J Intell Robot Syst (2015) 77:583–596
DOI 10.1007/s10846-014-0039-4

Online Dynamic Gesture Recognition for Human Robot Interaction

Dan Xu · Xinyu Wu · Yen-Lun Chen · Yangsheng Xu

Received: 7 May 2013 / Accepted: 7 February 2014 / Published online: 3 April 2014

Abstract This paper presents an online dynamic hand gesture recognition system with an RGB-D camera, which can automatically recognize hand gestures against complicated background. For background subtraction, we use a model-based method to perform human detection and segmentation in the depth map. Since a robust hand tracking approach is crucial for the performance of hand gesture recognition, our system uses both color information and depth information in the process of hand tracking. To extract spatio-temporal hand gesture sequences in the trajectory, a reliable gesture spotting scheme with detection on change of static postures is proposed. Then discrete HMMs with Left-Right Banded (LRB) topology are utilized to model and classify gestures based on multi-feature representation and quantization of the hand gesture sequences. Experimental evaluations on two self-built databases of dynamic hand gestures show the effectiveness of the proposed system. Furthermore, we develop a human-robot interactive system, and the performance of this system is demonstrated through interactive experiments in the dynamic environment.

Keywords Hand gesture recognition · Dynamic gesture spotting · Human-robot interaction

1 Introduction

Human-robot interaction (HRI) is an attractive topic in the research community of computer vision and robotics. As an effective and natural interface for human-robot interaction, vision-based gesture recognition has been studied for years by many researchers [9]. However, due to the difficulty of gesture spotting under complicated backgrounds and illumination conditions in practical interactive applications, dynamic hand gesture recognition remains to be a challenging problem.

A recognition system of dynamic gestures, generally speaking, includes hand detection/tracking, gesture spotting, gesture modeling and classification [17, 21]. Skin-color segmentation [16, 27] and 2D/3D template-matching [1, 4, 29] are widely used to detect the hands in color spaces. However, skin-color

The work described in this paper is partially supported by the National Natural Science Foundation of China (61005012), Shenzhen Fundamental Research Program (JC201105190948A), 2013 Outstanding Youth Innovation Fund in Shenzhen Institute of Advanced Technology, and Guangdong Innovative Research Team Program (201001D0104648280).

D. Xu · X. Wu · Y.-L. Chen (✉)
Guangdong Provincial Key Laboratory of Robotics and Intelligent System, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
e-mail: yl.chen@siat.ac.cn

D. Xu · X. Wu · Y. Xu
Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, Hong Kong

distribution is greatly influenced by the illumination change, and the matching with hand templates is usually seriously interfered by complicated backgrounds. In recent years, depth sensors have been introduced by researchers for improving the performance of the hand gesture recognition. Holte et al. propose a view-invariant hand gesture recognition approach based on 3D motion estimation with a depth camera [10]. In [23], a single stereo camera is used to efficiently track hands with 3D body posture estimation. In [13], an automatic gesture system is developed based on the detection and tracking of hands in the depth map. In [22], color and depth information are combined to detect hands for making the gesture recognition robust to the cluttered background.

Based on the accurate detection of hands, algorithms in [15, 26] have been successfully used to track hands and obtain robust tracking results. Hand gesture spotting, namely extracting the gesture from a hand trajectory, is an important step for the dynamic gesture recognition. Online automatic dynamic gesture recognition depends on an effective gesture spotting scheme. Elmezain et al. [8] design a non-gesture model with a stochastic method for the task of gesture spotting. Stiefmeier et al. [24] treat gestures as strings and use a fast string matching algorithm to spot gestures in real-time. Yang et al. [30] use a sophisticated method to construct a transition gesture model for gesture spotting and recognition. The spotted hand gesture trajectory is actually a spatio-temporal sequence. Mathematical models, such as Hidden Markov Model (HMM) [19], Input-Output Hidden Markov Model (IOHMM) [2], Dynamic Time Warping [6] and Hidden Conditional Random Fields (HCRF) [25], are extensively applied to model and recognize the hand gesture sequences.

In this paper, we propose a dynamic hand gesture recognition system based on an RGB-D camera. The basic framework of the system is shown in Fig. 1. Both depth and color images captured from the RGB-D camera in pairs are used as the input of the system. To resolve the problem of complicated background, we use a model-based human detection method to segment human body in the depth image, then perform background subtraction in the corresponding color image. A procedure of skin-color segmentation in the color image is performed using a model invariant to large illumination changes, and a procedure of hand detection in the depth image is performed with a Chamfer distance matching algorithm. The results of the two procedures are fused to realize a robust hand tracking. For gesture spotting, we use a detection method on change of static postures in the depth image. Finally, discrete HMMs with Left-Right Banded topology are used to model and classify hand gesture with multi-feature representation and quantization of trajectories. The evaluation experiments on two self-built hand gesture databases show the effectiveness of the proposed system. The performance of our system is further demonstrated with a real-world application on human-robot interaction.

The rest of this paper is organized as follows: Section 2 describes the details of the hand gesture detection, which consists of main components including human detection with background subtraction, hand detection, localization, and tracking. Section 3 illustrates the process of dynamic hand gesture recognition including hand gesture spotting, multi-feature extraction/quantization, and dynamic gesture modeling/classification. The system is evaluated in Section 4. Finally, this paper is concluded in Section 5.

