

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Hand Posture Segmentation, Recognition and Application for Human-Robot Interaction

Xiaoming Yin and Ming Xie

*Singapore Institute of Manufacturing Technology, Nanyang Technological University
Singapore*

1. Introduction

Human hand gestures provide the most important mean for non-verbal interaction among people. They range from simple manipulative gestures that are used to point at and move objects around to more complex communicative ones that express our feelings and allow us to communicate with others. Migrating the natural means that human employ to communicate with each other such as gestures, into Human-Computer Interaction (HCI) has been a long-term attempt. Numerous approaches have been applied to interpret hand gestures for HCI. In those approaches, two main categories of hand gesture models are used. The first group of models is based on appearance of the hand in the visual images. Gestures are modeled by relating the appearance of any gesture to that of the set of predefined template gestures [Pavlovic et al., 1996] [Ahmad et al., 1997]. Appearance-based approaches are simple and easy to implement in real time, but their application is limited to the recognition of a finite amount of hand gestures and they are mostly applicable to the communicative gestures.

The second group uses 3D hand models. 3D hand models offer a way to model hand gestures more elaborately. They are well suitable for modeling of both manipulative and communicative gestures. Several techniques have been developed in order to capture 3D hand gestures. Among those, glove-based devices are used to directly measure joint angles and spatial positions of the hand. Unfortunately, such devices remain insufficiently precise, too expensive and cumbersome, preventing the user from executing natural movements and interacting with the computer intuitively and efficiently. The awkwardness in using gloves and other devices can be overcome by using vision-based interaction techniques. These approaches suggest using a set of video cameras and computer vision techniques to reconstruct hand gestures [Lee and Kunii, 1995] [Lathuiliere and Herve, 2000]. Vision-based approaches are gaining more interest with the advantages of being intuitive, device-independent and non-contact.

Gesture-based interaction was firstly proposed by M. W. Krueger as a new form of human-computer interaction in the middle of the seventies [Krueger, 1991], 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 [Triesch and Malsburg, 1998]: 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. Render [Kjeldsen and Render, 1996b] 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 [Kahn et al., 1996] was used to recognize the pointing gesture. In the system, a variety of features, such as intensity, edge, motion, disparity and color have been used for gesture recognition. This system is implemented only in a restricted indoor environment. In the gesture-based human-robot interaction system proposed by J. Triesch and C. Ven Der Malsburg [Triesch and Malsburg, 1998], 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 is based on a defined prototypical skin color point in the HS plane.

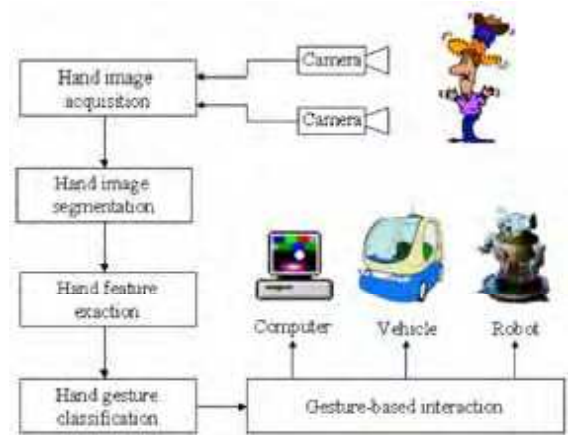


Figure 1. Process of hand gesture recognition

Usually the classical image processing pipeline as shown in Fig. 1 is used for hand gesture recognition. During the step of hand image acquisition, the pictures taken by the camera, in which a hand is to be seen, are digitized and prepared for further processing. It is usually automatically accomplished by a frame grabber. At the step of hand image segmentation, those areas in the picture, which represent the hand, are separated from the background. The aim of the step of hand feature extraction is to derive the smallest possible amount of features out of the segmented hand region, in order to differentiate the different given gestures. The last step is hand gesture classification, whereby the type of gesture shown in the picture are defined on the basis of extracted characteristics.

In this chapter, we present a novel hand posture recognition system. According to the process of hand gesture recognition, we first present a new color segmentation algorithm developed based on RCE neural network for hand image segmentation. Then we extract the topological features of the hand from the binary image of the segmented hand region. Based

on these features, we proposed a new method for accurate recognition of 2D hand postures. We also propose to use the stereo vision and 3D reconstruction techniques to recover 3D hand postures, and give a new approach to estimate the epipolar geometry between two uncalibrated hand images. Finally, we demonstrate the application of our hand gesture recognition system to human-robot interaction.

2. Hand Image Segmentation

Hand image segmentation separates the hand image from the background. It is the most important step in every hand gesture recognition system. All subsequent steps heavily rely on the quality of the segmentation. Two types of cues, color cues and motion cues, are often applied for hand image segmentation [Pavlovic et al., 1997]. Motion cues are used in conjunction with certain assumptions [Freeman and Weissman, 1995] [Maggioni, 1995]. For example, the gesturer is stationary with respect to the background that is also stationary. Such assumption restraints its application on 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 [Quek et al., 1995] [Kjeldsen and Render, 1996a]. In this section, a novel color segmentation approach based on RCE neural network is presented for hand segmentation.

2.1 Skin Color Modeling

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.

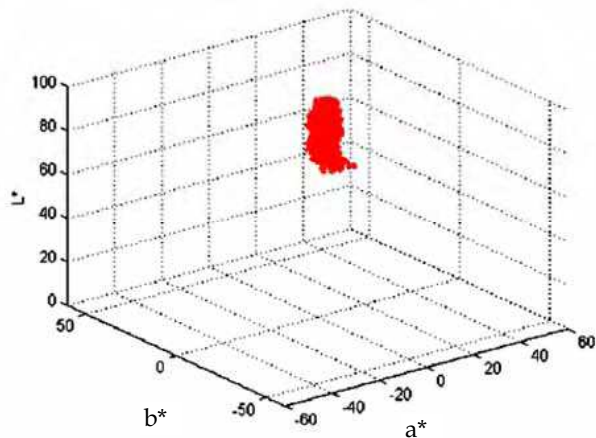


Figure 2. Skin color distribution in $L^*a^*b^*$ color space

$L^*a^*b^*$ 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 [Kasson and Plouffe, 1992]:

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

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

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

where

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

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

$$X = 0.607R + 0.174G + 0.201B \quad (5)$$

$$Y = 0.299R + 0.587G + 0.114B \quad (6)$$

$$Z = 0.000R + 0.066G + 1.117B \quad (7)$$

X_n , Y_n and Z_n are tristimulus values of a perfect reflecting diffuser, which are selected to be 237.448, 244.073 and 283.478 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 [Yang et al., 1998] [Jones and Rehg, 1999]. We quantitatively investigated the skin color distribution of different human hands under different lighting conditions. It is found that skin colors cluster in a small region in the $L^*a^*b^*$ color space and have a translation along the lightness axis with the change of lighting conditions, as shown in Fig. 2.

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 hand images from complex and dynamic backgrounds due to the difficulty of threshold selection. In our work, a new color segmentation algorithm based on RCE neural network has been developed.

RCE neural network was designed as a general-purpose, adaptive pattern classification engine [Reilly et al., 1982]. 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. 3(a) shows the network structure used for hand segmentation. Three cells on the input layer 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 of an example of the skin color class which occurred in the training data. The cell on the output layer corresponds to the skin color class.

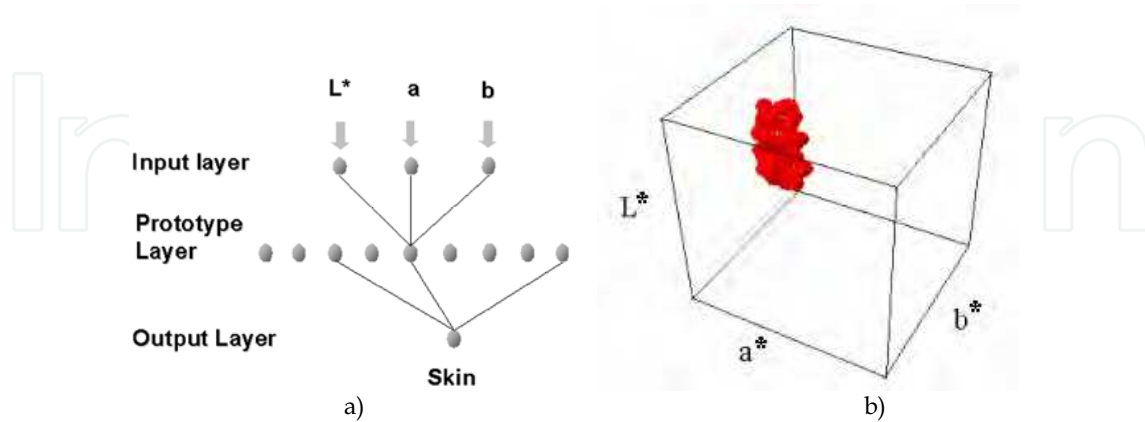


Figure 3. (a) Architecture of RCE neural network for hand segmentation, (b) Distribution region of skin colors in $L^*a^*b^*$ color space

2.2 Hand Segmentation

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 arbitrarily complex distribution region of skin colors in the color space. Fig 3(b) shows the distribution region of skin colors constructed by skin color prototype cells and their spherical influence fields in the $L^*a^*b^*$ color space. During running, the RCE network 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 is classified into the skin color class, and the pixel represented by this color signal is identified as skin texture in the image.

