

Hand Posture Recognition Using Finger Geometric Feature

Liwei Liu¹, Junliang Xing¹, Haizhou Ai¹, Xiang Ruan²

¹Computer Science and Technology Department, Tsinghua University, Beijing, 100084, China

²OMRON Corporation, Kusatsu-city, SHIGA, 525-0025, JAPAN

E-mail:ahz@mail.tsinghua.edu.cn

Abstract

Hand posture recognition (HPR) plays an important role in human-computer interaction (HCI) since it is one of the most common and natural ways of communication among human beings. Different fingers often represent different meanings which will attract more attentions in HPR research. Based on finger geometric feature and its classification, we develop a HPR system that can tell its posture on possible fingers. We explore kinematic constraints of the hand with forearm to extract finger geometric features which are translation, rotation and scale invariant. We first search hand components with the help of skeleton, and then order them into a serial arrangement according to either left hand or right hand and extract the geometric features among fingers, palm and forearm, finally those features are used in SVM classification for HPR. Our method can recognize twelve different types of hand postures for both hands respectively. Experiments under different illumination conditions and different scenes demonstrate the effectiveness and efficiency of the proposed method.

1 Introduction

Hand posture recognition (HPR) has attracted a lot of attentions in human-computer interaction (HCI) since it can be used to communicate with computer as an input device instead of traditional devices. One-to-one correspondence relationship can then be set up between functions and hand postures to facilitate our daily life and work. There are many potential applications of HPR such as device controls for domestic appliances like TV, the user's interaction with PC in computer games, as well as in a sign language system.

HPR has been researched for many years and significant advances have been achieved, but few can be used in practical applications due to limitations in robustness and speed. Most existing works of HPR can be categorized into glove-based methods and pure vision-based

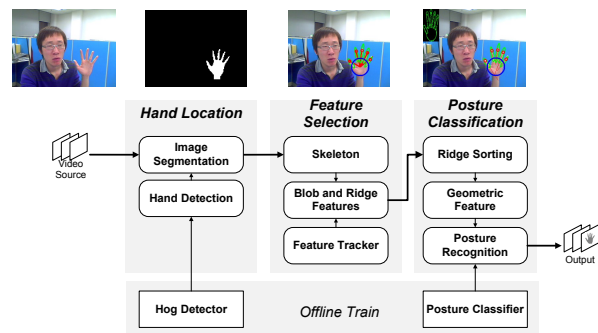


Figure 1: Overview of our approach.

methods.

As a state-of-the-art glove-based method, Wang and Popovic [8] used color gloves to achieve real-time hand-tracking and posture recognition. Their approach used a single camera to track a hand wearing and used Hausdorff-like distance as a metric to find the nearest neighbor in a database for posture recognition. The system is real time and robust in indoor scenes. Using color glove, however, sacrifices the flexibility of HPR and limits its generalization.

A pure vision-based method [5][4] used a Viola-Jones detector [7] to detect six types of posture which can tolerate rotation up to 15° . However, the detector cannot manage rotation in all range and the types of posture did not cover the postures people popularly used. Lindeberg et al. [2][6] proposed multi-scale color features for HPR. We adopt these features in our work. In their approach, a hierarchical hand model that approximates the relative scales and fixes the relative positions between features was combined to pursue HPR, which might lack of flexibility. In order to deal with this problem, we propose new methods to find hand components and to do HPR which can handle more types of postures for both hands respectively very fast and flexibly.

The contributions of our novel approach lie in: (1) fingers, palm and forearm are integrated and relative geometric features are extracted to explore the kinematic constraints for HPR; (2) skeleton is used to guide

the search for hand components represented by blobs and ridges, which greatly improves the robustness and the efficiency; (3) geometric features which are translation, rotation and scale invariant, are extracted for SVM classification for HPR that can tell the corresponding information on possible fingers.

2 THE PROPOSED APPROACH

An overview of our approach are shown in Fig. 1. Given an input image, an AdaBoosted detector locates the hand in skin connected regions from segmentation. And then the skeleton of the hand is extracted to help searching for the optimal combination of hand components. At last, the geometric features of hand components are computed to pursue posture classification.

2.1 Hand Location

First, we develop a real time hand location technique that consists of three basic components: skin region segmentation, region information computation and hand detection using HOG feature, in which three complementary color models, including the background model, the skin color model and clothes color model, are learnt and updated to pursue skin segmentation. We train these models in Yuv color space using channel u and v which makes the model robust to illumination changes. The combination of the three models works well in indoor scenes even with lighting variations. After segmentation, we have skin connected regions.