2 Hand Gesture Detection

2.1 Human Body Detection and Background Subtraction

Complicated background in the dynamic environment brings a great impact on hand localization and tracking. This situation widely exists in scenarios of HRI applications. Skin-color detection based methods can not deal with the problem of distinguishing objects with skin-like colors in the background, which leads to the complexity of subsequent processings. To solve this problem, we first use a strategy to detect and segment human body in the depth image, then perform background substraction in the corresponding color image. In the system, a model-based approach proposed by Xia et al. [28] is chosen to perform human detection in the depth image. The detection process mainly consists of three steps: (a) using a distance matching algorithm to search possible head regions with a 2D head template; (b) removing fake regions through matching with a 3-D head model produced with the depth information; (c) using a classic region growing algorithm to segment the human body. The

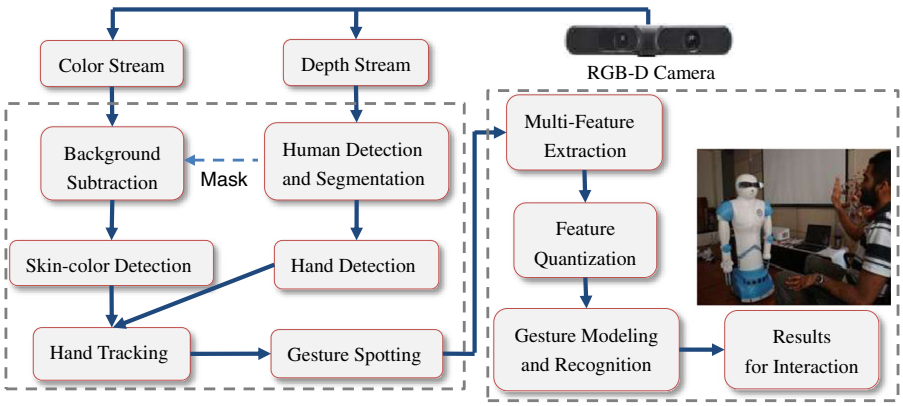

Fig. 1 Flowchart of the proposed human-robot gesture interactive system

segmented region of human body is then used as a mask to perform background substraction. Figure 2 shows experimental examples of human detection in the depth image and background substraction in the color image.

2.2 Hand Detection and Localization

Before tracking hands, we first need to perform hand detection and localization. To avoid the restrictions of hand detection in the color space, the hands are detected only using the depth images. Since the depth images do not have rich local texture compared with color images, they can provide better edge maps for shape matching. In the system we use the chamfer distance matching to detect the hand. Chamfer distance matching is a popular technology for matching two edge images, which can be used for object detection and recognition. It measures the similarity of two binary edge images by chamfer distance. If let $U_T(u_i \in U_T, i = 1, 2, ..., n)$ and $V_Q(v_j \in V_Q, j = 1, 2, ..., m)$ represent the point set of the template edge image and local edge image to be matched, respectively. The chamfer distance between U_T and V_Q can be calculated as follows:

$$d_{cham}(U_T, V_Q) = \frac{1}{n} \sum_{u_i \in U_T} \min_{v_j \in V_Q} \|u_i - v_j\|, \qquad (1)$$

where d_{cham} expresses the mean of distances between each point $u_i \in U_T$ and its nearest edge point in V_Q. In order to reduce the matching cost, the chamfer distance between two edge images can be computed effectively by the distance transform (DT), which converts the binary edge image to be queried into gray images assigning each edge pixel with zero and each non-edge pixel with the distance to its nearest edge point.

In order to find possible regions of the hands in the depth image, Canny edge detector is first used to obtain all edges of the depth image, then a binary hand

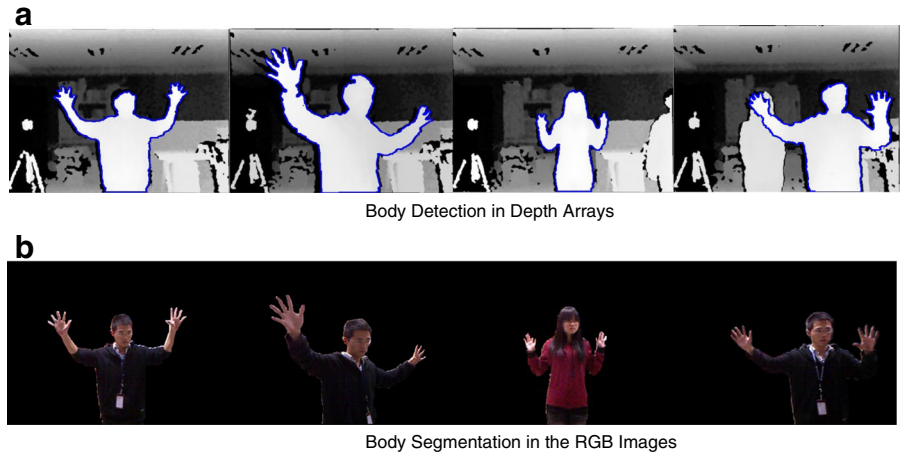

Fig. 2 Human detection and background substraction. **a** Shows examples of human detection in depth images. The extracted body contours are marked with blue lines. **b** Shows examples of background substraction in the corresponding RGB images

Fig. 3 Hand detection and localization using chamfer distance matching in the depth image. **a** Shows a template of hand posture 'Palm'; **b** shows a current input frame after depth smoothing; **c** shows the binary image obtained from a Canny edge detector; **d** shows the distance map from distance transform of the edge image; **e** shows hand detection in which the region with the highest match score is marked

posture template is utilized to match the resulted edge image according to chamfer distance. The distance transform from the resulted edge image to the distance map is made to improve the calculation efficiency during the matching process. Through the sliding window match in the depth image after background subtraction, several hand candidates are kept according to the chamfer distance score d_{cham}. Since the hand is in front of the background when the gesture interaction starts, we can define the final match score M_{score} for every hand candidate by using the distance information as a prior:

$$M_{score} = \frac{s_{max}}{s_{candidate}} * d_{cham}. \tag{2}$$

Here, s_{max} represents the maximal depth distance of the depth sensor and $s_{candidate}$ represents the depth distance of the hand candidate. The value of M_{score} is used to confirm the position of best hand match in the system. Results of the hand detection and localization based on the chamfer distance matching are shown in Fig. 3.