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 abundant skin color prototype cells together with their different spherical influence fields, the RCE network is capable of accurately characterizing the distribution region of skin colors in the color space and efficiently segment various hand images under variable lighting conditions from complex backgrounds after having been trained properly. Fig. 4 shows some segmentation results, in that the hand regions are separated perfectly from the complex backgrounds. The RCE neural network based hand image segmentation algorithm is described in more detail in our paper [Yin et al., 2001].

3. 2D Hand Posture Recognition

Hand segmentation is followed by feature extraction. Contour is the commonly used feature for accurate recognition of hand postures, and can be extracted easily from the silhouette of the segmented hand region. Segen and Kumar [Segen and Kumar, 1998] extracted the points along the boundary where the curvature reaches a local extremum as 'local features', and used those features that 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.

In our 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 RCE neural network is not trained sufficiently. The topological features of the hand, such as the number and positions of fingers, are other distinctive features of hand postures. In this section, we present a new method for accurate recognition of hand postures, which extract topological features of the hand from the silhouette of the segmented hand region, and recognize hand postures on the basis of the analysis of these features.

3.1 Feature 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 mass center 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 $E = \{P_i, i = 0,1,2,..., n\}$ that have the transition either from pixel value 0 to 1, or 1 to 0 along the circle;
- 4. Delete P_i and P_{i-1} , if the distance between two conjoint points $D = |P_i P_{i-1}| < \text{threshold } \lambda_d$;
- 5. Increment the radius r and iterate Step 2 to 4, until $r > 1/2$ (the width of the hand region).



Figure 4. Hand segmentation results

The purpose of Step 4 is to remove the falsely detected edge points resulted from imperfect segmentation. This step can removal most of falsely detected edge points. However, there are still occasions that one finger is divided into several branches because there are big holes in the image, or several fingers are merged into one branch because these fingers are too close. So we define the branch as follows:

Definition 3.1 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 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.

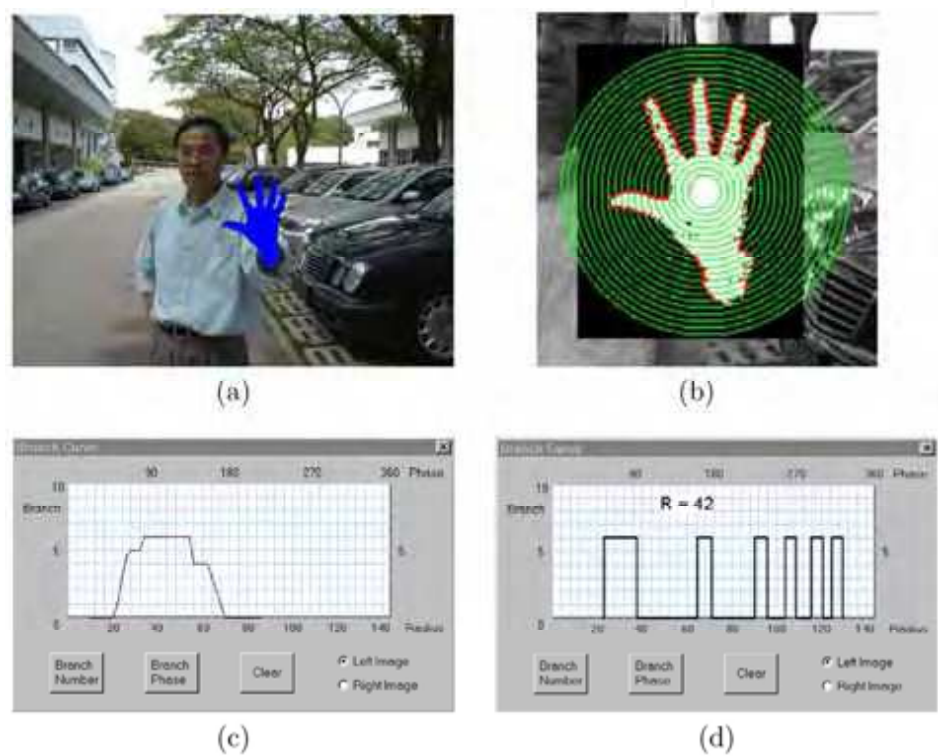


Figure 5. (a) segmented hand image, (b) Feature points extracted from the binary image of the segmented hand region, (c) Plot of branch number of the hand posture vs the radius of the search circle, (d) Plot of branch phase of the hand posture on the selected search circle

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 arm. Fig. 5 (a) shows a segmented hand image. Fig. 5(b) shows the part of Fig. 5(a) with the scale of 200%, in that the green circles represent the search circles and the red points represent the extracted feature points.

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 our method, we define the following function to determine the possibility p_i of each branch number:

$$p_i = a_i * \frac{C_i}{N}, i = 0, 1, 2, \dots, 6 \quad (8)$$

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 $a_1 < a_2 < \dots < a_i < \dots < a_6$, 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 possibility 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 \mathbf{K} (a set) whose occurrences are bigger than a threshold λ_n .
2. Choose the biggest one as the branch number BN among the numbers in \mathbf{K} , The biggest number in \mathbf{K} , but not the number with the most occurrence, is selected as BN , because the biggest number may not have the most occurrence if there are some search circles beyond the fingers. But when its occurrence is bigger than the threshold, it should be the most possible branch number. For example, Fig. 5(c) shows the relationship between the branch number and the radius of the search circle. In this case, branch number 5 occurs 7 times, and 0 occurs 15 times. However, we select 5 but not 0 as BN . This method is easier to implement, and is very effective and reliable with the threshold λ_n selected to be 6 in our implementation.

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

Definition 3.2 *The branch phase is the positions of the detected branches on the search circle, described by angle.*

In our method, we selected the middle one of the search circles, on which there are BN branches, to obtain the branch phase. Fig. 5 (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 helpful for improvement of the binary image of the segmented hand region, but the branch number and phase obtained from the improved image are the same as those obtained from the original one. It indicates that our 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.

3.2 Posture Recognition

After the branch phase is determined, the width of each branch BW_i can be obtained easily from the branch phase. In most cases, the widest branch should be the arm. We use it as the base branch BQ . Then the distance from other branch B^\wedge to BQ can be calculated, that is just the distance between the finger and the arm BD^\wedge . Using these aforementioned parameters: the branch number BN , the width of the branch BW_i , the distance between the finger and the arm BD^\wedge , the hand posture can be recognized accurately.

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

The postures shown in Fig. 6 all have distinctive features and are easy to recognize. We have used them for gesture-based robot programming and human-robot interaction of a real humanoid robot. The classification criterion of these postures is shown in Fig. 7. Preliminary experiments were conducted with users of different age, gender and skin color. The robot successfully recognized postures 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 or lighting condition changes too much, because the previous training of the RCE network becomes 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. In addition, the proposed posture recognition algorithm is only invariant to the hand rotation on the palm plane. If the hand is rotated more than 10 degree on the plane perpendicular to the palm, the posture recognition may be failed. The algorithms for topological feature extraction and hand posture recognition are described in more detail in our paper [Yin and Xie, 2007].

4. 3D Hand Posture Reconstruction

All of the 3D hand models employed so far use 3D kinematic models of the hand. Two sets of parameters are used in such models: angular (joint angles) and linear (phalange lengths and palm dimensions). However, The estimation of these kinematic parameters is a complex and cumbersome task because the human hand is an articulated structure with more than 27 degree of freedom. In this section, we propose to infer 3D information of the hand from the images taken from different viewpoints and reconstruct hand gestures using 3D reconstruction techniques.

4.1 Find robust matches

There are two approaches that can be used for the reconstruction of 3D vision models of hand postures. The first is to use calibrated stereo cameras, and the second is to use uncalibrated cameras. Camera calibration requires expensive calibration apparatus and elaborate procedures, and is only valid for the space near the position of the calibration object. Furthermore, the variation of focal lengths or relative positions of cameras will cause the previous calibration invalid. These drawbacks make camera calibration not feasible for gesture-based interaction, especially for human-robot interaction. Because service robots usually operate in dynamic and unstructured environments and their cameras need to be adjusted to track human hands frequently.

With uncalibrated stereo, there is an equivalence to the epipolar geometry which is presented by the fundamental matrix [Luong and Faugeras, 1996]. We have proposed a new method to estimate the fundamental matrix from uncalibrated stereo hand images. The proposed method consists of the following major steps: extracting points of interest; matching a set of at least 8 points; recovering the fundamental matrix.

In most approaches reported in the literature, high curvature points are extracted as points of interest. In our method, we use the edge points of the extended fingers, which are similar to those described in Section 3, as points of interest, and find robust matches from these points.