Usually there are more edges on the palm region than on the arm region. Based on this observation, we use HOG (Histogram of orientation gradient) features [3] to train a cascade detector by AdaBoost algorithm [7], and locate the hand in skin regions. Our hand detector is trained for all hand postures whose samples are nearly vertical with small perturbations.

To locate hand position in realtime, we first calculate the main orientation of each connected regions of skin color segmentation and accordingly rotate its region to make it vertical, and then use the hand detector in all those skin regions.

The main orientation of connected region can be computed by Eq. (1):

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \quad (1)$$

where μ_{pq} is:

$$\mu_{pq} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} (i - x_c)^p (j - y_c)^q f(i, j) \quad (2)$$

(i, j) represents the coordinate of each pixel in the binary image; $f(i, j)$ denotes whether the pixel contribute to μ_{pq} or not, 1 for pixel in the same connected region and 0 for pixel in background or in other connected regions; (x_c, y_c) is the centroid of the connected region.

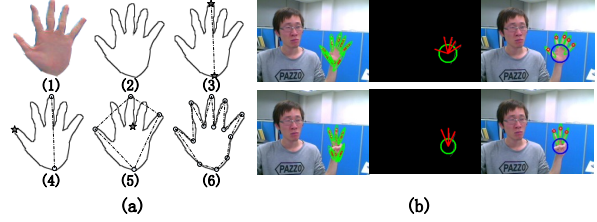


Figure 2: Polygon approximation and feature selection.

2.2 Hand Component Feature Selection

With the multi-scale color features method [2][6] applied to the hand area, blob and ridge features can be extracted. Hand components are then selected from those features in which blob features stand for palm and fingertips while ridges features for fingers. Since those features are originally not organized, we introduce hand skeleton to facilitate this feature selection procedure.

We use a polygon approximation algorithm [1] to extract skeleton and which is summarized as follows. Given the binary hand image (Fig.2a(1)), two contour points with the longest diameter (Fig.2a(3)) are selected to initialize a polygon point set. At each iteration, the most distant point will be added to the point set until all the points on outer contour are less than the distance indicated by the precision parameter. The boundary polygon result can be seen in Fig. 2a(6).

The objective of polygon approximation is to select fingertip points from the polygon boundary points, which is equivalent to exclude the boundary points of the palm. Apparently, the palm is the blob with the largest scale. Some skeleton results are shown in Fig.2b.

According to the skeleton, we select the optimal combination of ridge and blob for each finger. The combination must satisfy these properties: (1) the skin overlap rate is greater than T_s (empirically set to 0.3); (2) it has the minimal differences in angle and distance. To handle with sudden missing of fingers due to poor segmentation, hand component trackers are utilized to track ridges and smooth their variations.

2.3 Classification with Geometric Features

Given a posture, if we know which fingers appear, we can easily tell what the posture is. So we manage to distinguish fingers using geometric features by exploring kinematic constraints of hand with forearm.

2.3.1 Geometric Features Extraction

(1) Pre-processing: Since the orders of fingers in left (clockwise) and right (counter-clockwise) hands are different, we must determine which hand it is before sorting the fingers. So a LDA based classifier with the spatial relationship of face, hand and elbow (distances and angles) is learnt to distinguish left and right hands.

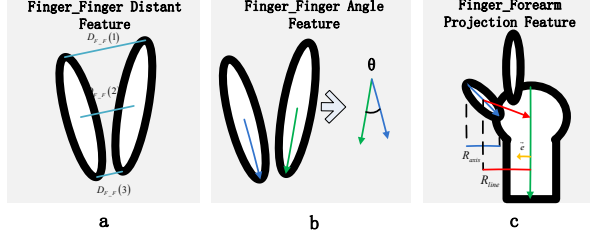


Figure 3: Geometric features.

To put fingers in a serial arrangement, the end-points A and B of the major axis of the reference ridge E_r and the center C of the observation ridge E_o are used:

$$V_l = (x_A - x_B)(y_C - y_B) - (y_A - y_B)(x_C - x_B) \quad (3)$$

$V_l > 0$ means E_o on the left of E_r and vice versa. For left hand, a ridge will be positioned according to the number of ridges on the right of it, and on the contrary for right hand. After that, ridges are in a serial arrangement, which facilitates geometric feature extraction.