2.3 Dynamic Hand Tracking

When the hand is located in the depth image, the hand region is segmented in the corresponding color image through a statistical elliptical boundary of skin-color model [14]. To make skin-color detection robust to illumination changes, we model skin-color in YCbCr color space considering luminance component Y and chrominance components Cb and Cr in YCbCr color space are separated explicitly. Figure 4a shows the distribution of skin-color in YCbCr color space. The elliptical boundary can be derived from a single Gaussian model (SGM) [31]. SGM is a simple skin detection method which models the distribution as a multivariate Gaussian with the estimated mean μ and covariance matrix Σ. The values of μ and Σ are given as follows:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{3}$$

$$\Sigma = \frac{1}{N}\sum_{i=1}^{n} f_i (x_i - u)(x_i - u)^T, \tag{4}$$

where $N = \sum_{i=1}^{n} f_i$ is the total number of samples in the training data set and $f_i = f(x_i)$ is the number of samples with the chrominance value of x_i, for $i = 1, 2, \ldots, n$. Then the joint probability distribution

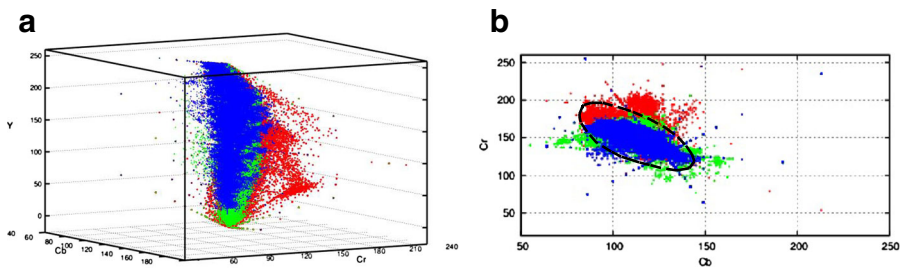

Fig. 4 The statistical results of skin-color distributions in the YCbCr color space

Three-dimensional distribution of skin-color in the YCbCr color space

Two-dimensional distribution of skin-color in the Cb-Cr chrominance space

function (PDF) with a d-dimensional random variable x is defined as:

$$p(x/skin) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\}, \tag{5}$$

where x denotes the color vector of the pixel and $p(x/skin)$ represents the likelihood of the skin-color pixels. Let $D^2 = (x-\mu)^T \Sigma^{-1}(x-\mu)$ to be the Mahalanobis distance from color vector x to the skin-color mean vector μ, which describes the similarity between the pixel and the skin color. If a threshold value D_T^2 is selected, then $D^2 <= D_T^2$ actually defines an ellipse skin-color region as illustrated in 4(b). The elliptical boundary model for skin detection is give as follow:

$$\Phi(x/skin) = [x-\mu]^T \Sigma^{-1}[x-\mu]. \tag{6}$$

Although an elliptical region can also be constructed by a single Gaussian model to describe the skin, the axis of this boundary deviates more from that of skin-color region than the elliptical boundary skin-color detection model. In other words, the elliptical boundary model can fit the skin-color region into an oval better. Then we can segment the hand region in the color image and track the hands using a CAMshift module [3] in a series of video frames with the location and skin-color information of hand blobs.

3 Dynamic Gesture Recognition

3.1 Hand Gesture Spotting

In practical applications, it is still a difficult and critical problem to detect the start and end points of a hand gesture trajectory in a dynamic gesture recognition system. In this paper, we propose an effective scheme for gesture spotting by using two static hand postures (palm and fist) to mark the start and end of a dynamic gesture, as shown in Fig. 5. During hand tracking, the static posture is recognized in every frame using the chamfer distance matching method mentioned in Section 2.2.

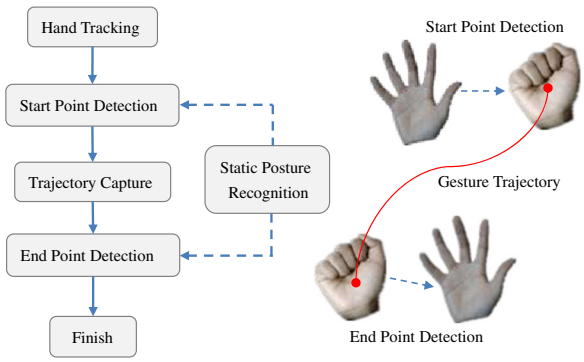

Fig. 5 The proposed scheme for dynamic hand gesture spotting

Considering that the detection and recognition of the static posture with a sliding window on the whole frame has high computational overhead, we delimit a rectangular region of interest (ROI) in the color image estimated by using the point of tracking and the skin-color blob obtained from skin-color detection, then a region in the corresponding depth image is mapped from this region for the static posture recognition. When the static posture changes from palm to fist, the system starts to capture the coordinates of a dynamic gesture trajectory; the capture is finished when the static posture changes from fist to palm. The proposed hand gesture spotting scheme can make the dynamic posture system perform online recognition effectively, and it also can be used to extract continuous dynamic hand gesture in the trajectory.

3.2 Multi-Feature Extraction and Quantization

The recorded coordinate sequence of hand centroid points is converted to a feature vector for training models and recognition. In the proposed system, the orientation, location and velocity features are chosen and combined for improving the recognition performance. For any hand centroid point $p_t(x_t, y_t)$ at time t in the normalized gesture sequence, its orientation in the image plane coordinate system can be approximately described by the direction of the vector $\overrightarrow{p_{t-1}p_t}$, where p_{t-1} denotes the previous point at time $t-1$. $\overrightarrow{p_{t-1}p_t}$ can be represented by the angel α_t, as shown in Fig. 6a. With Eq. (7), we can calculate the value of angle $\alpha_t \in [0, 360°)$ for the point p_t. Then the orientation feature can be represented with an orientation chain code c_t from 1 to 8 by dividing the coordinate system into eight equal portions, as shown in Fig. 6b. To consider 3D gestures, we use the coordinate values (x_t, y_t, z_t) for the location feature. If Δt denotes the duration between two adjacent points, the velocity feature v_t is calculated with $v_t = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}/\Delta t$. Then the final feature vector f_t is $\{c_t, x_t, y_t, z_t, v_t\}$. After performing a

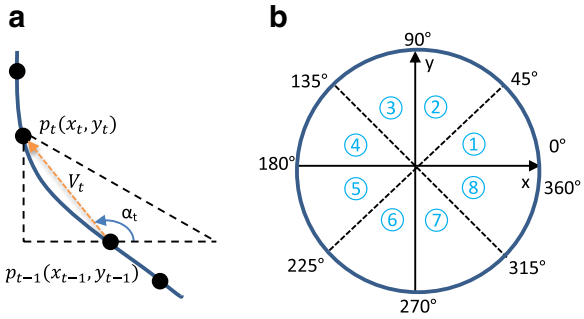

Fig. 6 The illustration of the extraction of the orientation feature from hand gesture sequences

normalization process, all the feature values c_t, x_t, y_t, z_t and v_t are normalized to be in the interval [0, 1].