Matching different images of a single scene remains one of the bottlenecks in computer vision. A large amount of work has been carried out during the last decades, but the results are not satisfactory. The numerous algorithms for image matching that have been proposed can roughly be classified into two categories: correlation-based matching and feature-based matching. Correlation-based methods are not robust for hand image matching due to the ambiguity caused by the similar color of the hand. The topological features of the hand, such as the number and positions of the extended fingers that are described in the above section, are more distinct and stable in stereo hand images, only if the distance and angles between two cameras are not too big. In our method, we propose to take advantage of the topological features of the hand to establish robust correspondences between two perspective hand images.

We first detect fingertips by searching the furthest edge points from the mass center of the hand in the range between $B_i + BW_i$ and $B_i - BW_i$. Here B_i is the branch phase and BW_i is the branch width. The fingertips of two perspective hand images are found using this method, respectively. Simultaneously, their correspondences are established by the order of the finger. For example, the fingertip of B_1^r in the right image corresponds to the fingertip of B_1^l in the left image.

Then, we define the center of the palm as the point whose distance to the closest region boundary is maximum, and use the morphological erosion operation to find it. The procedure is as follows:

1. Apply dilation operation once to the segmented hand region.
2. Apply erosion operations until the area of the region becomes small enough. As a result, a small region at the center of the palm is obtained.
3. Calculate the center of mass of the resulting region as the center of the palm.

The purpose of the first step is to remove little holes in the imperfectly segmented hand image. These little holes can affect the result of erosion greatly. Fig. 8 shows the procedure to find the center of the palm by erosion operations.

The palm centers of two hand images are found by this method, respectively. In most case, they should correspond to each other because the shapes of the hand in two perspective images are almost the same under the assumption that the distance and angle between two cameras are small. However, because the corresponding centers of the palm are very critical for finding matches in our approach, we further use the following procedure to evaluate the accuracy of correspondence and determine the corresponding palm centers more robustly:

1. Find the fingertips T_i^l, T_i^r and the palm centers C^l, C^r for the left image and right image, respectively.
2. Calculate $d = \sum_1^{BN-1} |C^l T_i^l - C^r T_i^r|$. Here, $C^l T_i^l$ is the distance between the palm center and a fingertip in the left image, and $C^r T_i^r$ is that in the right image. $(BN - 1)$ represents the number of the extended fingers.
3. Take C^l and C^r as the corresponding palm centers if $d < \lambda_c$. λ_c is the threshold and is set to 2 pixels in our implementation.

The evaluation procedure above is used because we can assume $C^l T_i^l$ is equal to $C^r T_i^r$ according to projective invariance. If $d > \lambda_c$ we take the point, whose distance to each fingertip in the right image is the same as the distance between the palm center and each fingertip in the left image, as new C^r corresponding to C^l . Such a point is determined in theory by calculating the intersection of all the circles that are drawn in the right hand image

with the radius $C^lT_i^l$ at the positions of T_i^r . Referring to the coordinates of this point as x and y , they satisfy the following equation:

$$\min_{(x,y)} \sum_{i=1}^{BN-1} (x - (T_i^r)_x)^2 + (y - (T_i^r)_y)^2 - (C^lT_i^l)^2$$

(9)

where, $(T_i^r)_x$ and $(T_i^r)_y$ denote the coordinates of a fingertip in the right image. Such an equation is difficult to be solved by mathematical methods. In practice, we can determine an intersection within the right hand region for every two circles, then calculate the mass center of all the intersections as new C^r .

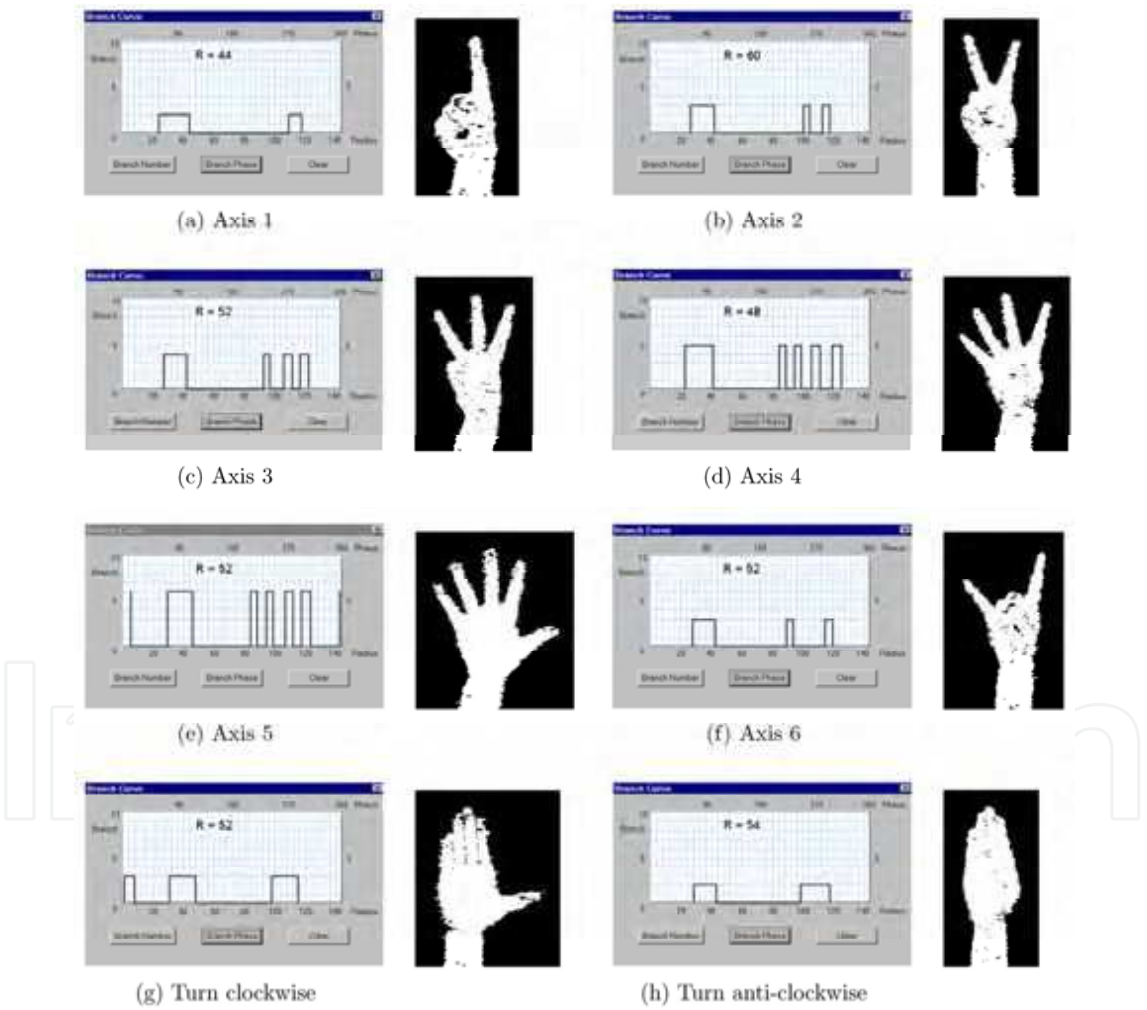


Figure 6. Hand postures used for robot programming and human-robot interaction

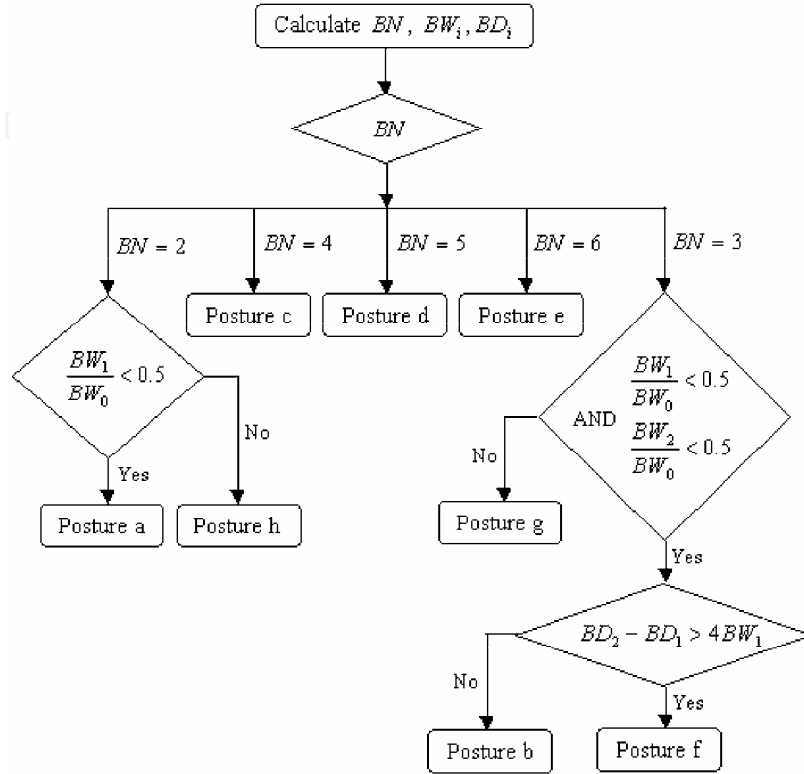


Figure 7. Classification criterion of hand postures



Figure 8. Procedure for finding the center of the palm by the morphological erosion operation

