recognition accuracy may decrease in the case that the user and lighting condition changed too much, because the previous training of the RCE network became insufficient. But this problem can be solved easily by selecting parts of undetected hand sections as the training data using the mouse, and incrementally performing the online training. There is no need to re-present the entire training set to the network.

5 Conclusion and Future Work

We have presented a gesture recognition system implemented on a real humanoid service robot. The system applies RCE neural network to segment hand images. The RCE network is capable to characterize the distribution region of all skin colors in color space with numerous skin color prototype cells and their influence fields. The recognition of hand postures is based on the topological features of the hand that are extracted from the binary image of the segmented hand region. The topological features of human being are quite similar and stable. So the recognition system has the following properties, meeting all the requirements for human-service robot interaction: robustness in dynamic and complex background; adaptability to lighting variations; rotational invariance; real-time performance;

user and device independence. Eight hand postures have been used for gesture-based programming of the service robot HARO-1. Experimental results demonstrated the effectiveness and robustness of the system.

In our future work, we intend to introduce binocular stereo vision to estimate parameters of 3D pose, and handle more gestures, in order to make robot programming and interaction more efficiently and intuitively.

References

1. G. Bekey, "Needs for robotics in emerging application: a research agenda," *IEEE Robotics & Automation*, vol. 4, no. 4, pp. 12–14, 1997.
2. G. Kaplan, "Technology 1998 analysis & forecast-industrial electronics," *IEEE Spectrum*, no. 1, pp. 73–76, 1998.
3. P. Dario, E. Guglielmelli, V. Genovese, and M. Toro, "Robot assistants: Applications and evolution," *Robotics and Autonomous Systems*, vol. 18, pp. 225–234, 1996.
4. M. Ejiri, "Towards meaningful robotics for the future: Are we headed in the right direction," *Robotics and Autonomous Systems*, vol. 18, pp. 1–5, 1996.
5. K. Kawamura, R. Pack, M. Bishay, and M. Iskarous, "Design philosophy for service robots," *Robotics and Autonomous Systems*, vol. 18, pp. 109–116, 1996.
6. M. W. Krueger, *Artificial Reality II*. Addison-Wesley, 1991.
7. J. Triesch and C. V. D. Malsburg, "A gesture interface for human-robot interaction," in *Proceedings of 3th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 546–551, 1998.
8. R. Kjeldsen and J. Kender, "Finding skin in color images," in *Proceedings of International Conference on Automatic Face and Gesture Recognition*, (Killington, Vt.), pp. 312–317, 1996.
9. R. E. Kahn, M. J. Swain, P. N. Prokopowicz, and R. J. Firby, "Gesture recognition using the perseus architecture," in *Proceedings of 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (San Francisco), pp. 734–741, 1996.
10. V. L. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, 1997.
11. W. T. Freeman and C. D. Weissman, "Television control by hand gestures," in *Proceedings of International Workshop on Automatic Face and Gesture Recognition*, (Zurich, Switzerland), pp. 179–183, 1995.
12. C. Maggioni, "Gesturecomputer - new ways of operation a computer," in *Proceedings of International Workshop on Automatic Face and Gesture Recognition*, (Zurich, Switzerland), 1995.
13. J. K. Kasson and W. Plouffe, "A analysis of selected computer interchange color spaces," *ACM Transaction on Graphics*, vol. 11, no. 4, pp. 373–405, 1992.
14. J. Yang, W. Lu, and A. Waibel, "Skin-color modeling and adaptation," in *Proceedings of ACCV'98*, (Hong Kong), pp. 687–694, 1998.
15. D. L. Reilly, L. N. Cooper, and C. Elbaum, "A neural network mode for category leaning," *Biological Cybernetics*, vol. 45, pp. 35–41, 1982.
16. J. Segen and S. Kumar, "Fast and accurate 3d gesture recognition interface," in *Proceedings of 14th International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 86–91, 1998.