$$\alpha_t = \arctan\left(\frac{\Delta y}{\Delta x}\right) * \left(\frac{180}{\pi}\right). \tag{7}$$

To obtain the observation symbols in a discrete form, we need a procedure of vector quantization for the feature vectors. Let $F = \{f_1, f_2, \ldots, f_n\}$ denote the feature vector set of all training gesture trajectories, where f_i represents the normalized and weighted feature vector. We use the mean shift algorithm [5] to automatically classify all the feature vectors of the set F into k clusters in the feature space R^5. Each cluster is represented by a mean value m_i and an optional variance d_i which describes the spread. Then a codebook $VQ = \{(m_i, d_i)\}$ can be obtained for the vector quantization, where $i = 1, 2, ..., k$. For any incoming feature vector produced from a point in a gesture sequence, we can assign a cluster index (i.e. observation symbol) corresponding to the closest cluster for it through calculating the Euclidean distances with all the mean values. Then the gesture sequence can be described with a vector of discrete observation symbols.

3.3 Hand Gesture Modeling and Classification

Hidden Markov Models (HMMs) are effective statistical probability models for modeling and classifying sequence data [11, 20]. In our dynamic hand gesture recognition system, the discrete HMMs are applied to model and classify the hand trajectories. We cluster all the feature vectors of one gesture trajectory into K clusters, which is actually equivalent to segment the gesture trajectory into K parts. Then we can obtain K cluster indexes, namely observation symbols O, for the discrete HMMs. The hidden states of the discrete HMMs are the segmented parts of the hand gesture trajectory, which is denoted as $S = \{S_1, S_2, \ldots, S_K\}$, and the observation symbols O is denoted as $O = \{O_1, O_2, \ldots, O_K\}$. The left-right banded structure, in which a state can only go to the next state or go to itself, is adopted for the HMMs. Figure 7 shows the left-right banded state graph with five states. Then the observation probability and the sate transition probability can be described with $P(O_i|S_i)$ and $P(S_i|S_{i-1})$ respectively, where S_i and S_{i-1} represent the current state and the previous state respectively; O_i represents the observation symbol at state S_i.

For the hand gesture recognition, we need to train M Hidden Markov Models corresponding to M hand gestures, and obtain an HMM vector $H = \{\lambda_L =< \pi_L, A_L, B_L >, L \in [1, M]\}$, where π, A and B denote the initial state probability vector, the state transition probability matrix and the observation symbol probability matrix, respectively. The Baum-Welch algorithm can be used to estimate the parameters of the HMMs in the training hand gesture trajectories. During the recognition phase, an incoming hand gesture trajectory is converted to an observation symbol set O, which is the input to the HMM vector H, then the hand gesture label L matched with the vector O can be obtained in the following formula:

$$L = \arg\max\{P(O| < \pi_L, A_L, B_L >)\}, L \in [1, M]. \tag{8}$$

4 Experiments

In this section, experiments are carried out to evaluate the performance of the proposed system. The experimental results include two aspects: dynamic hand gesture recognition with two self-built databases, and an application on human-robot interaction with

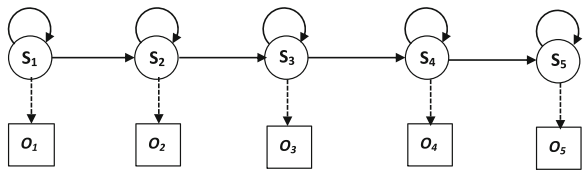

Fig. 7 The Left-Right Banded structure of HMM with five states

a dynamic hand gesture database. The system is deployed in a mobile robot designed in our lab, as shown in Fig. 13. The hardware platform is Intel dual-core i5 2.53 GHz CPU with 2G memory; the software platform is 32-bit Windows 7 and Visual Studio developing environment. We implement the system using C# in conjunction with OpenCVSharp [18] which is a cross platform wrapper of OpenCV for .NET framework written in C#.

4.1 Gesture Data Acquisition

The proposed dynamic hand gesture system uses both depth and color images captured from the Kinect sensor, which is a 3D depth-aware camera launched by Microsoft capable of synchronously acquire real-time 3D depth maps and 2D color images in an indoor environment. The camera can provide 24-bit color images with 640 × 480 resolution and 11-bit grayscale-coded depth images with 320 × 240 resolution in the real-time acquisition of almost 31 frames per second on a standard PC. Within a certain distance range, the Kinect camera system can collect good quality of color and depth images. For the depth images, there are some missing points with depth value 0 due to the high refraction of the structure light. We first perform a preprocessing by using a media filter with a 4 × 4 sliding window to smooth the depth images. Figure 8 shows four pairs of color and depth images after preprocessing, which are obtained from the Kinect sensor under different environmental conditions.

To test the recognition performance of the system, we design two different databases of dynamic hand gestures as shown in Fig. 9. One database consists of six different gestures including "Turn Right" (TR), "Turn Left" (TL), "Move Forward" (MF), "Move Backward" (MB), "Rotate" and "Brake", which are commonly used for human-robot interactions; the other one consists of ten different gesture classes of Arabic numbers from 0 to 9.

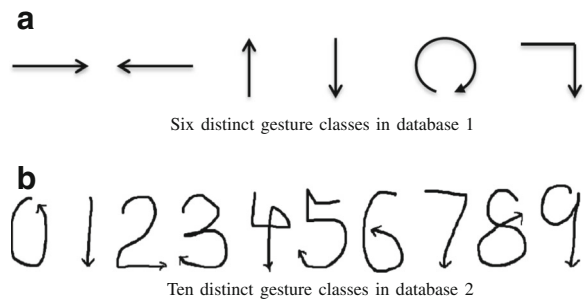

Fig. 9 Description of two different gesture databases for testing recognition performance. **a** "TR", "TL", "MF", "MB", "Rotate" and "Brake" in database 1. **b** Arabic numbers from 0 to 9 in database 2

4.2 Recognition Results on Two Dynamic Gesture Databases

Hand Tracking Hand tracking is a preliminary but vital step of the dynamic hand gesture recognition. The elliptical boundary of skin-color model is used to segment the regions of hand and face after human body detection. Three skin blobs of face and hands can be obtained after background subtraction. Then the hand blob to be tracked is detected and identified in the depth image. The Camshift module implemented in the OpenCV distribution [3] is used to track