(2) Geometric Features: Since the relationship among specific hand components doesn't vary with rotation, the relative features we extract are rotation invariant which will be demonstrated by the experiments in Sec. 3. We adopt three types of geometric features:

Finger-Finger Feature: Geometric features between each pair of fingers are two kinds: the distance (Fig.3a), and the angle (Fig.3b). The normalized distances $D_{F,F}$ are computed to exploit the adjacency relation of the ridges. The angle $V_{F,F}$ between two ridges is useful for determining whether the thumb denoted by the ridge appears. In Eq.(4), P_r is the coordinate of the key points of the reference ridge which is the same as P_o of the observation ridge, $i = 1, 2, 3$ represent for the three key points, and r_{palm} is the radius of palm. θ_r and θ_o , the angles of the ridges, are used to compute $V_{F,F}$.

$$D_{F,F}(i) = \frac{\|P_r(i) - P_o(i)\|}{2 \cdot r_{palm}} \quad (4)$$

$$V_{F,F} = \tan(\theta_r - \theta_o) \quad (5)$$

Finger-Palm Feature: There are also two kinds of geometric features between one finger and the palm. Their formulation are similar to Eq.(4) and Eq.(5), but the usages are quite different. The distance feature in Eq.(6) can tell the short fingers such as thumb and pinky from the others, where P_c is the center of palm; while the angle feature in Eq.(7), where θ_s is the angle of line connecting the center of ridge and the center of palm, can help to distinguish fingers according to their relative position to the center of palm, for an example index and pinky can be distinguished effectively.

$$D_{F,P}(i) = \frac{\|P_r(i) - P_c\|}{2 \cdot r_{palm}} \quad (6)$$

$$V_{F,P} = \tan(\theta_r - \theta_s) \quad (7)$$

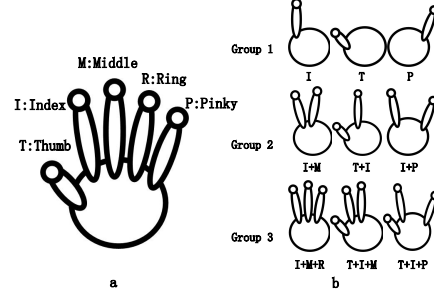


Figure 4: Hand posture groups.

Finger-Forearm Feature: The two kinds of geometric features between finger and forearm are angle feature and projection feature. The angle feature $V_{F,C}$ is used mainly for discriminating thumb and other fingers. The projection feature (Fig.3c) exploits the differences among thumb, index and pinky. The main direction of the connected region (the hand with forearm) is used to approximate the angle of forearm.

$$V_{F,C} = \tan(\theta_r - \theta_c) \quad (8)$$

We denote e as the unit normal vector of the main direction of the connected region. The projections of the major axis of ridge v_{axis} and the line v_{line} connecting the center of ridge and the center of palm (Fig.3c), are computed as follows:

$$R_{axis} = \frac{v_{axis} \cdot e}{2 \cdot r_{axis}} \quad (9)$$

$$R_{line} = \frac{v_{line} \cdot e}{r_{palm} + r_{axis}} \quad (10)$$

(3) Normalization: In order to balance those features, we normalize them as follows:

$$X_i = \left(\frac{x_{i1}}{\sigma_1}, \frac{x_{i2}}{\sigma_2}, \dots, \frac{x_{in}}{\sigma_n} \right)^T \quad (11)$$

$$\sigma_k^2 = \frac{1}{M} \sum_{i=1}^M (x_{ik} - \mu_k)^2 \quad (k = 1, 2, \dots, n) \quad (12)$$

X_i is the i th sample, M is the number of samples. μ_k and σ_k^2 are the mean and variance of the k th feature, n is the number of features.

2.3.2 Hand Posture Classification

Postures are divided into several groups according to the number of fingers. Corresponding SVM classifiers are trained to classify each hand posture into its corresponding groups under the strategy of one versus others. We use the first letters to denote fingers and use their combinations to denote postures (see Fig.4, groups with only one posture are not listed).

3 EXPERIMENTS

Our approach are evaluated on six live videos taken from three persons in four different scenes. The system runs at about 15 fps on QVGA size (320x240) video on an Intel Core Quad 2.66 GHz CPU with 4G RAM.