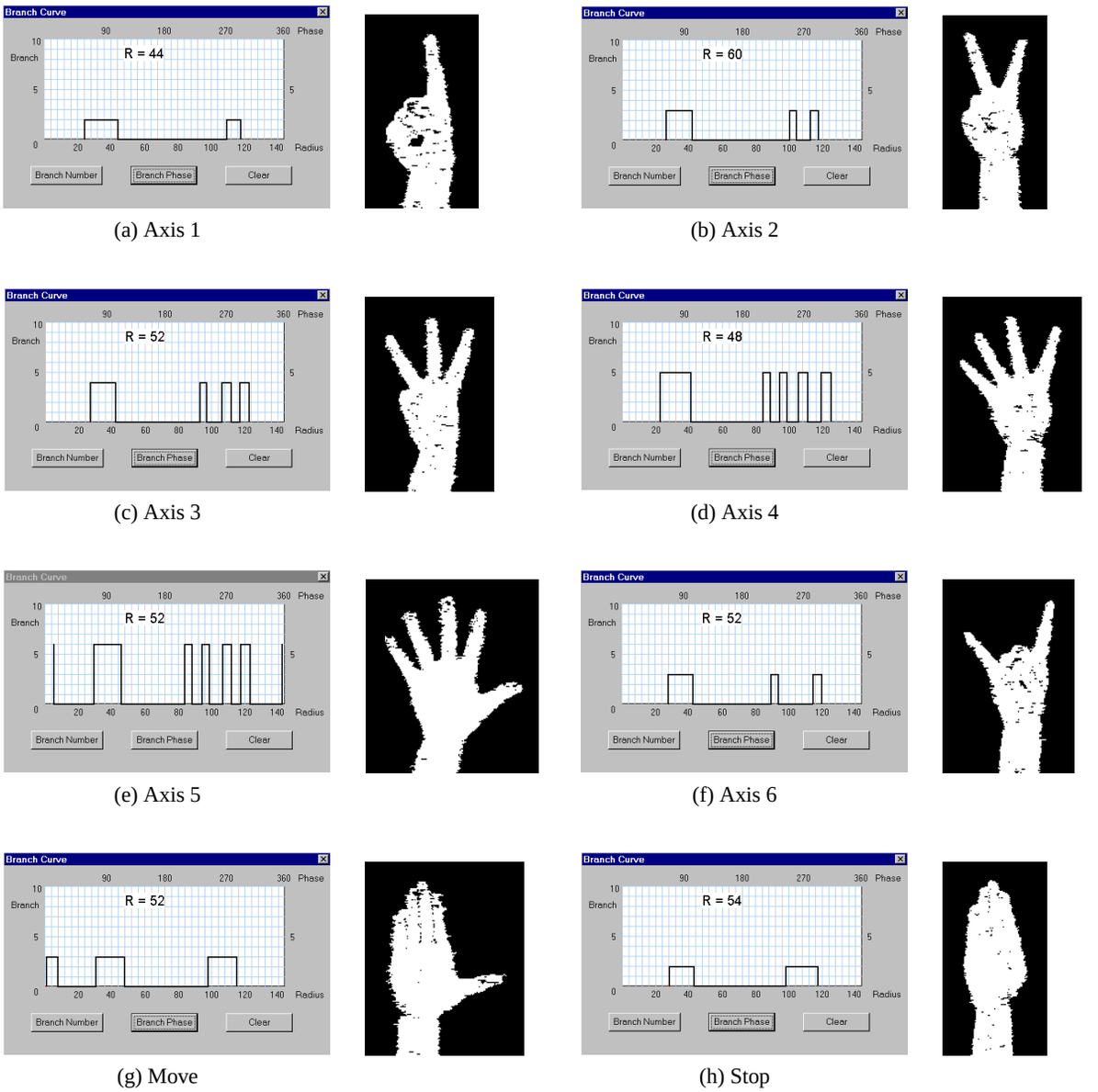


Fig. 8. Hand postures used for robot programming