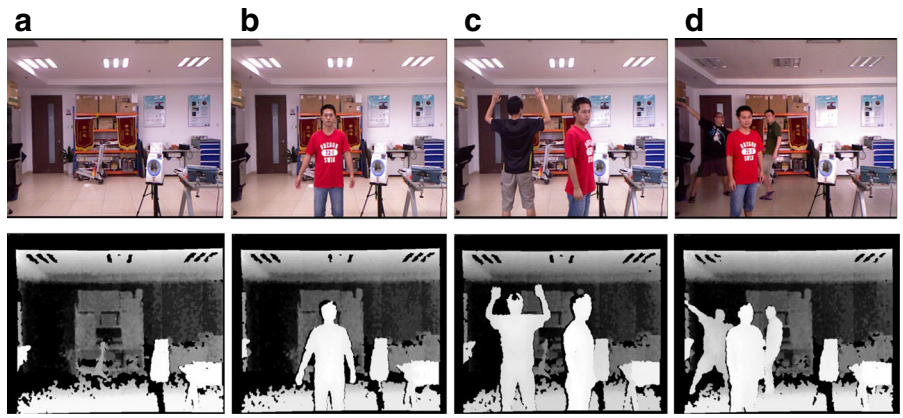

Fig. 8 Color and depth images under different environmental conditions after preprocessing: **a** scene without humans, **b** one person in a lighted room, **c** multiple people in a lighted room, and **d** multiple people in a dark room

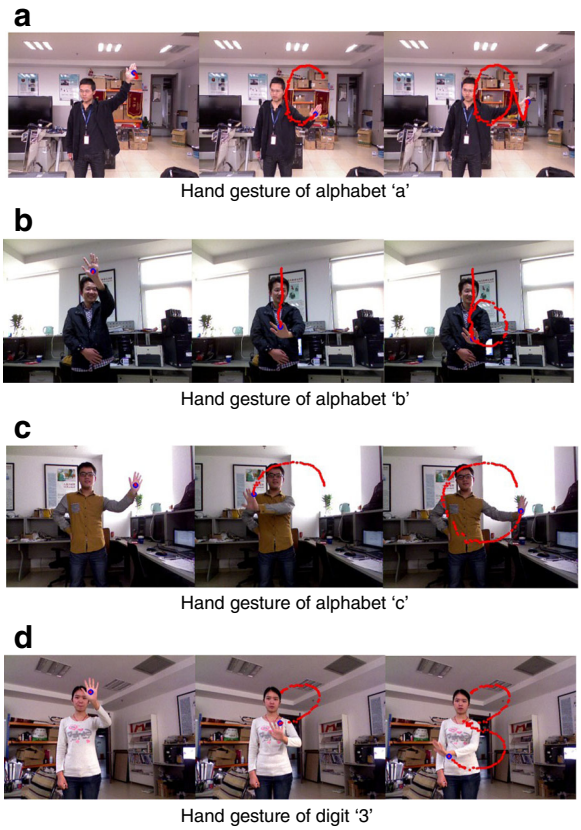

Fig. 10 Hand tracking results on several sequences of frames with different hand gestures under complicated backgrounds. The blue circles indicate the current position of hands, and the red dots record the trajectory

the hand(s) in a series of frames. Figure 10 shows examples of hand tracking results with different hand gestures. Due to the background removal and the skin detection model robust to the illumination changes, the system can obtain reliable hand tracking results for the further hand gesture extraction and recognition.

Gesture Extraction With the proposed gesture-spotting strategy, the gestures performed by people can be extracted automatically. The gesture extraction is a vital step for allowing the system to recognize gestures online, which also directly influences the gesture recognition performance, for example, the gesture extracted incompletely or incorrectly, leading to unexpected recognition results. To evaluate the effectiveness of our strategy, we carry out a gesture extraction experiment under different light conditions and complicated background. The illumination

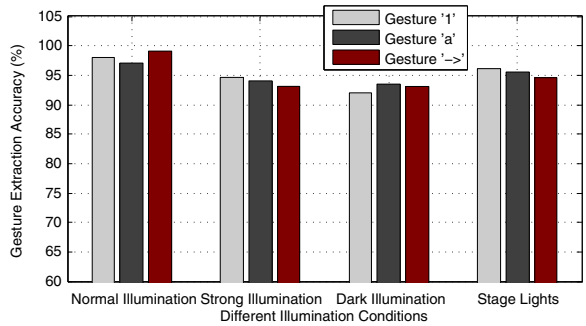

Fig. 11 Results of gesture extraction with three distinct gesture types of '1', 'a' and '→'. The gestures are performed by ten different people under four different illumination conditions including normal illumination, strong illumination, dark illumination and stage lights in the night

conditions include normal illumination, strong illumination, dark illumination and stage lights in the night. To guarantee the user independence, ten different people perform three selected gesture types of '1', 'a' and '→' under each illumination condition, and for each type of gesture, every individual performs 10 times. The accuracy of gesture extraction is calculated as follows:

$$\text{Accuracy}\% = (\#\text{Accurate\_Extraction}/\#\text{Test}) * 100\%. \quad (9)$$

Figure 11 shows the results of gesture extraction experiments. Under the normal illumination condition, the results are best, with an average accuracy of 98 %. Under the other three conditions, the extraction performance decreases with varying degrees. Though the illumination condition does not affect the hand localization based on chamfer matching in the depth image, it gives a strong impact on hand tracking based on skin-color detection. Nevertheless, with the proposed strategy, our tests obtain an average accuracy of 94.3 % under the illumination conditions and the complicated background.