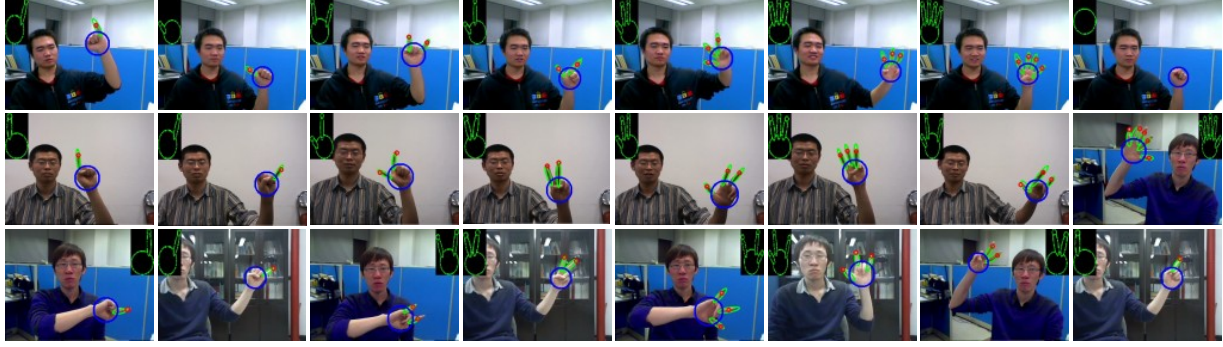


Figure 5: Typical recognition results (upper left symbols for left hand postures, upper right for right hands).

3.1 Experiment Settings

SVM classifiers are trained on a database of 4803 samples for the postures, including 2165 left hand samples and 2638 right hand samples. In practice, to overcome the problem of lacking of Finger-Finger features in one-finger situation, it is desirable to introduce more samples into the training stage.

3.2 SVM Classification for HPR

The evaluation set consists of 1358 test samples, which are completely different with the training samples and contain both hands samples with size and rotation variation. From Table 1, it can be observed that the efficient fusion of three types of geometric features achieve high precision. Without Finger-Finger features in Group 1, the errors occurred when there is a large gap between the angle of forearm and the angle of the connected region. Meanwhile, the viewing angle (not frontal) will shorten distances between fingers, which will sometimes cause the misclassification in Group 2 and Group 3. And the performance will decrease when there are large viewpoint changes.

Overall, the precision is 96.02%, while left hand is 96.53% and right hand is 94.34%. The size of hand posture varies from 39×44 to 119×197 . Besides translation and scale invariant, our features are also robust with rotation. From Table 2, we can see that our method maintains high accuracy with rotation variation.

4 CONCLUSION

In this paper, a robust and efficient hand posture recognition system based on finger geometric feature classification is proposed. Unlike most previous methods, we explore kinematic constraints of the hand with forearm which can tell the corresponding information on possible fingers. The features we extract are translation, rotation and scale invariant. Experiments under different illumination conditions and different scenes demonstrate the effectiveness and efficiency of the proposed method. Since poor skin segmentation caused by

Table 1: PRECISION ON EACH GROUP

Group 1	I	T	P
I	85.52%	14.48%	0%
T	0%	100%	0%
P	0%	3.36%	96.64%
Group 2	I+M	T+I	I+P
I+M	96.89%	0%	3.11%
T+I	2.21%	97.79%	0%
I+P	0.71%	0%	99.29%
Group 3	I+M+R	T+I+M	T+I+P
I+M+R	100%	0%	0%
T+I+M	1.57%	94.49%	3.94%
T+I+P	0.83%	0%	99.17%

Table 2: PRECISION WITH ROTATION

Rotation	$0^\circ \sim 60^\circ$	$60^\circ \sim 120^\circ$	$120^\circ \sim 180^\circ$
Precision	96.08%	97.65%	92.15%

complex backgrounds may lead to wrong recognition result, we will introduce binocular camera into our system, which offers the depth information and improve the skin segmentation.

References

- [1] OpenCV. <http://www.opencv.org.cn/index.php/Download>.
- [2] L. Bretzner, I. Laptev, and T. Lindeberg. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In *FGR*, 2002.
- [3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.
- [4] M. Kolsch and M. Turk. Analysis of rotational robustness of hand detection with viola&jones. In *ICPR*, 2004.
- [5] M. Kolsch and M. Turk. Robust hand detection. In *FGR*, 2004.
- [6] I. Laptev and T. Lindeberg. A distance measure and a feature likelihood map concept for scale-invariant model matching. *Int. J. Comput. Vision*, 52(2/3):97–120, 2003.
- [7] P. Viola and M. Jones. Robust real-time object detection. *Int. J. Comput. Vision*, 57(2):137–154, 2002.
- [8] R. Y. Wang and J. Popovic. Real-time hand-tracking with a color glove. *ACM Trans. Graphics*, 28(3), 2009.