After the corresponding palm centers are determined, matches can be found by comparing the edge points on the i^{th} ($i = 1, \dots, m$) search circle of the left image with those of the right image. The criterion is as follows:

1. Calculate $d_a = |P_{i,j}^l P_{i,j-1}^l - P_{i,j}^r P_{i,j-1}^r|$ and $d_b = |P_{i,j}^l P_{i,j+1}^l - P_{i,j}^r P_{i,j+1}^r|$. Here, $P_{i,j}^l$ is the j^{th} edge point on the i^{th} search circle in the left image, and $P_{i,j}^r$ is that in the right image. $P_{i,j}^l P_{i,j-1}^l$ is the distance between the edge points $P_{i,j}^l$ and $P_{i,j-1}^l$.
2. Calculate $d = (d_a + d_b)/2$. If $d < \text{threshold } \lambda_m$, $P_{i,j}^l$ and $P_{i,j}^r$ are taken as a pair of matches. λ_m is set to 2 pixels in our implementation.

The basic idea underlying this matching algorithm is to extract the edge points, whose distances to its previous and following points as well as to the center of the palm are almost identical in two images, as matches. The algorithm works very well under the situation that the distance and angle between two cameras are small. In Fig. 9, (a) shows the edge points extracted from the segmented hand regions of two perspective images and (b) shows the matches extracted from these edge points. The green circles represent the search circles and the red points are the extracted edge points.

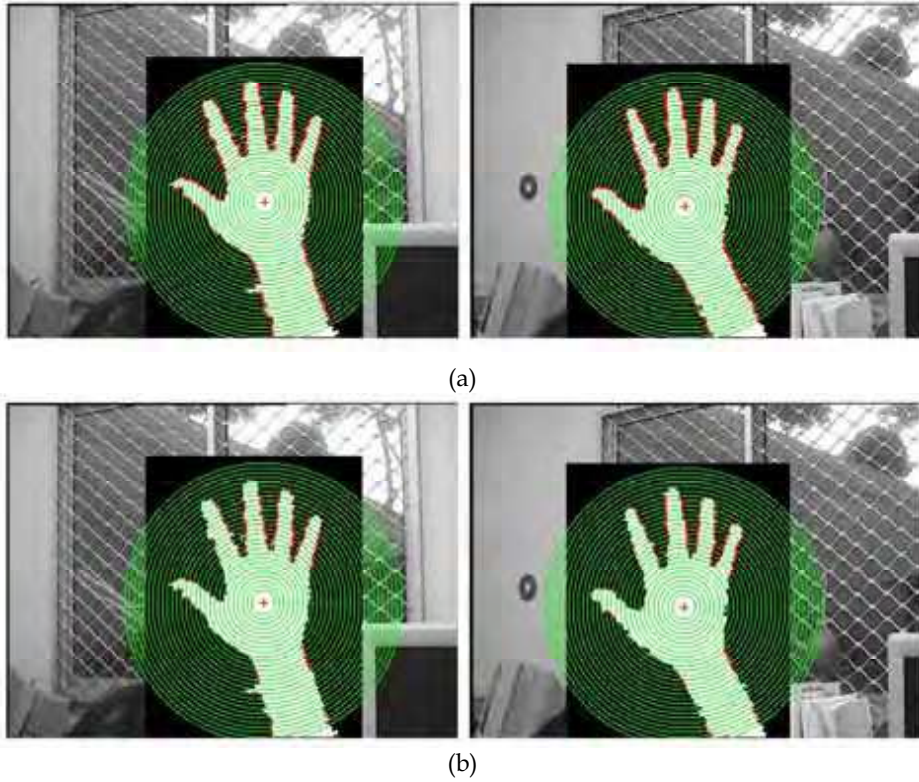


Figure 9. (a) Edge points extracted from stereo hand images, (b) Matches extracted from edge points

4.2 Estimate the Fundamental Matrix

Using the set of matched points established in the previous step, the epipolar geometry between two uncalibrated hand images can be recovered. It contains all geometric information that is necessary for establishing correspondences between two perspective images, from which 3D structure of an object can be inferred.

The epipolar geometry is the basic constraint which arises from the existence of two viewpoints [Faugeras, 1993]. Considering the case of two cameras, we have the following fundamental equation:

$$(\tilde{\mathbf{m}}^r)^T \mathbf{F} \tilde{\mathbf{m}}^l = 0 \quad (10)$$

where $\tilde{\mathbf{m}}^l = [u^l, v^l, 1]^T$ and $\tilde{\mathbf{m}}^r = [u^r, v^r, 1]^T$ are the homogeneous image coordinates of a 3D point in the left and right images, respectively. \mathbf{F} is known as the *fundamental matrix*. Geometrically, $\mathbf{F}\tilde{\mathbf{m}}^l$ defines the epipolar line of a left image point \mathbf{m}^l in the right image. Equation (10) says no more than that the correspondence in the right image of point \mathbf{m}^l lies on the corresponding epipolar line. Transposing equation (10) yields the symmetric relation from the right image to the left image.

\mathbf{F} is of rank 2. Besides, it is defined up to a scalar factor. Therefore, a fundamental matrix has only seven degrees of freedom. That is, there are only 7 independent parameters among the 9 elements of the fundamental matrix. Various techniques have been reported in the literature for estimation of the fundamental matrix (see [Zhang, 1996] for a review). The classical method for computing the fundamental matrix from a set of 8 or more point matches is the 8-point algorithm introduced by Longuet-Higgins in [Longuet-Higgins, 1981]. This method is the linear criterion and has the advantage of simplicity of implementation. However, it is quite sensitive to noise. In order to recover the epipolar geometry as accurately as possible, we use a combination of techniques such as input data normalization, rank-2 constraint, linear criterion, nonlinear criterion as well as robust estimator to yield an optimal estimation of the fundamental matrix. The algorithm is as follows:

1. Normalize pixel coordinates of matches.
2. Initialize the weights $\omega_i = 1$ and $\gamma_i = 1$ for all matches.
3. For a number of iterations:
 - 3.1. Weight the i^{th} linear equation by multiplying it by $\omega_i\sqrt{\gamma_i}$.
 - 3.2. Estimate the fundamental matrix \mathbf{F} using the linear least-squares algorithm.
 - 3.3. Impose the rank-2 constraint to the estimated \mathbf{F} by the singular value decomposition.
 - 3.4. Calculate the residuals of matches $r_i = (\tilde{\mathbf{m}}_i^r)^T \mathbf{F} \tilde{\mathbf{m}}_i^l$.
 - 3.5. Calculate the nonlinear method weight:

$$\omega_i = \left(\frac{1}{(l_{i1}^l)^2 + (l_{i2}^l)^2} + \frac{1}{(l_{i1}^r)^2 + (l_{i2}^r)^2} \right)^{1/2} \quad (11)$$

here $\mathbf{l}_i^r = \mathbf{F}\tilde{\mathbf{m}}_i^l = [l_{i1}^r, l_{i2}^r, l_{i3}^r]^T$ is the corresponding epipolar line of point \mathbf{m}_i^l and $\mathbf{l}_i^l = \mathbf{F}^T \tilde{\mathbf{m}}_i^r = [l_{i1}^l, l_{i2}^l, l_{i3}^l]^T$ the corresponding epipolar line of point \mathbf{m}_i^r .

- 3.6. Calculate the distances between matching points and the corresponding epipolar lines $d_i = \omega_i r_i$.
- 3.7. Calculate the robust method weight:

$$\gamma_i = \begin{cases} 1 & |d_i| \leq \sigma \\ \sigma/|d_i| & \sigma < |d_i| \leq 3\sigma \\ 0 & |d_i| > 3\sigma \end{cases} \quad (12)$$

By combining several simple methods together, the proposed approach becomes more effective and robust, but still easily to be implemented.