Gesture Recognition To evaluate the recognition performance of the proposed system, we use two gesture databases illustrated in Section 4.1. Ten persons in our lab are selected to perform hand gestures and a pair of 30-second video clips from both the depth stream and color stream is captured as a hand gesture sample. The experimental environment for capturing is under the normal illumination condition. To test the recognition performance only, the video samples with incorrect

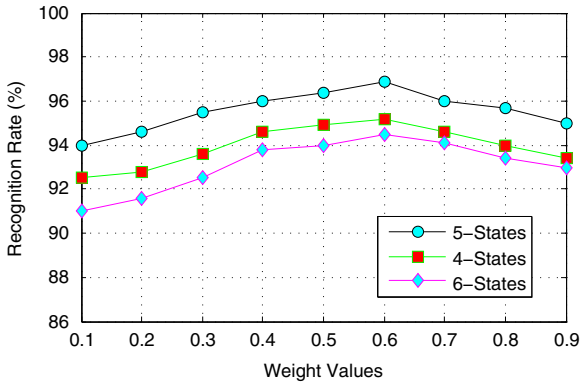

Fig. 12 Recognition results on gesture database 2 with varying weights of velocity and location features in the feature space $c - x - y - z - v$, and the HMMs with 4, 5, 6 states are tested respectively

gesture extraction are manually removed from the database. To guarantee the user independence, the database is divided into two parts: gesture samples from four individuals are used to train the HMMs, and the rest samples are used to test. Each HMM is trained with 60 gesture samples for each gesture class in database 1, and with 80 gesture samples for each gesture class in database 2.

In the system, orientation, location and velocity features are extracted and combined for representing a gesture trajectory. Because the orientation feature is critical for the recognition performance [7], we set the weight of the orientation to 1. To obtain optimal weight values of the location and velocity features, we perform gesture recognition experiments with varying weights of these two features from 0.1 to 0.9. The location and velocity features use the same weight value in our system. Figure 12 shows the recognition results on gesture database 2. As shown in the figure, when the weight value of the velocity and location feature is 0.6, the system can obtain the best recognition performance. With the further increase of the weight value from 0.6, the recognition rate decrease, which shows that the orientation feature is more importance than the location and velocity features. Besides, the HMMs with five states can produce better recognition performance than that with four or six states.

After determining the optimal weight of the location and velocity features, we test the dynamic hand gesture recognition with gesture databases 1 and 2. The HMMs with left-right banded structure and five states are used to model hand gestures. 110 gesture samples for each gesture class in gesture database 1 and 120 gesture samples for each gesture class in gesture database 2 are used to test. Tables 1 and 2 show the statistical experimental results on these two gesture databases respectively. The recognition rate is calculated with the formula:

$$\text{Recognition\_Rate}\% = (\#\text{Accuracy}/\#\text{Test}) * 100\%. \tag{10}$$

The proposed system can obtain an average recognition rate of 98.1 % on gesture database 1 and an average recognition rate of 96.1 % on gesture databases 2, where the results are tested using only the gesture recognition part without combining the extraction part of the proposed system.

Table 3 shows a performance comparison of different methods under three kinds of illumination conditions: normal, dark and strong illuminations. In the comparison of experiments, a standard Dynamic Time warping (DTW) model [12] is use to replace HMM for gesture classification. The gesture extraction module and gesture recognition module are combined for testing the overall performance of the system. The gesture database 2 is used to train and test in the experiments. As shown in Table 3, the system obtains a better recognition performance (an accuracy of 93.6 %) than the DTW method (an accuracy of 88.9 %) under the normal illumination. Besides, the recognition performance decreases to some extent when the illumination changes to dark or strong. It has a recognition accuracy of 90.6 % and 91.6 % under the dark and strong illumination, respectively. This is because the illumination condition directly influences the accuracy of gesture extraction, which leads to incorrect gesture-classification results. Overall, the system yields an average accuracy of 92 % under different illumination conditions and complicated background.

4.3 Gesture Commands for Robot Navigation

As shown in Fig. 13, the service robot in our lab, called "SIATRob", is used as the interaction platform. In the head of "SIATRob", there is an RGB-D camera in the eye position. The stereo camera can capture both color images and depth images. Vision based gesture recognition system can use this camera to track the human and detect the hand gesture. "SIATRob" has two robotic arms with 4-DOF revolute joint, which

Table 1 The statistical results of gesture recognition on database 1

Gesture label	#Train	#Test	#Miss	#Accuracy	Recognition rate(%)
Turn right	60	110	2	108	98.2 %
Turn left	60	110	1	109	99.1 %
Forward	60	110	1	109	99.1 %
Backward	60	110	3	107	97.3 %
Rotate	60	110	2	108	98.2 %
Brake	60	110	4	106	96.4 %
Average recognition rate					98.1 %

Table 2 The statistical results of gesture recognition on database 2

Gesture label	#Train	#Test	#Miss	#Accuracy	Recognition rate(%)
Number 0	80	120	5	115	95.8 %
Number 1	80	120	2	118	98.3 %
Number 2	80	120	4	116	96.7 %
Number 3	80	120	5	115	95.8 %
Number 4	80	120	6	114	95.0 %
Number 5	80	120	6	114	95.0 %
Number 6	80	120	7	113	94.2 %
Number 7	80	120	3	117	97.5 %
Number 8	80	120	4	116	96.7 %
Number 9	80	120	5	115	95.8 %
Average recognition rate					96.1 %

Table 3 The performance comparison of different methods with gesture database 2

Condition	Method	Accuracy (%)
Normal Illumination	Gesture Recognition with HMMs	96.1 %
Normal Illumination	Gesture Extraction + Recognition with HMMs	93.6 %
Normal Illumination	Gesture Extraction + Recognition with DTW	88.9 %
Normal Illumination	Gesture Recognition with DTW	92.2 %
Dark Illumination	Gesture Extraction + Recognition with HMMs	90.6 %
Strong Illumination	Gesture Extraction + Recognition with HMMs	91.6 %

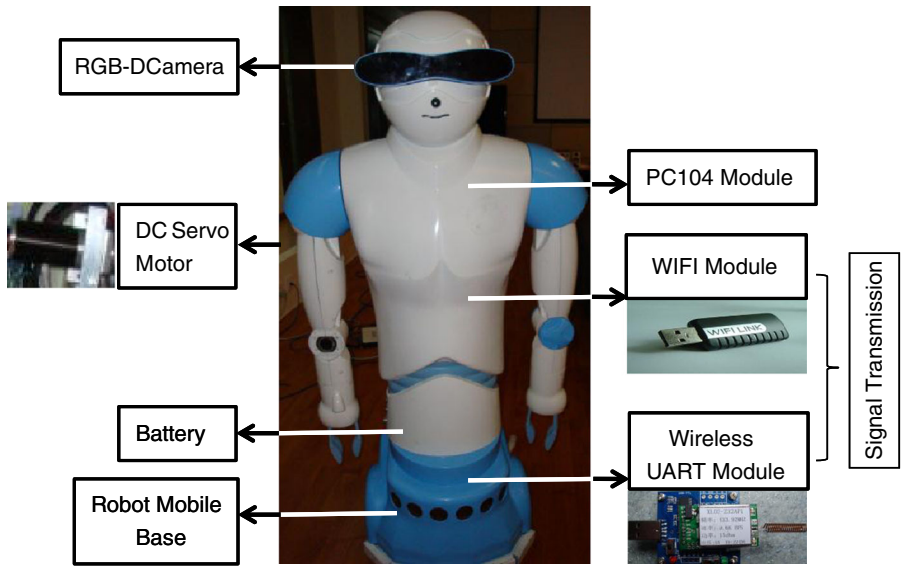

Fig. 13 The basic structure of our service robot "SIATRob", which is used as the platform of human-robot interaction

can be used to grab objects through hand gesture commands. The core module for "SIATRob" is the PC 104 module, which can perform high level tasks following the commands, such as the gesture navigation. The mobile base of "SIATRob" is the Pioneer 3 DX. The platform can rotate with zero radius, climb 25-degree and slope cross a ditch of 2.5 cm. The mobile base has a motor with 500-tick decoders and can perform different kinds of robot motions, such as "Turn Left", "Turn Right", "Move Forward", "Move Back", "Rotate" and "Brake".

The human-robot interaction experiment uses gesture database 1. All defined hand gestures are recognized and converted to commands for the robot navigation. Figure 14 shows the robot navigation in the reality scene using hand gestures. We construct a walking path via putting some furniture in the room, as shown in Fig. 15a. The people perform hand gestures to navigate the robot to walk along the path. Two metrics are used to evaluate the navigation performance. One is the similarity between the ground truth reference trajectory and the walking trajectory of

Fig. 14 The robot navigation with dynamic hand gestures. From the first row to the fourth row: 'Move Forward', 'Move Backward', 'Rotate' and 'Turn Left', respectively

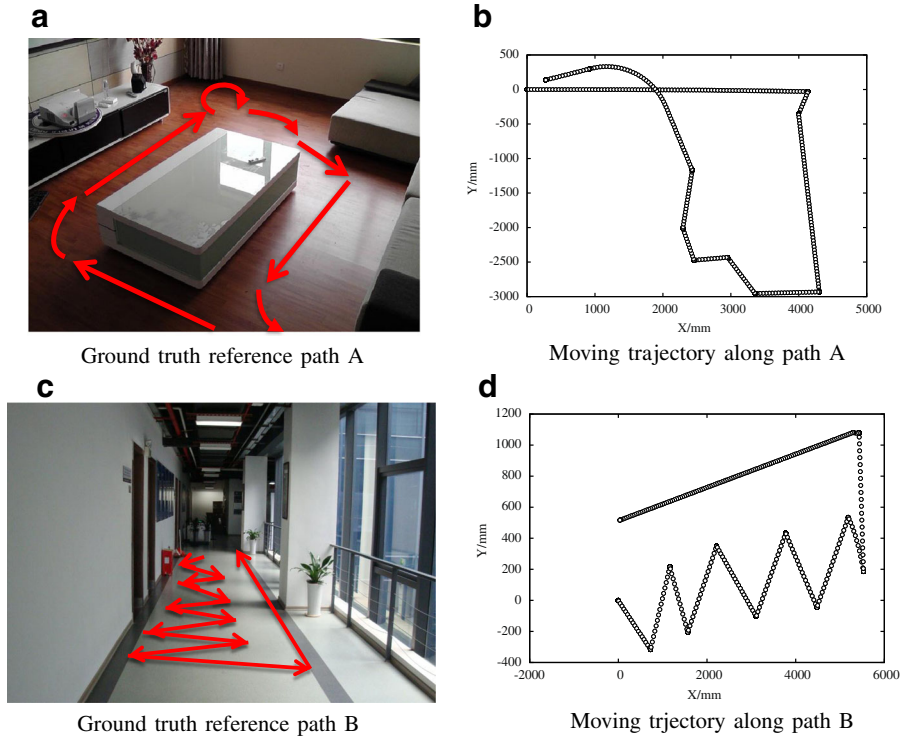

Fig. 15 Experimental results of robot navigation under different scenes

the robot; the other one is the growth of the number of hand-gesture commands the robot received compared with the minimal number of the gesture commands required for a test. To record the real-time coordinates of the robot position in the navigation test, we develop a localization module installed in the robot system. Figure 15b shows the moving trajectory of the robot during a navigation test. Figure 15c and d show a similar navigation experiment in another scene. Obviously, the trajectories produced by the robot are very similar to the ground truth reference trajectories. Table 4 shows the results of four navigation tests with ground truth reference path A. The average growth rate for all the tests compared with the minimal case is 27.8 %, which is a low growth rate for the navigation experiment. The experimental results for the two metrics demonstrate the effective performance of our navigation system. From the aspect of computational speed, the processing of a pair of color and depth frames requires 122 ms on average, namely about 8 frames per second running on our robot platform. This is a nearly real-time processing for hand-robot interaction applications.

5 Conclusion

In this paper, we proposed a dynamic hand gesture recognition system with an RGB-D camera. A chamfer distance matching-based method was used to detect hands in the depth images and a large-illumination invariant skin-color model was used for

Table 4 Statistical results of gesture commands on ground truth reference path A in four navigation tests

Gesture command	#TR	#TL	#MF	#MB	#Rotate	#Brake
Minimal case	3	1	1	0	4	0
test 1	3	2	1	0	5	1
test 2	3	1	1	0	4	1
test 3	4	1	2	1	4	0
test 4	3	1	1	1	4	2

hand segmentation in the color images. The integration of depth and color information made the system perform robust hand tracking against complicated background. To realize automatic and online hand gesture interaction, a reliable dynamic gesture spotting module was developed in the system. Based on multi-feature extraction and quantization of the gesture trajectory, HMMs with Left-Right Banded state graph topology were used to model and classify gestures. Two self-built hand gesture databases were designed to test the performance of the proposed system. The proposed system was evaluated through combining gesture extraction module and gesture recognition module, and obtained an average recognition accuracy of 92 % under different illumination conditions and complicated background. A robot-navigation experiment based on the system further demonstrated our system can work as an appropriate interface for real-life HRI applications.