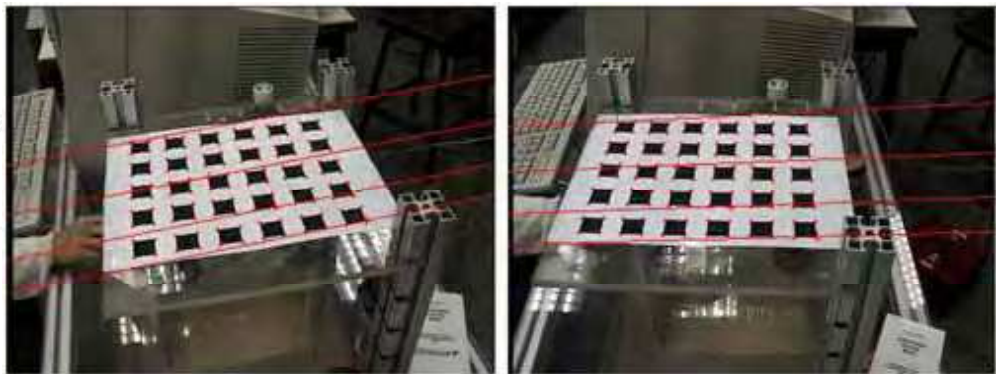


Figure 10. The epipolar geometry computed from the calibration matrices

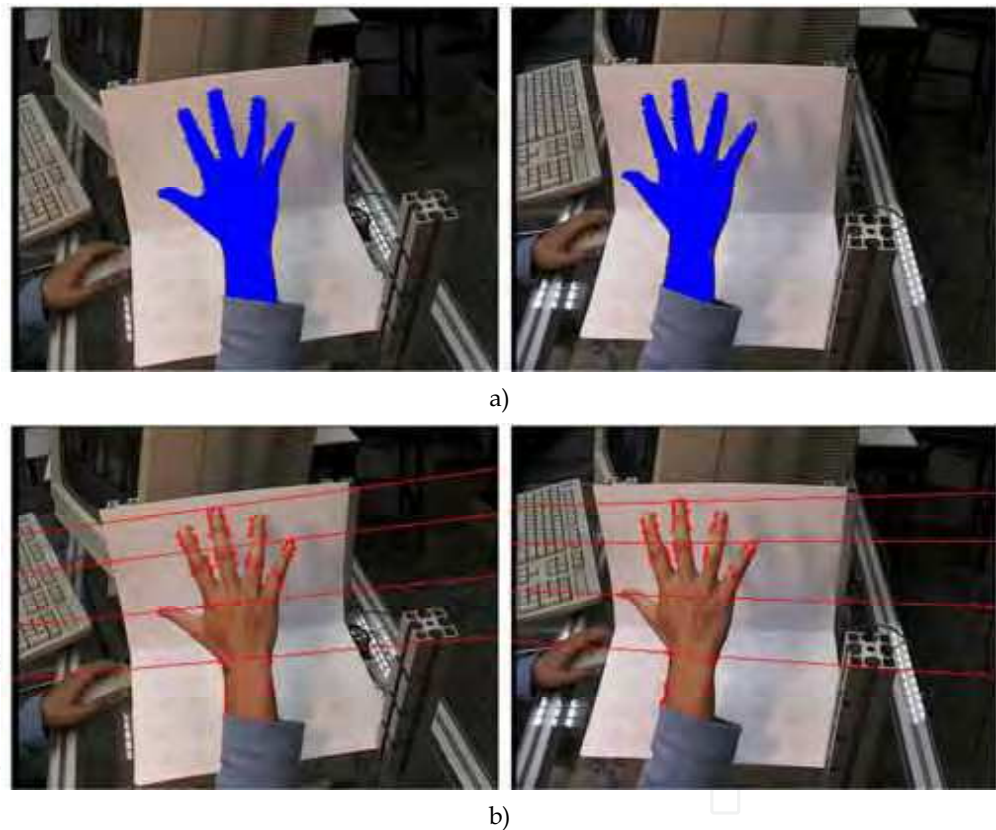


Figure 11. (a) Segmentation results of one pair of calibrated hand images, (b) Extracted matches and the estimated epipolar geometry

Our experimental results demonstrate the performance of the proposed algorithm. We first use this algorithm to estimate the fundamental matrix between two calibrated cameras, and compare the obtained epipolar geometry with that computed from the calibration matrices

of the cameras. The epipolar geometry computed from the calibration matrices is shown in Fig. 10. It serves as a ground truth. Fig. 11 shows a pair of hand images taken by the calibrated cameras with the size of 384×288 . In that, (a) shows the segmentation results of the hand images using the method presented in Section 2, and (b) shows the extracted corresponding points using the approach presented in Section 3 as well as the epipolar geometry estimated from these matches using the algorithm described in this section.

Sometimes, the matches extracted from the hand images may lie on a plane. This will cause degeneracy in the data, and affect the accuracy of the estimation of the fundamental matrix. We can take more hand images with the hand at different positions and use all the matches extracted from these images to get a more accurate estimation of the fundamental matrix. The epipolar geometry estimated using all the matches obtained from several hand images is shown in Fig. 12. The red solid lines represent the epipolar lines estimated from the extracted matches, and the green dash lines represent those computed from the calibration matrices. It can be observed the estimated epipolar geometry is very closed to the calibrated one.

Fig. 13 shows a pair of hand images taken by two uncalibrated cameras with the size of 384×288 . In that, (a) shows the segmentation results of the hand images and (b) shows the extracted corresponding points as well as the epipolar geometry estimated from these matches. In order to avoid the problem of degeneracy, and obtain more accurate and robust estimation of the fundamental matrix, we take more than one pairs of hand images with the hand at different positions, and use all the matches found in these images to estimate the fundamental matrix. Fig. 14 shows another pair of images taken by the same cameras, where the epipolar geometry is estimated from all the matches obtained from several hand images. It can be observed that the estimated epipolar lines match the corresponding points well even though there is no point in this figure used for the estimation of the fundamental matrix. So at the beginning of hand gesture recognition, we can take several hand images with the hand at different positions, and use the matches extracted from these images to recovery the epipolar geometry of the uncalibrated cameras. Then the recovered epipolar geometry can be applied to match other hand images and reconstruct hand postures. If the parameters of the cameras change, the new fundamental matrix is easy to be estimated by taking some hand images again.

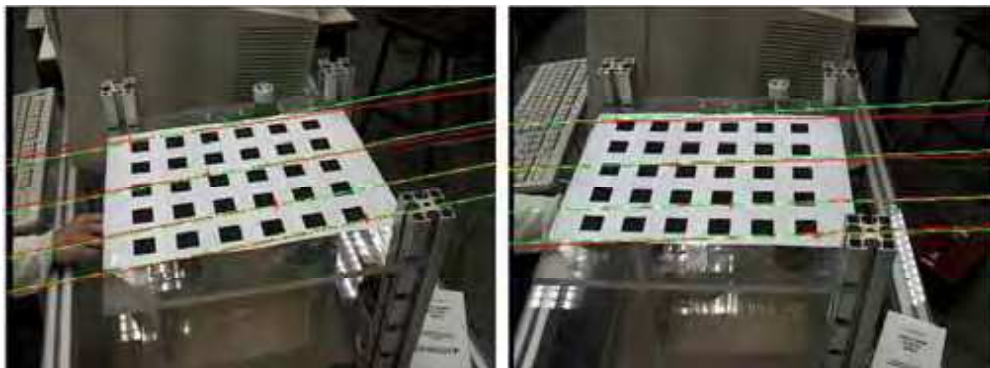


Figure 12. Comparison of the estimated epipolar geometry with the calibrated one

In [Zhang et al., 1995], Zhang proposed an approach to match images by exploiting the epipolar constraint. They extracted high curvature points as points of interest, and match

them using a classical correlation technique followed by a new fuzzy relaxation procedure. Then the fundamental matrix is estimated by using a robust method: the Least Median of Squares (Lmeds). Zhang provides a demo program to compute the epipolar geometry between two perspective images of a single scene using this method at his home page: <http://www.inria.fr/robotvis/personnel/zzhang/zzhang-eng.html>. We submitted the images in Figs. 13 and 14 to this program and obtain the results as shown in Figs. 15 and 16, where (a) shows the extracted correspondences which are marked by white crosses, and (b) shows the estimated epipolar geometry. It can be seen the epipolar lines are very far from the corresponding points on the hand.

The approach presented in this section can also be used for other practical applications. For example, at some occasions when the calibration apparatus is not available and the feature points of the scene, such as corners, are difficult to be extracted from the images, we can take advantage of our hands, and use the method presented above to derive the unknown epipolar geometry for the uncalibrated cameras. This method is described in more detail in our paper [Yin and Xie, 2003].