Acknowledgments The authors would like to thank Hao Li, Lei Zhang, Ming Cheng, Chuan Lin, Xin Kong and Jiping Wang for their constructive advice in paper writing and substantial help during experimental data collection.

References

1. Barczak, A., Dadgostar, F.: Real-time hand tracking using a set of cooperative classifiers based on haar-like features. Res. Lett. Inform. Math. Sci. **7**, 29–42 (2005)
2. Bengio, Y., Frasconi, P.: Input-output hmms for sequence processing. IEEE Trans. Neural Netw. **7**(5), 1231–1249 (1996)
3. Bradski, G.: Computer vision face tracking for use in a perceptual user interface. Intel Technol. J. (1998)
4. Chen, Q., Georganas, N., Petriu, E.: Hand gesture recognition using haar-like features and a stochastic context-free grammar. IEEE Trans. Instrument. Meas. **57**(8), 1562–1571 (2008)
5. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Mach. Intell. **17**(8), 790–799 (1995)
6. Corradini, A.: Dynamic time warping for off-line recognition of a small gesture vocabulary. In: ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, pp. 82–89. IEEE (2001)
7. Elmezain, M., Al-Hamadi, A., Appenrodt, J., Michaelis, B.: A hidden markov model-based continuous gesture recognition system for hand motion trajectory. In: Proceedings of International Conference on Pattern Recognition (ICPR), pp. 1–4 (2008)
8. Elmezain, M., Al-Hamadi, A., Sadek, S., Michaelis, B.: Robust methods for hand gesture spotting and recognition using hidden markov models and conditional random fields. In: IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 131–136. IEEE (2010)
9. Garg, P., Aggarwal, N., Sofat, S.: Vision based hand gesture recognition. World Acad. Sci. Eng. Technol. **49**(1), 972–977 (2009)
10. Holte, M.B., Moeslund, T.B., Fihl, P.: View-invariant gesture recognition using 3d optical flow and harmonic motion context. Comput. Vis. Image Understanding **114**(12), 1353–1361 (2010)
11. Juang, B.H., Rabiner, L.R.: Hidden markov models for speech recognition. Technometrics **33**(3), 251–272 (1991)
12. Keogh, E.J., Pazzani, M.J.: Derivative dynamic time warping. In: The 1st SIAM International Conference on Data Mining (SDM-2001), Chicago (2001)
13. Kurakin, A., Zhang, Z., Liu, Z.: A real time system for dynamic hand gesture recognition with a depth sensor. In: Proceedings of the 20th European Signal Processing Conference (EUSIPCO), pp. 1975–1979. IEEE (2012)
14. Lee, J., Yoo, S.: An elliptical boundary model for skin color detection. In: International Conference on Imaging Science, Systems, and Technology. pp. 572–584 (2002)
15. Maggio, E., Cavallaro, A.: Hybrid particle filter and mean shift tracker with adaptive transition model. In: International Conference on Acoustics, Speech, and Signal Processing. pp. 221–224 (2005)
16. Manders, C., Farbiz, F., Chong, J., Tang, K., Chua, G., Loke, M., Yuan, M.: Robust hand tracking using a skin tone and depth joint probability model. In: The 8th IEEE International Conference on Automatic Face & Gesture Recognition, pp. 1–6. IEEE (2008)
17. Mitra, S., Acharya, T.: Gesture recognition: a survey. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **37**(3), 311–324 (2007)
18. OpenCvSharp: http://code.google.com/p/opencvsharp/ (2011)
19. Rabiner, L., Juang, B.: An introduction to hidden markov models. ASSP Mag. **3**(1), 4–16 (1986)
20. Rabiner, L., Juang, B.: An introduction to hidden markov models. IEEE ASSP Mag. **3**(1), 4–16 (1986)
21. Ramamoorthy, A., Vaswani, N., Chaudhury, S., Banerjee, S.: Recognition of dynamic hand gestures. Pattern Recogn. **36**(9), 2069–2081 (2003)
22. Ren, Z., Meng, J., Yuan, J., Zhang, Z.: Robust hand gesture recognition with kinect sensor. In: Proceedings of the 19th ACM international conference on Multimedia, pp. 759–760. ACM (2011)
23. Song, Y., Demirdjian, D., Davis, R.: Continuous body and hand gesture recognition for natural human-computer interaction. ACM Trans. Interact. Intell. Syst. (TiiS) **2**(1), 5 (2012)
24. Stiefmeier, T., Roggen, D., Tröster, G.: Gestures are strings: efficient online gesture spotting and classification using string matching. In: Proceedings of the ICST 2nd international conference on Body area networks, p. 16. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (2007)
25. Wang, S., Quattoni, A., Morency, L., Demirdjian, D., Darrell, T.: Hidden conditional random fields for gesture recognition. In: IEEE Computer Society Conference on

Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 1521–1527. IEEE (2006)
26. Wang, Z., Yang, X., Xu, Y., Yu, S.: Camshift guided particle filter for visual tracking. Pattern Recogn. Lett. **30**(4), 407–413 (2009)
27. Wu, Y., Liu, Q., Huang, T.S.: An adaptive self-organizing color segmentation algorithm with application to robust real-time human hand localization. In: Asian Conference on Computer Vision (ACCV), pp. 1106–1111 (2000)
28. Xia, L., Chen, C., Aggarwal, J.: Human detection using depth information by kinect. In: Workshop on Human Activity Understanding from 3D Data in Conjunction with CVPR 2011 (HAU3D) (2011)
29. Xu, J., Wu, Y., Katsaggelos, A.: Part-based initialization for hand tracking. In: The 17th IEEE International Conference on Image Processing (ICIP), pp. 3257–3260. IEEE (2010)
30. Yang, H.D., Park, A.Y., Lee, S.W.: Gesture spotting and recognition for human–robot interaction. IEEE Trans. Robot. **23**(2), 256–270 (2007)
31. Yang, J., Lu, W., Waibel, A.: Skin-color modeling and adaptation. Computer Vision-ACCV'98, pp. 687–694 (1997)