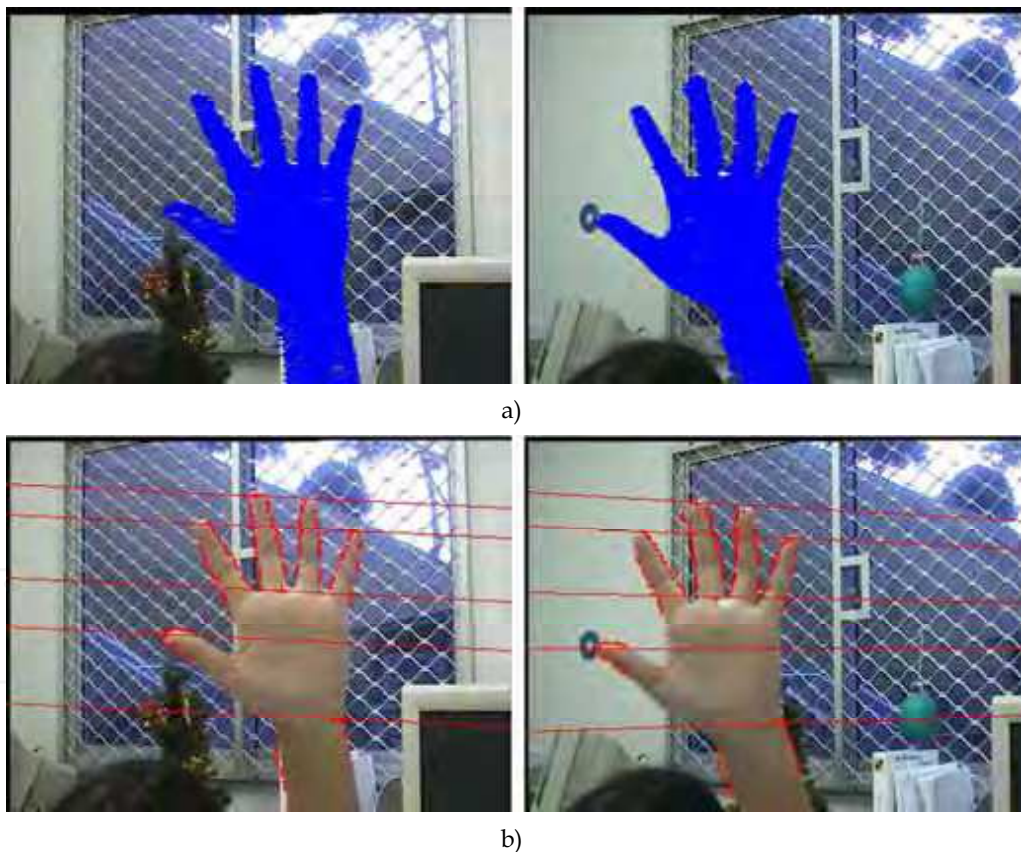


Figure 13. (a) Segmentation results of one pair of uncalibrated hand images, (b) Extracted matches and the estimated epipolar geometry

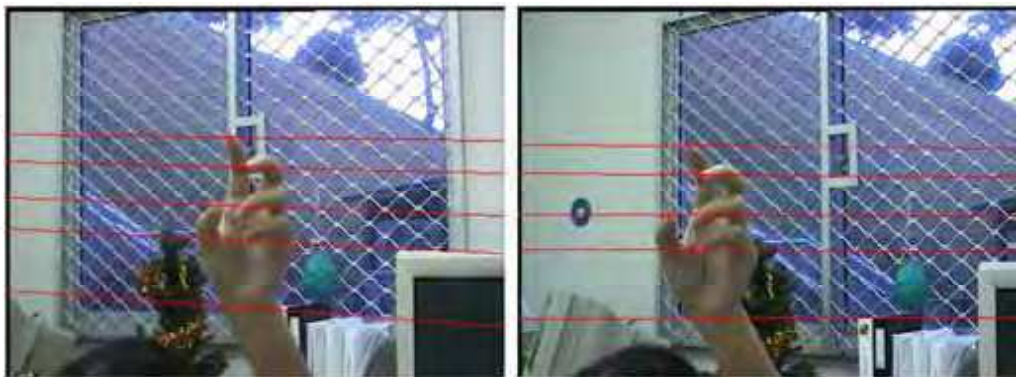
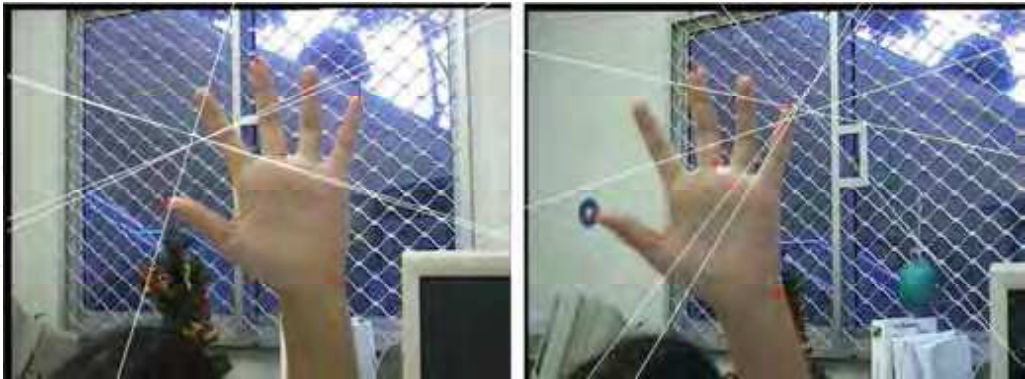


Figure 14. Application of the estimated epipolar geometry to one pair of uncalibrated hand images



a)



b)

Figure 15. (a) Extracted matches using the method proposed by Zhang from uncalibrated hand images shown in Fig. 13, (b) Estimated epipolar geometry from these matches

4.3 Reconstruct hand postures

After the epipolar geometry between two uncalibrated cameras are recovered, it can be applied to match other hand images and reconstruct 3D hand postures. Although stereo images taken by uncalibrated cameras allow reconstruction of 3D structure only up to a projective transformation, it is sufficient for hand gesture recognition, where the shape of the hand, not the scale, is important.

The epipolar geometry is the basic constraint which arises from the existence of two viewpoints. For a given point in one image, its corresponding point in the other image must lie on its epipolar line. This is known as the *epipolar constraint*. It establishes a mapping between points in the left image and lines in the right image and vice versa. So, if we determine the epipolar line $l_{m^l}^r$ in the right image for a point m^l in the left image, we can restrict the search for the match of m^l along $l_{m^l}^r$. The search for correspondences is thus reduced to a 1D problem.

After the set of matching candidates \mathbf{m}^r is obtained, the correct match of m^l in the right image, denoted by m^r , is further determined using correlation-based method. In correlation-based methods, the elements to match are image windows of fixed size, and the similarity criterion is a measure of correlation between windows in two images. The corresponding element is given by the window that maximizes the similarity criterion within a search region. For intensity images, the following cross-correlation is usually used [Faugeras, 1993]:

$$C(u^l, v^l, u^r, v^r) = \frac{1}{K\sigma^l(u^l, v^l)\sigma^r(u^r, v^r)} \sum_{i=-n}^n \sum_{j=-m}^m \left[I^l(u^l + i, v^l + j) - \overline{I^l(u^l, v^l)} \right] \left[I^r(u^r + i, v^r + j) - \overline{I^r(u^r, v^r)} \right] \quad (13)$$

with

$$K = (2n + 1)(2m + 1) \quad (14)$$

$$\overline{I^l(u^l, v^l)} = \frac{1}{(2n + 1)(2m + 1)} \sum_{i=-n}^n \sum_{j=-m}^m I^l(u^l + i, v^l + j) \quad (15)$$

$$\sigma^l(u^l, v^l) = \sqrt{\frac{\sum_{i=-n}^n \sum_{j=-m}^m \left[I^l(u^l + i, v^l + j) - \overline{I^l(u^l, v^l)} \right]^2}{(2n + 1)(2m + 1)}} \quad (16)$$

where, I^l and I^r are the intensity functions of the left and right images. $\overline{I^l(u^l, v^l)}$ and $\sigma^l(u^l, v^l)$ are the mean intensity and standard deviation of the left image at the point (u^l, v^l) in the window $(2n + 1) \times (2m + 1)$. $\overline{I^r(u^r, v^r)}$ and $\sigma^r(u^r, v^r)$ are similar to $\overline{I^l(u^l, v^l)}$ and $\sigma^l(u^l, v^l)$, respectively. The correlation C ranges from -1 for two correlation windows which are not similar at all, to 1 for two correlation windows which are identical. However, this cross-correlation method is unsuitable for color images, because in color images, a pixel is represented by a combination of three primary color components (R (red), G (green), B (blue)). One combination of (R, G, B) corresponds to only one physical color, and a same intensity value may correspond to a wide range of color combinations. In our method, we use the following color distance based similarity function to establish correspondences between two color hand images [Xie, 1997].

$$C(u^l, v^l, u^r, v^r) = 1 - \frac{1}{Ks} \sum_{i=-n}^n \sum_{j=-m}^m (A^2 + B^2 + C^2) \quad (17)$$

with

$$K = (2n + 1)(2m + 1) \quad (18)$$

$$s = 3 \cdot 255^2 \quad (19)$$

$$A = R^l(u^l + i, v^l + j) - R^r(u^r + i, v^r + j) \quad (20)$$

$$B = G^l(u^l + i, v^l + j) - G^r(u^r + i, v^r + j) \quad (21)$$

$$C = B^l(u^l + i, v^l + j) - B^r(u^r + i, v^r + j) \quad (22)$$

where, R^l , G^l and B^l are the color values of the left image corresponding to red, green and blue color components, respectively. R^r , G^r and B^r are those of the right image.



a)



b)

Figure 16. (a) Extracted matches using the method proposed by Zhang from uncalibrated hand images shown in Fig. 14, (b) Estimated epipolar geometry from these matches

The similarity function defined in Equation (17) varies in the range $[0, 1]$. Then stereo matching can be summarized as follows: Given a pixel (u^l, v^l) in the left image, find a pixel (\hat{u}^r, \hat{v}^r) in the right image which maximizes the similarity function in Equation (17):

$$C(u^l, v^l, \hat{u}^r, \hat{v}^r) = \max_{(u^r, v^r) \in W} \{C(u^l, v^l, u^r, v^r)\} \quad (23)$$

where, W denotes the searching area in the right image. In our implementation, the searching area is limited in the segmented hand region and on the epipolar line.

The computation of C is time consuming because each pixel involves three multiplications. In practice, a good approximation is to use the following similarity function.

$$C(u^l, v^l, u^r, v^r) = \frac{1}{3}(C_R + C_G + C_B) \quad (24)$$

where

$$C_R = 1 - \frac{\sum_{i=-n}^n \sum_{j=-m}^m |R^l(u^l + i, v^l + j) - R^r(u^r + i, v^r + j)|}{255 \cdot (2n + 1)(2m + 1)} \quad (25)$$

$$C_G = 1 - \frac{\sum_{i=-n}^n \sum_{j=-m}^m |G^l(u^l + i, v^l + j) - G^r(u^r + i, v^r + j)|}{255 \cdot (2n + 1)(2m + 1)} \quad (26)$$

$$C_B = 1 - \frac{\sum_{i=-n}^n \sum_{j=-m}^m |B^l(u^l + i, v^l + j) - B^r(u^r + i, v^r + j)|}{255 \cdot (2n + 1)(2m + 1)} \quad (27)$$

The similarity function defined in Equation (24) also takes values in the range $[0, 1]$.

As shown in Figure 17, for the points marked by red crosses in the left image, their matching candidates in the right image found by the technique described above are marked by red points. Figure 18 shows all detected corresponding points of the hand, and Figure 19 shows 4 views of the reconstructed 3D hand posture.

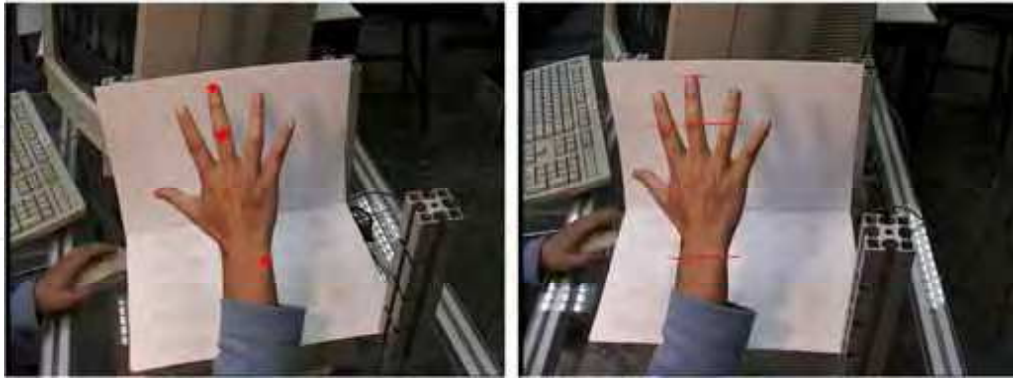
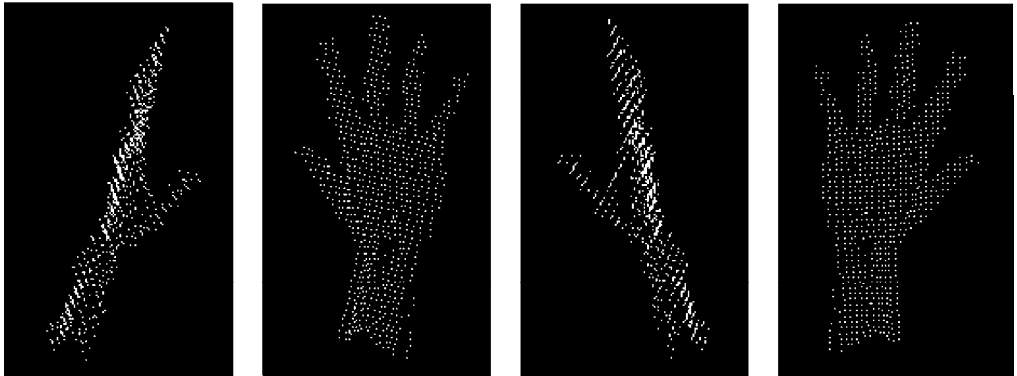


Figure 17. Find corresponding points in the right image which are marked by red points, for points in the left image which are marked by red crosses, using the color correlation and epipolar geometry



Figure 18. Detected corresponding points of the hand



(a) Right view (b) Front view (c) Left view (d) Back view

Figure 19. Different views of the reconstructed 3D hand posture

5. Gesture-Based Human-Robot Interaction

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, human-robot interaction etc., to build a multi-modal and intelligent human-robot interface. Fig. 20(a) shows the human-alike service robot HARO-1 at our lab. It was designed and developed by ourselves, and mainly consists of an active stereo vision head on modular neck, two modular arms with active links, an omnidirectional mobile base, dextrous hands under development and the computer system. Each modular arm has 3 serially connected active links with 6 axes, as shown in 20 (b).

5.1 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 [Sing and Ikeuchi, 1997]. However, 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.

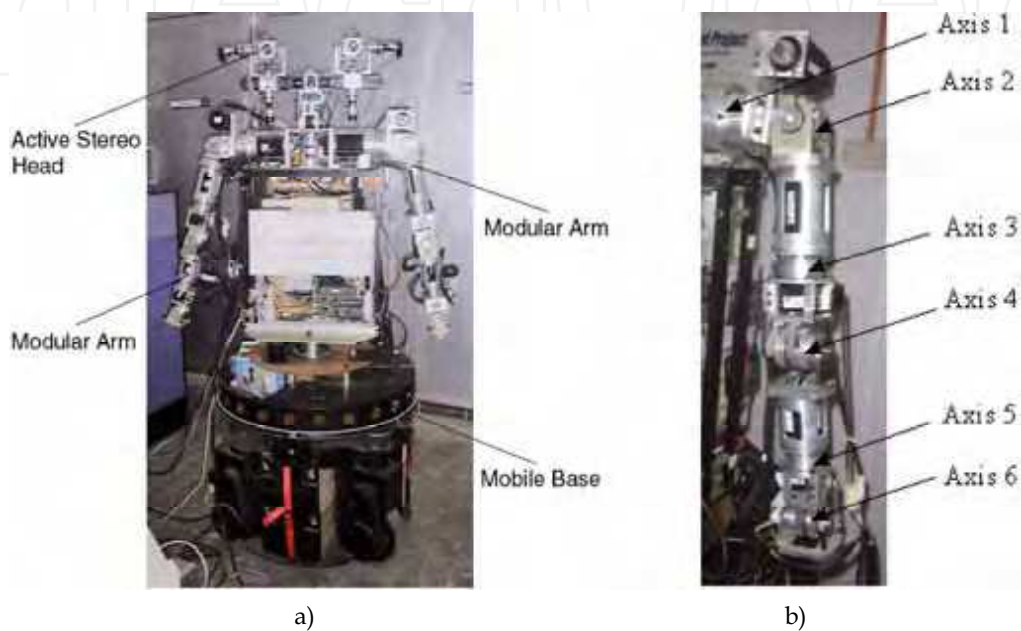


Figure 20. (a) Humanoid service robot HARO-1; (b) Modular robot arm with 6 axes

Based on our approach of 2D hand posture recognition, we have proposed a posture programming method for our service robot. In this method, we define task postures and corresponding motion postures respectively, and associate them during the training procedure, so that the robot will perform all the motions associated with a task if that task posture 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 postures easily and efficiently.

The postures shown in Fig. 6 is used for both robot programming and human-robot interaction. In the programming mode, Postures *a* to *f* represent the six axes of the robot arm respectively, Posture *g* means 'turn clockwise', and Posture *h* means 'turn anti-clockwise'. We use them as motion gestures to control the movements of the six axes of either robot arm. Using these postures, we can guide the robot arm to do any motion, and record any motion sequence as a task.

In the interaction mode, these postures are redefined as task postures and associated with corresponding tasks. For example, some motion sequence is defined as Task 1, and is associated with Posture *a*. When Posture *a* is presented to the robot in the interaction stage, the robot will move its arm according to the predefined motion sequence. A task posture is easy to be associated with different motion sequences in different applications by programming using corresponding motion postures.

5.2 Gesture-Based Interaction System

Fig. 21 shows the Graphic User Interface (GUI) of the gesture-based interaction system implemented on robot HARO-1, in that (a) represents the Vision section of the interface, and (b) shows the virtual robot developed using Open GL.

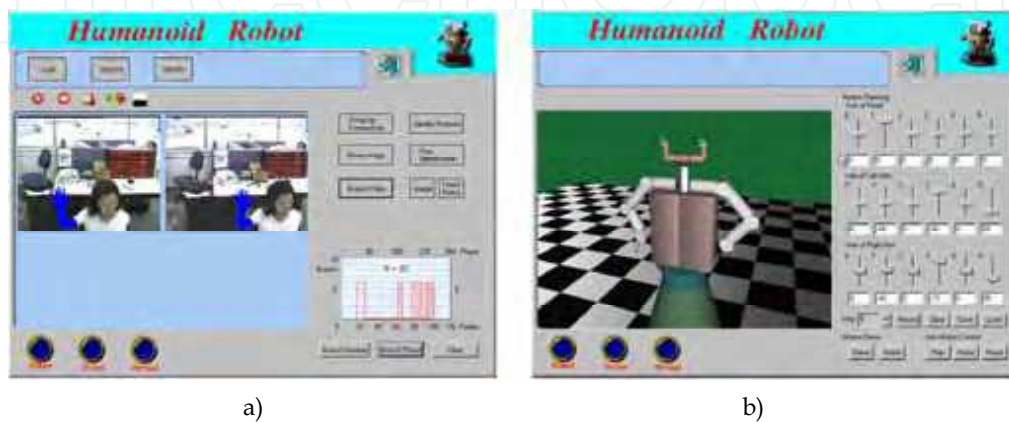


Figure 21. Graphic user interface of the robot HARO-1: (a) Posture recognition; (b) Virtual robot

As shown in Fig. 21 (a), live images with the size of 384x288 are captured through two CCD video cameras (EVID31, SONY) in the system. At the end of each video field the system processes the pair of images, and output the detected hand information. 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 about 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 4-6Hz 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. 6, the robot will perform motions associated with this posture. If the segmented image can not be recognized because of the presence 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.

6. Conclusions

Vision-based hand gesture recognition provide a more nature and powerful way for human-computer interaction. In the chapter, we present some new approaches for hand image segmentation, 2D hand posture recognition and 3D hand posture reconstruction. We segment hand images using the color segmentation approach which is based on the RCE neural network. Then we extract topological features of the hand from the binary image of the segmented hand region, and recognize 2D hand postures base on the analysis of these features. We also propose to use the stereo vision and 3D reconstruction techniques to recover 3D hand postures and present a new method to estimate the fundamental matrix from uncalibrated stereo hand images in this chapter. A human-robot interaction system has been developed to demonstrate the application of our hand posture recognition approaches.

7. References

- Ahmad, T., Taylor, C. J., Lanitis, A., and Cootes, T. F. (1997). Tracking and recognizing hand gestures using statistical shape models. *Image and Vision Computing*, 15:345-352. [Ahmad et al., 1997]
- Faugeras, O. (1993). *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, Massachusetts. [Faugeras, 1993]
- Freeman, W. T. and Weissman, C. D. (1995). Television control by hand gestures. In *Proceedings of International Workshop on Automatic Face and Gesture Recognition*, pages 179-183, Zurich, Switzerland. [Freeman and Weissman, 1995]
- Jones, M. J. and Rehg, J. M. (1999). Statistical color models with application to skin detection. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 274-280, Fort Collins, CO. [Jones and Rehg, 1999]
- Kahn, R. E., Swain, M. J., Prokopowicz, P. N., and Firby, R. J. (1996). Gesture recognition using the perseus architecture. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 734–741, San Francisco. [Kahn et al., 1996]
- Kasson, J. K. and Plouffe, W. (1992). A analysis of selected computer interchange color spaces. *ACM Transaction on Graphics*, 11(4):373-405. [Kasson and Plouffe, 1992]
- Kjeldsen, R. and Render, J. (1996a). Finding skin in color images. In *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pages 312–317, Killington, Vt. [Kjeldsen and Render, 1996a]
- Kjeldsen, R. and Render, J. (1996b). Toward the use of gesture in traditional user interfaces. In *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pages 151-156, Killington, Vt. [Kjeldsen and Render, 1996b]
- Krueger, M. W. (1991). *Artificial Reality H*. Addison-Wesley. [Krueger, 1991]
- Lathuiliere, F. and Herve, J.-Y. (2000). Visual hand posture tracking in a gripper guiding application. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1688-1694, San Francisco, CA. [Lathuiliere and Herve, 2000]
- Lee, J. and Kunii, T. L. (1995). Model-based analysis of hand posture. *IEEE Transactions on Computer Graphics and Application*, 15(5):77–86. [Lee and Kunii, 1995]
- Longuet-Higgins, H. C. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133-135. [Longuet-Higgins, 1981]

- Luong, Q.-T. and Faugeras, O. D. (1996). The fundamental matrix: Theory, algorithms and stability analysis. *International Journal of Computer Vision*, 1(17):43–76. [Luong and Faugeras, 1996]
- Maggioni, C. (1995). Gesturecomputer - new ways of operation a computer. In *Proceedings of International Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland. [Maggioni, 1995]
- Pavlovic, V. I., Sharma, R., and Huang, T. S. (1996). Gestural interface to a visual computing environment for molecular biologists. In *Proceedings of International Conference on Face and Gesture Recognition*, pages 30-35, Killington, Vt. [Pavlovic et al., 1996]
- Pavlovic, V. L., Sharma, R., and Huang, T. S. (1997). Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677-695. [Pavlovic et al., 1997]
- Quek, F. K. H., Mysliviec, T., and Zhao, M. (1995). Finger mouse: A freehand pointing interface. In *Proceedings of International Workshop on Automatic Face and Gesture Recognition*, pages 372-377, Zurich, Switzerland. [Quek et al., 1995]
- Reilly, D. L., Cooper, L. N., and Elbaum, C. (1982). A neural network mode for category leaning. *Biological Cybernetics*, 45:35-41. [Reilly et al., 1982]
- Segen, J. and Kumar, S. (1998). Fast and accurate 3d gesture recognition interface. In *Proceedings of International Conference on Pattern Recognition*, pages 86–91, Brisbane, Australia. [Segen and Kumar, 1998]
- Sing, B. K. and Ikeuchi, K. (1997). Toward automatic robot instruction from perception—mapping human grasps to manipulator grasps. *IEEE Transaction on Robotics and Automation*, 13(1):81-95. [Sing and Ikeuchi, 1997]
- Triesch, J. and Malsburg, C. V. D. (1998). A gesture interface for human-robot interaction. In *Proceedings of 3rd IEEE International Conference on Automatic Face and Gesture Recognition*, pages 546-551. [Triesch and Malsburg, 1998]
- Xie, M. (1997). Automatic feature matching in uncalibrated stereo vision through the use of color. *Robotic and Autonomous System*, 21(4):355-364. [Xie, 1997]
- Yang, J., Lu, W., and Waibel, A. (1998). Skin-color modeling and adaptation. In *Proceedings of ACCV98*, pages 687-694, Hong Kong. [Yang et al., 1998]
- Yin, X., Guo, D., and Xie, M. (2001). Hand image segmentation using color and rce neural network. *International journal of Robotics and Autonomous System*, 34(4):235-250. [Yin et al., 2001]
- Yin, X. and Xie, M. (2003). Estimation of the fundamental matrix from uncalibrated stereo hand images for 3d hand gesture recognition. *Pattern Recognition*, 36(3):23-40. [Yin and Xie, 2003]
- Yin, X. and Xie, M. (2007). Finger identification and hand posture recognition for human-robot interaction. *Image and Vision Computing*, 25(8):1291-1300. [Yin and Xie, 2007]
- Zhang, Z. (1996). Determining the epipolar geometry and its uncertainty: A review. Technical Report 2927, INRIA Sophia-Antipolis, France. [Zhang, 1996]
- Zhang, Z., Deriche, R., Faugeras, O., and Luong, Q.-T. (1995). A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, 78:87-119. [Zhang et al., 1995]



Human Robot Interaction

Edited by Nilanjan Sarkar

ISBN 978-3-902613-13-4

Hard cover, 522 pages

Publisher I-Tech Education and Publishing

Published online 01, September, 2007

Published in print edition September, 2007

Human-robot interaction research is diverse and covers a wide range of topics. All aspects of human factors and robotics are within the purview of HRI research so far as they provide insight into how to improve our understanding in developing effective tools, protocols, and systems to enhance HRI. For example, a significant research effort is being devoted to designing human-robot interface that makes it easier for the people to interact with robots. HRI is an extremely active research field where new and important work is being published at a fast pace. It is neither possible nor is it our intention to cover every important work in this important research field in one volume. However, we believe that HRI as a research field has matured enough to merit a compilation of the outstanding work in the field in the form of a book. This book, which presents outstanding work from the leading HRI researchers covering a wide spectrum of topics, is an effort to capture and present some of the important contributions in HRI in one volume. We hope that this book will benefit both experts and novice and provide a thorough understanding of the exciting field of HRI.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Xiaoming Yin and Ming Xie (2007). Hand Posture Segmentation, Recognition and Application for Human-Robot Interaction, Human Robot Interaction, Nilanjan Sarkar (Ed.), ISBN: 978-3-902613-13-4, InTech, Available from:

http://www.intechopen.com/books/human_robot_interaction/hand_posture_segmentation__recognition_and_application_for_human-robot_interaction

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2007 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen