J Sign Process Syst (2014) 74:47–58
DOI 10.1007/s11265-013-0778-7

Real-time Hand Gesture Recognition from Depth Images Using Convex Shape Decomposition Method

Shuxin Qin · Xiaoyang Zhu · Yiping Yang · Yongshi Jiang

Received: 24 December 2012 / Revised: 20 March 2013 / Accepted: 13 May 2013 / Published online: 29 June 2013

Abstract Hand gesture recognition is one of the most natural and intuitive ways to communicate between people and machines, since it closely mimics how human interact with each other. This paper presents a novel method for real-time markerless hand gesture recognition from depth images. The proposed method encompasses a collection of techniques that enable the detection, segmentation and recognition of hand gestures. A Hand detection and location method is employed using the depth information acquired from a depth sensor. Then, the hand is robustly segmented in cluttered background without any marker around. A convex shape decomposition method based on Radius Morse function is proposed for hand shape decomposition in real-time. Hand palm, fingertips and hand skeleton are recognized based on the hand shape decomposition and hand features. Moreover, we present a method for recognition of two-hand gestures. Representative experimental results demonstrate qualitatively and quantitatively that accurate hand gesture recognition can be achieved for real-time applications.

Keywords Hand gesture recognition · Depth image · Convex shape decomposition · Skeleton extraction

1 Introduction

The problem of efficient and accurate recognition of hand gesture is theoretically interesting and challenging. Real-time hand gesture recognition affords users the ability to interact with computers in more natural and intuitive ways. Thus, it is fully used in virtual reality and computer games [1]. Conventional hand gesture recognition systems detect and segment hands based on the methods including employing color gloves [2], and skin color detection [3–7], both of which have advantages and drawbacks. Other hand gesture recognition systems detect and segment hands using marker-aided methods [2, 8]. However, these methods are inconvenient compared with markerless vision based solutions. The key problem in gesture interaction is how to make hand gesture understood by computers. Extra instruments or sensors, such as data gloves, might be very easy to collect hand state information. However, these equipments are expensive and inconvenient to users. Thus, markerless and vision based hand gesture interaction has many appealing advantages.

Since Lindberg [9] published his work on scale-space framework for geometric features detection, scale-space feature detection has been widely applied in object recognition, image processing and registering etc. Bretzner et al. [3] and Fang et al. [10, 11] have employed scale-space feature detection method to detect blob and ridge structures of a hand. Both of them define palm and fingers as blob and ridges. However, the scale space feature detection is time-consuming for real-time applications. Moreover, it is difficult for this method to perform in cluttered background because shapes that are similar to palm and fingers in background might interfere with the detection results. Although Fang et al. [11] improves the detection method to reduce the computational cost for real-time application, the accuracy of recognition results is decreased.

A significant amount of literature has been devoted to the problem of skin color detection and segmentation [3, 4, 7]. Lee et al. [7] use skin color segmentation for hand AR (Augmented Reality), which requires a high accuracy of hand contours. They employ a skin color based

S. Qin (✉) · X. Zhu · Y. Yang · Y. Jiang
Institute of Automation, Chinese Academy of Sciences,
Beijing, China
e-mail: shuxin.qin@ia.ac.cn

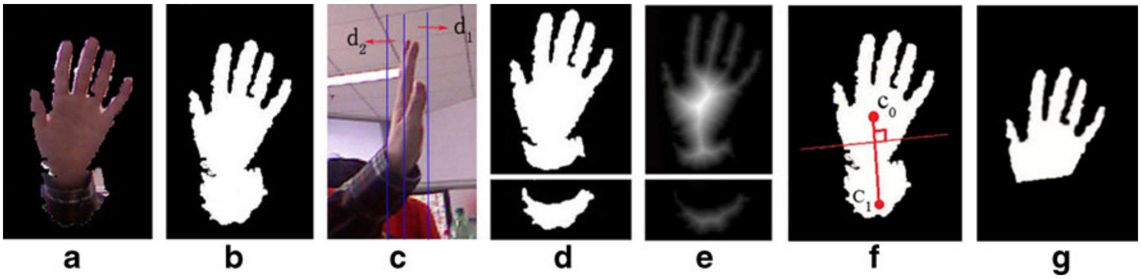


Figure 1 Graphical illustration of the proposed hand detection and segmentation method. **a** The rough hand region; **b** The binary region of (**a**); **c** The segmentation threshold d_1 and d_2; **d** The two parts of the rough hand region; **e** Distance Transform of **d**. In **f**, c_0 and c_1 demonstrate the centers of the two parts. The line perpendicular to $\mathbf{c_0 c_1}$ is the cut line. **g** presents the final hand shape after cutting.

classifier with an adaptively learned skin color histogram. But this method needs a lot of training samples. Argyros et al. [4] select a special color space to reduce the effect caused by background and illumination. Moreover, a technique is proposed that permits the avoidance of much of the burden involved in the process of generating training data in their work.

Depth imaging technology has advanced dramatically over the last few years. Depth cameras offer several advantages over traditional intensity sensors, working in low light levels, being color and texture invariant and resolving silhouette ambiguities in pose [12]. Thanks to the recent development of inexpensive depth cameras, such as Kinect sensor, new opportunities for gesture recognition emerge. Although there are many successful applications for human body tracking [12] and face recognition [13], it is still an challenge to use the low-resolution depth map for hand gesture recognition. A robust hand gesture recognition system using Kinect depth map is developed and used in some applications successfully [8]. However, the user need to wear a black belt, which is inconvenient. Another research on 3D tracking of hand articulations using Kinect [14] presents a good work on modeling a hand, but they segment the hand using skin color method, which can be easily confused by face, bared arm and skin-liked objects. Real-time human pose recognition from single depth images is proposed in [12]. They present a new method to predict 3D positions of body joints from a single depth image with training data, which proves the practical applicability of depth information. Another type of tracking and recognition method is based on time-of-flight (ToF) camera. By employing a ToF camera, a system, which is capable of recognizing gestures at the finger level in real-time, is constructed in [15–17]. A method for human full-body pose estimation from ToF camera images is presented in [18]. Their method can track various full-body movements, including self-occlusions and estimate 3D full-body poses with a high accuracy. However, the method based on ToF camera is hard to provide an accurate result on hand gesture recognition because of its low-resolution. The original depth data from depth sensor, such as Kinect, contains a numerous occlusions and uncovered areas due to the nature of the device and environments. Several studies try to inpaint a low-resolution depth image to achieve an qualified depth map [19, 20].

This paper presents a novel method that segments hand precisely based on depth information without any marker. With the help of depth map filtering, an qualified hand contour is available in real-time. Then, a new robust hand recognition method is proposed. Our hand recognition method is based on approximate convex shape decomposition which is very useful in some graphics and vision tasks. The method is well designed for real-time applications compared with the conventional convex shape decomposition

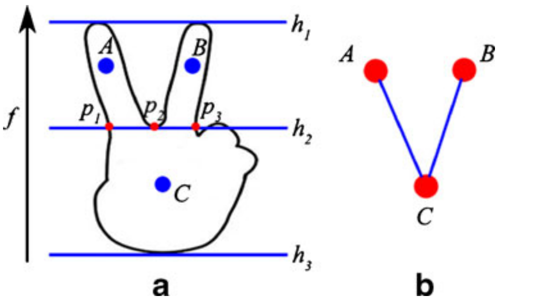


Figure 2 **a** Height based Morse function of a hand gesture; **b** The Reeb Graph.

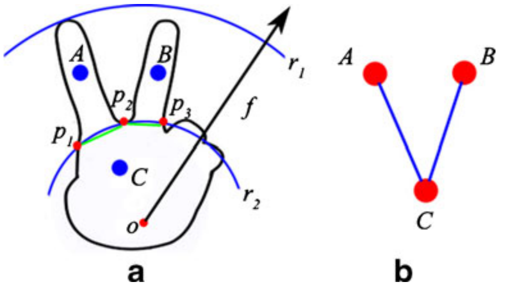


Figure 3 **a** Radius based Morse function of a hand gesture; **b** The Reeb Graph.

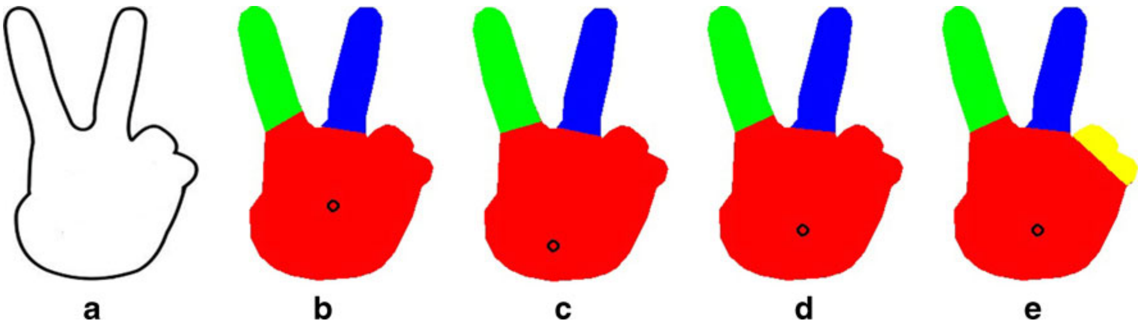


Figure 4 Shape decomposition results with two different parameters (λ employed to specify the central point o and the threshold of shape concavity ε). **a** is the original shape contour. **b**, **c**, **d**, **e** are decomposition results with different parameters where the *black circle* denotes the Morse function center.

which is time-consuming. By employing this shape decomposition method, the hand is decomposed to palm and fingers, which are useful for gesture recognition. Fingertips are detected using a smart method with the finger shapes acquired from the decomposition. We provide a method for hand skeleton extraction, which is successfully used for single hand and two-hand gestures recognition. A simple hand gesture dataset is collected to test the efficiency and accuracy of our method. Initial results of this work have been presented in [21]. The better experiment results reveal that the employed method is very efficient.

The rest of the paper is organized as follows. In Section 2 we present the method for hand detection and segmentation. Section 3 explains in detail the hand shape decomposition and representation approaches. In Section 4, the proposed two-hand gesture recognition method are demonstrated.

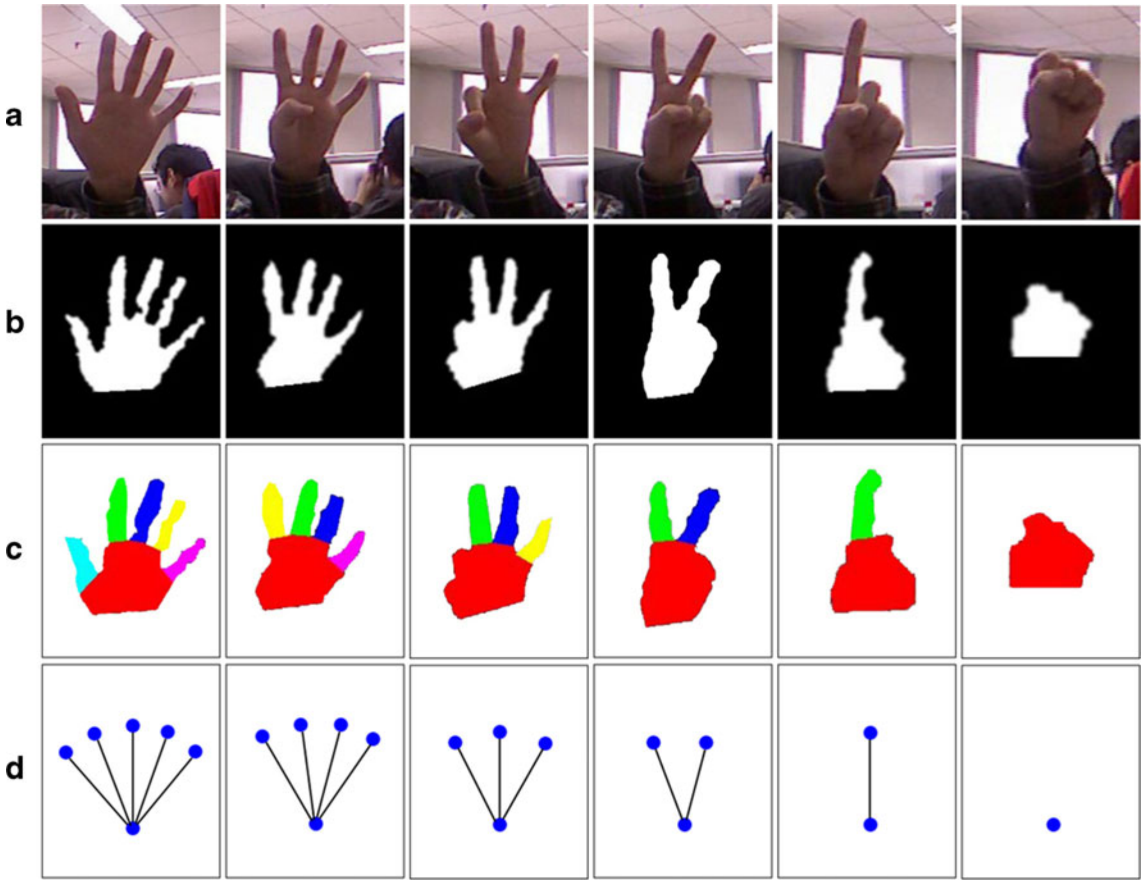


Figure 5 Hand shape decomposition. **a** The color images of some hand gestures. The color images are only used for a better view of the decomposition results. **b** The binary hand maps obtained using the proposed hand detection and segmentation methods. **c** Hand shape decomposition results. **d** The convex graphs of these hand shapes.

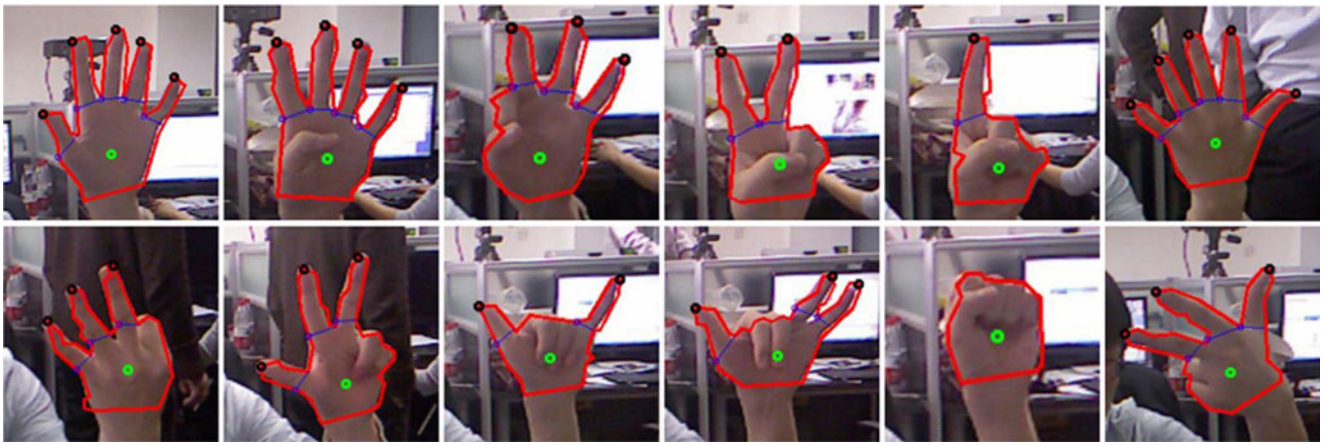


Figure 6 Fingertips detection results. The *black circles* are the detected fingertips.

Experiments and test results of the hand shape decomposition and the hand gesture recognition are shown in Section 5. Section 6 presents the main conclusions of this work.

2 Hand Detection and Segmentation

The proposed hand detection scheme consists of three steps: foreground segmentation, palm localization and hand segmentation. We make several assumptions on hand gesture. First, we assume the hand is the nearest object to the camera. Second, we assume the distance between hand and camera is in the range [0.5, 3.5] in meters. Third, we assume the angles between hand and camera plane are constrained by: $-20° \leqslant \alpha_x \leqslant 20°, -20° \leqslant \alpha_y \leqslant 20°, -180° \leqslant \alpha_z \leqslant 180°$, where $(\alpha_x, \alpha_y, \alpha_z)$ are the three rotation angles between palm plane and camera plane. It starts with thresholding the depth frame to obtain the foreground F. F is given by:

$$F = \{(p, z(p))\,|z(p) < z_0 + z_D\}, \quad (1)$$

where $(p, z(p))$ denotes the pixel in the depth image at cordinate p with value $z(p)$, z_0 is the minimal value of the depth image and z_D is a threshold. We set $z_D = 100mm$ to ensure that the whole hand region is extracted from the depth frame. The detection result of the rough hand region is shown in Fig. 1a and b. In order to detect a more precise hand shape, we define another two thresholds d_1 and d_2 as shown in Fig. 1c, where $d_1 + d_2 = z_D$. Experimentally, we set $d_1 = 70mm, d_2 = 30mm$. Then, the rough hand region is segmented into two parts described in Fig. 1d. Distance Transform operation is employed to calculate the distance map of each part, which is shown in Fig. 1e. The point with maximum distance is selected as the center of each part. We define $R_{in}(x, y)$ and $R_{out}(x, y)$ as the input and output regions of hand; define $l(x, y)$ as the cut line function. The accurate hand region is computed employing the following rule:

$$R_{out}(x, y) = l(x, y) < 0 \cap R_{in}(x, y). \quad (2)$$

In detail, $R_{in}(x, y)$ denotes the hand region before segmentation (coarse hand region); $R_{out}(x, y)$ is the accurate hand

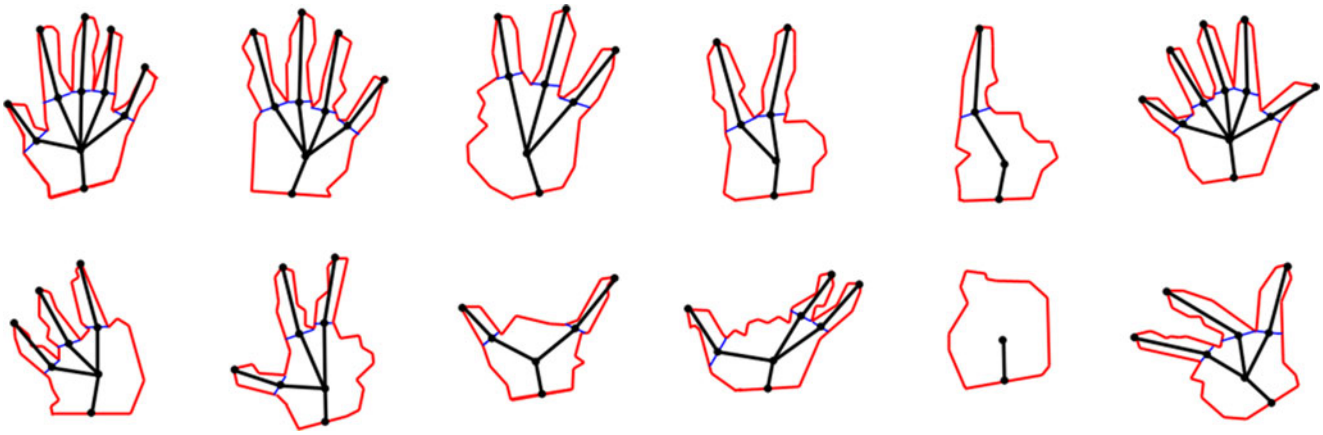


Figure 7 Skeleton Extraction results. The skeleton of each hand shape is denoted as *black line* segments.

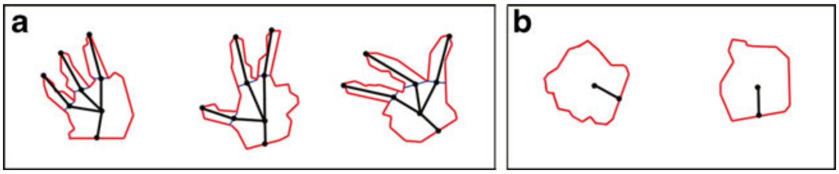


Figure 8 Some hand shapes with same number of components but different skeletons.

region; $l(x, y) < 0$ is the half plane segmented by l. So, the $R_{out}(x, y)$ is the overlap region of $l(x, y) < 0$ and $R_{in}(x, y)$. Thus a more precise hand region is detected, as shown in Fig. 1g. The cut line l is a line perpendicular to the line segment $\mathbf{c_0c_1}$. Then $\mathbf{c_0c_1}$ is cut into two parts. Experimentally, the intersection point of l and $\mathbf{c_0c_1}$ is set to the midpoint of $\mathbf{c_0c_1}$. Thus, given the two points c_0 and c_1, the cut line l is easy to compute. In Fig. 1f, c_0 and c_1 demonstrate the centers of the two parts; the line perpendicular to $\mathbf{c_0c_1}$ is the cut line l.

Due to the nature of the depth sensor, the hand region on the depth map may be have holes and cracks, which will seriously affect the accuracy of hand shape decomposition. Although some inpainting and filtering methods [20, 22, 23] are able to get a better result, the algorithms are usually too complex to be used in real-time applications. We just employ some simple morphological operations to achieve an qualified result.

3 Hand Shape Decomposition and Representation

Shape decomposition and representation is very useful in shape analysis, shape matching, topology extraction, collision detection and other geometric processing methods employing divide-and-conquer strategies [24]. Lien et al. [25] propose methods to decompose polygons into approximately convex parts. Their methods usually result in smaller number of parts. Mi et al. [26] present methods to decompose shapes taking into account relativity to determine part boundaries and achieve a better result. These methods are usually complicated and time-consuming.

3.1 Radius Based Convex Shape Decomposition

We now present our main idea about convex shape decomposition. Our method is inspired by the convex shape decomposition idea [24], which employs the Reeb graph and Morse functions to compute candidate cuts. However, their algorithms compute multiple Morse functions from a number of directions, which is inefficient. As proposed in [24], each decomposed part may not be strictly convex, thus a parameter ε which indicates the convex tolerance of the decomposed parts is defined. Formally, for a shape S, $R(S, \varepsilon)$ is defined as a decomposition that the concavity of every decomposed part is no more than ε. So, $R(S, \varepsilon) = \cup_{i=1}^{n} P_i$, $\forall_{i \neq j} P_i \cap P_j = \emptyset$ and $\forall_{i \leqslant n} Concavity(P_i) \leqslant \varepsilon$, where n is the number of decomposed parts, P_i is a decomposed part and the degree of its concavity is denoted by $Concavity(P_i)$.

The $Concavity(P_i)$ is measured by projecting the shape contour in multiple Morse functions, which is obtained by changing the projecting direction. As shown in Fig. 2a, Morse function $f : M \to S$, is constructed using the Height Function. In Fig. 2b the Reeb graph is determined by the changes in the number of connected components of Morse function f^{-1}. The Reeb graph has three nodes, which reflects partial topological information of the shape in Fig. 2a. However, multiple Morse functions must be computed because the topological information of the shapes is assumed to be unknown to users. This is similar to brute force computing. In order to better use this method on hand shape decomposition, a new Morse function is proposed as shown in Fig. 3a. The new Morse function f is constructed as follows: for every point p in this object, $f(p)$ is the distance between the point p and the central point o, thus called

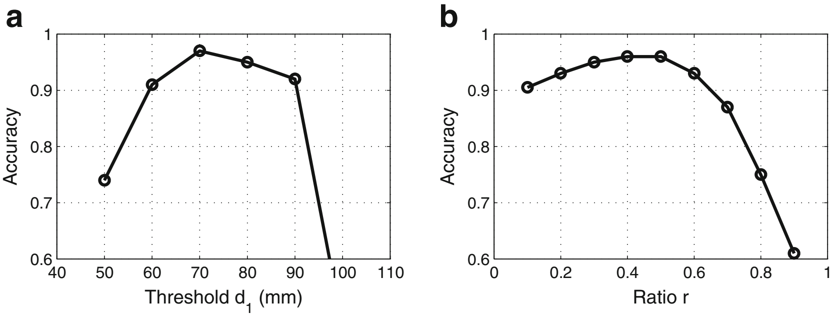


Figure 9 Hand segmentation evaluation. **a** Accuracy with changing d_1. **b** Accuracy with changing ratio r.

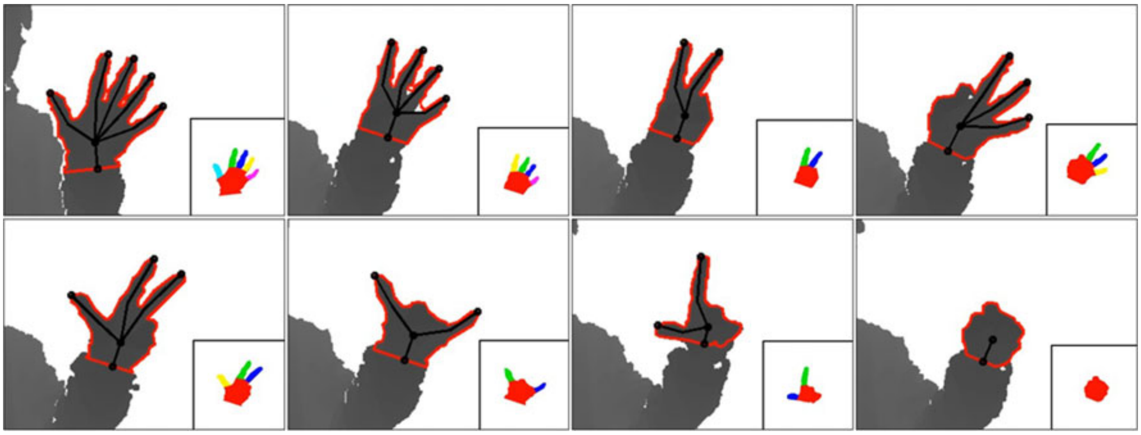


Figure 10 Single hand gesture test samples. For each hand gesture, hand contour and skeleton are described. A hand shape decomposition result is combined with each hand.

Radius Function. Same as the Height based Morse function, the Reeb graph is shown in Fig. 3b. The radius based Morse function is efficient in hand shape decomposition due to the fact that only one Morse function is computed for the decomposition when the central point o is specified. The feasibility of radius based Morse function is based on the topological information of the hand, which is already known to users. As we know, the topological structure of a hand can be defined as a palm and some fingers which are outward around the palm. Moreover, the angle of any two fingers is less than π.

3.2 Candidate Cuts

In order to solve the problem $\forall_{i \leqslant n} Concavity\left(P_i\right) \leqslant \varepsilon$, candidate cuts that can separate a shape S with $Concavity\left(S_i\right) > \varepsilon$, are employed. The way to find a shape S_i with $Concavity\left(S_i\right) > \varepsilon$ is to use the Reeb graph constructed from Radius based Morse function f. The cuts between adjacent nodes of the Reeb graph are all candidate cuts. All the n candidate cuts of a shape S form a candidate cut set, denoted by $C\left(S\right) = \{cut_1, \cdots, cut_n\}$. The final decomposition consists of a subset of $C\left(S\right)$, denoted by $I\left(S\right) \subseteq C\left(S\right)$. A binary variable is assigned to each cut_i in $C\left(S\right)$, as is shown:

$$x_i = \begin{cases} 1 & cut_i \in I\left(S\right) \\ 0 & cut_i \notin I\left(S\right) \end{cases} \tag{3}$$

Thus $\mathbf{x} = \left(x_1, x_2, \cdots, x_n\right)^{\mathrm{T}}$ is a binary vector indicating the selectivity of cuts from $C\left(S\right)$. Each is assigned a value to weight the cost of the cut, denoted by $w\left(cut_i\right)$. Define $\mathbf{w} = \left(w\left(cut_1\right), w\left(cut_2\right), \cdots, w\left(cut_n\right)\right)^{\mathrm{T}}$, thus a decomposition problem is translated in to a integer linear programming. We use the same method proposed in [24] to solve the programming problem:

$$\min \mathbf{w}^{\mathrm{T}}\mathbf{x} \qquad x_i \in \{0, 1\} \tag{4}$$

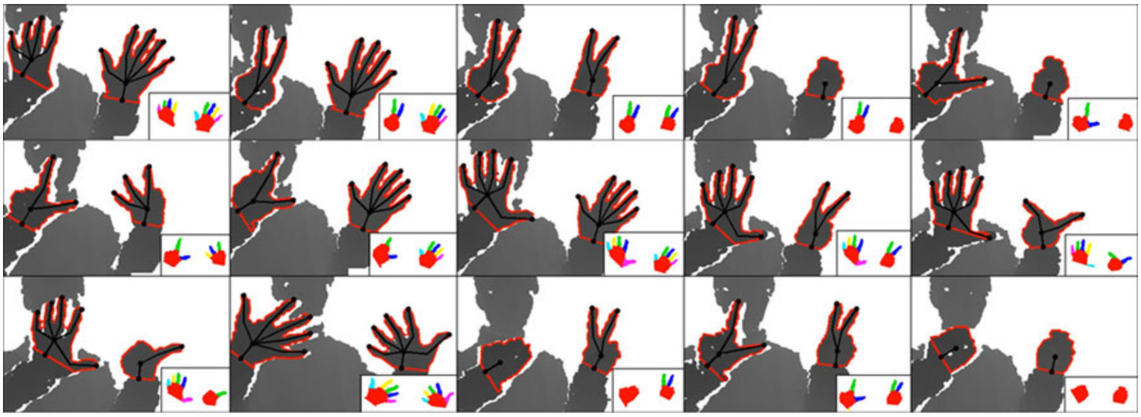


Figure 11 Two-hand gesture test samples. For each two-hand gesture, hand contours and skeletons are described. Hand shape decomposition result is combined with each hand.

For a given cut cut_i, $w(cut_i)$ is defined as function (4):

$$w(cut_i) = \frac{length(cut_i)}{dist(cut_i, o) - r} \tag{5}$$

where $length(cut_i)$ is the length of cut_i, $dist(cut_i, o)$ is the distance between central point o and cut_i, r is the palm radius calculated by Distance Transform of the shape. The central point o is specified when the accurate hand region is segmented. As shown in Fig. 1f, c_0, c_1 are two centers. Thus o is calculated as following:

$$\mathbf{c_0 o} = \lambda \cdot \mathbf{c_0 c_1} \qquad 0 < \lambda < 1 \tag{6}$$

Thus, The procedure of radius based convex shape decomposition method is summarized as Algorithm 1.

Algorithm 1: Radius Based Convex Shape Decomposition

```
Input: A Shape S
Output: Final cut set I (S)
Compute central point o;
Compute Morse Function and Reeb Graph;
Compute candidate cut set C(S);
Define n as the size of C(S);
for i = 0; i < n ; i = i + 1 do
    Compute w (cut_i);
    Check whether cut_i intersects with already
    checked cuts;
Solve the linear programming problem (4) ;
Obtain final cut set I (S).
```

3.3 Hand Shape Decomposition

The proposed shape decomposition method is very useful, at least in two applications. First, it's right for shape representation. After decomposition, since every decomposed part is approximately convex, it can be approximately represented by its convex hull; thus, a compact representation of original object is obtained. Such representation captures not only all the important topological information, but also all the important geometric information of the original object. Second, it's easy to extract the topology of the shape. After decomposition, if we regard each part as a node and two nodes have an edge if and only if they are adjacent, a graph, named convex graph [24], is obtained. Convex graph captures all important topological information of the shape, which is useful in pattern recognition.

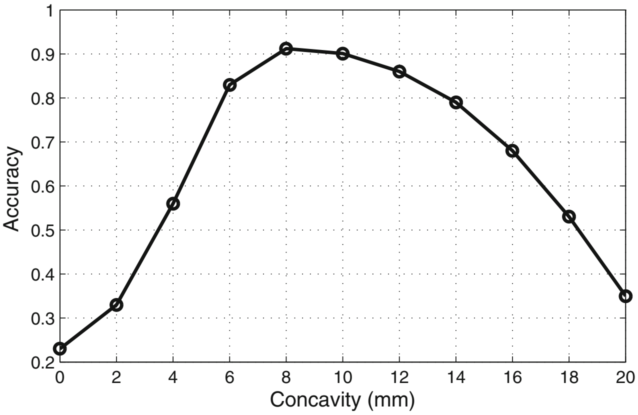


Figure 13 Recognition accuracy with changing concavity.

To use the aforementioned algorithm for hand shape decomposition, two parameters ε and λ are specified. ε is the threshold of shape concavity, λ is the parameter to specify the central point o. Figure 4 shows the results of shape decomposition with different central point and shape concavity. In Fig. 4b, c, d use the same threshold of shape concavity $\varepsilon = 10$, and (e) $\varepsilon = 4$. The final parameters are set based on a large number of experiments. With a set of proper parameters, hand shapes extracted from real environments using depth camera are correctly separated with different colors as shown in Fig. 5c. In Fig. 5, row (a) is the color images used for a better view of the hand gestures; row (b) is binary images obtained by the proposed hand detection and segmentation methods using depth information; row (d) presents the convex graphs of these hand shapes.

3.4 Fingertips Detection

Fingertips are detected from the result of the hand shape decomposition. From the shape decomposition the fingers

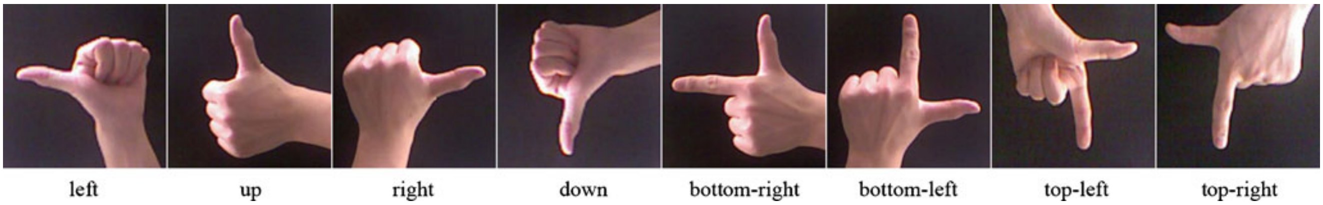


Figure 12 Gesture definition.

Table 1 Recognition results of the gestures for single hand performance.

Gesture	Left	Up	Right	Down	B-right	B-left	T-left	T-right	All
Total	215	233	242	236	262	271	277	282	2018
Correct	203	213	227	218	237	254	245	243	1840
Accuracy	0.944	0.914	0.938	0.924	0.905	0.937	0.884	0.865	0.912

and palm are easy to find. The decomposed shape with hand center is a palm shape and others are finger shapes. For a finger shape S_f, there is a corresponding cut denoted as cut_i. The fingertip point t_{tip} is defined as:

$$t_{tip} = \left\{t_j | \max dist \left(cut_i, t_j\right), t_j \in S_f\right\} \tag{7}$$

which means that the fingertip is the point on the finger shape with the maximum distance against the cut line. Then, we define $T(S)$ as the fingertips set of a shape S. The validity of this method is based on the convexity of the finger shape and the topological structure of a hand. The fingertips detection results are shown in Fig. 6, where the red contours are the recognized hand shape contours, the green circles are the hand palm centers and the black circles denote the detected fingertips in each hand shape.

Each of the fingers has unique functional significance. From the thumb on the radial side to the ulnar side of the hand, the fingers are in this order: Thumb, Index finger, Middle finger, Ring finger, Little finger. With the fingertips detection above, the number of fingers is easy to obtain. If the number of fingers is 5, we only need to find the Thumb or Little finger. But it's hard to recognize the significance of each finger because some of the hand gesture shapes are approximately symmetric. So, the number and positions of fingers should be taken into consideration when defining hand gestures to avoid ambiguity.

3.5 Skeleton Extraction

Skeleton can be viewed as a compact shape representation in that the shape can be completely reconstructed from the skeleton [27]. Some methods have been proposed for skeleton applications, such as human motion tracking [28] and graph matching [29]. Although we have obtained the shape decomposition result and its convex graph, they are not enough to recognize complex gestures. Thus, shape skeleton is a good choice to help represent the hand gestures. The proposed skeleton extraction method is based on the results of shape decomposition and fingertips detection. For a hand gesture shape S, we define c_b as the base point of this shape. c_b is the intersection point of cut line l and line segment joining the two points c_0 and c_1, which is shown in Fig. 1f. Then, we connect the two points c_b and c_0. It is the first skeleton fragment of the shape. With the final cut set $I(S)$, it's easy to obtain the midpoint of each cut line segment cut_i, defined as p_i. And the midpoint set of a shape S is defined as $H(S)$. Thus, the line segment connecting p_i and the corresponding fingertip is a skeleton fragment. Finally, we connect the hand center point c_0 to each of the p_i in $H(S)$. In this way, we simplify a shape skeleton as a set of line segments. We add direction to each line segment of the skeleton. Then skeleton becomes a vector set. Thus, for a shape S, the skeleton $K(S)$ is hierarchically defined as:

$$K(S) = \{\mathbf{c_b c_0}\} \cup \{\mathbf{c_0 p_i} | p_i \in H(S)\} \cup \{\mathbf{p_i t_i} | p_i \in H(S), t_i \in T(S)\} \tag{8}$$

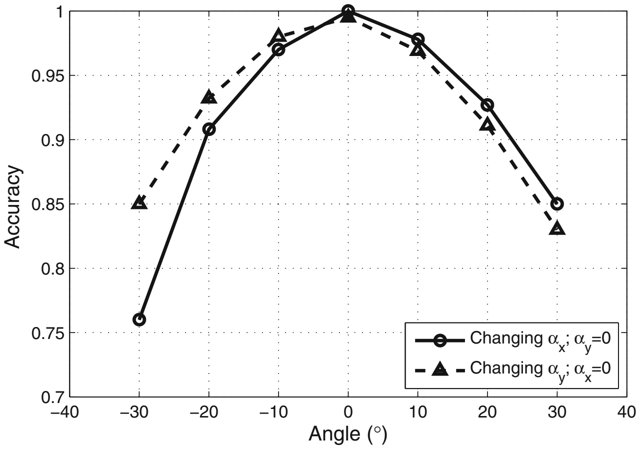


Figure 14 Recognition accuracy with changing α_x and α_y separately.

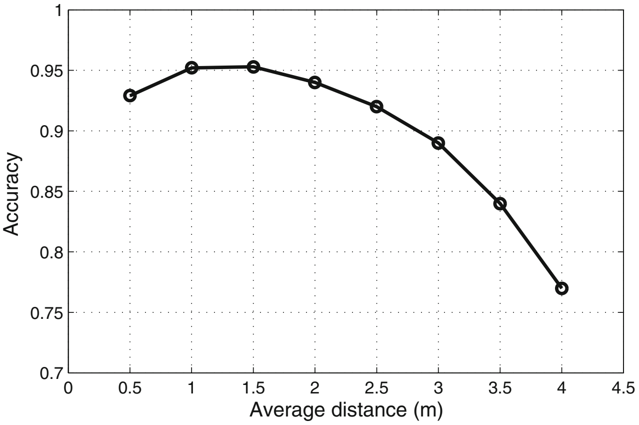
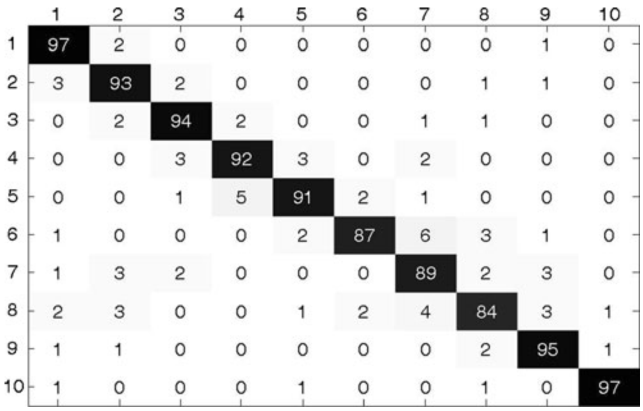


Figure 15 Recognition accuracy with changing distance.



	1	2	3	4	5	6	7	8	9	10
1	97	2	0	0	0	0	0	0	1	0
2	3	93	2	0	0	0	0	1	1	0
3	0	2	94	2	0	0	1	1	0	0
4	0	0	3	92	3	0	2	0	0	0
5	0	0	1	5	91	2	1	0	0	0
6	1	0	0	0	2	87	6	3	1	0
7	1	3	2	0	0	0	89	2	3	0
8	2	3	0	0	1	2	4	84	3	1
9	1	1	0	0	0	0	0	2	95	1
10	1	0	0	0	1	0	0	1	0	97

Figure 16 The confusion matrix of the testing results.

A few skeleton extraction results are shown in Fig. 7, where the black line segments plot the skeleton of each shape. Using this skeleton representation method, gesture recognition is simplified as distance measure between the gesture skeleton and predefined gesture template skeletons. In Fig. 8a, each of the three hand shapes is decomposed into 4 parts, but they are defined as different gestures. So, skeleton distance is employed to distinguish them. To calculate the distance between two skeletons, we encode each vector in the skeleton vector set based on the vector direction to achieve invariance to translation and scale. Then, the distance is easy to obtain. In the case shown in Fig. 8b, we assume they signify the same gesture.

4 Two-Hand Gesture Recognition

Two-hand gesture recognition is an extension of single hand gesture recognition. The number of gestures made by a hand is limited as we know, so two-hand gestures have a lot of room to develop. Moreover, using both hands is a more natural way for people to interact with computers. Two-hand gesture contains not only the gesture message of each hand, but also the relative relationship such as relative positions. First, we assume that the two hands are the nearest object to the camera. Second, the two hands cannot overlap with each other from the camera view. Third, the average depth difference within two hands is less than a threshold $z_M = 30\,mm$. Based on these assumptions, the depth image is divided into two parts, each of which contains a hand region. In detail, the foreground F is given by:

$$F = \{(p, z(p))\,|z(p) < z_0 + z_D + z_M\} \tag{9}$$

which is similar to (1). The foreground F contains two main parts due to the assumptions above. Then, we find the central point of each part using distance transform method. A cut line based on the two central points is used to cut the original depth image into two parts. Finally, we use the proposed hand detection and segmentation method, combined with the hand shape decomposition approach, to deal with each part.

5 Experiments

5.1 Hand Segmentation

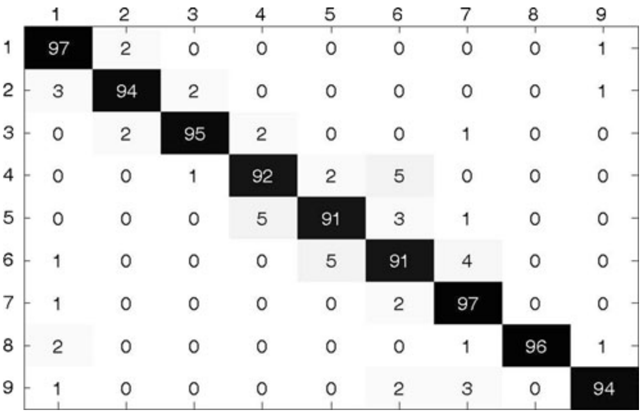
To evaluate the hand segmentation method proposed in this work, the accuracy of hand shape decomposition is employed to demonstrate the quality of segmentation. This is due to that high quality of segmentation brings accurate decomposition result. First, we test the two thresholds d_1 and d_2 (see Section 2). Because we set $d_1 + d_2 = 100mm$, we just need to test one of the two thresholds. Figure 9a presents the accuracy with changing d_1. Then, we test the cut line l defined in Section 2. Define c_x as the intersection point of l and $\mathbf{c_0c_1}$. The ratio $r = \frac{|\mathbf{c_0c_x}|}{|\mathbf{c_0c_1}|}$ is used to determine the line l. The test result is shown in Fig. 9b. So, the configuration of $d_1 = 70mm$ and $r = 0.5$ is the best choice and retained in all further experiments. It should be noted that r is set to 0.5 in Fig. 9a and $d_1 = 70\,mm$ is used in Fig. 9b.

5.2 Shape Decomposition and Skeleton Representation

To assess the validity of our approach, we use real-world depth image sequences obtained by a Kinect sensor to test the proposed hand shape decomposition method and skeleton representation method. Real-world depth image sequences are employed to assess shape decomposition and skeleton representation. Some test results are shown in Fig. 10, where hands are correctly decomposed and skeleton of each hand is denoted as black line segments. Then, two-hand gesture recognition from real-world depth images

Table 2 Comparative testing results of the method in [8] and proposed method.

Method	Accuracy	Running Time
Thresholding Decomposition + FEMD in [8]	90.6 %	0.5004 s
Near-convex Decomposition + FEMD in [8]	93.9 %	4.0012 s
Proposed method	91.9 %	0.026s



	1	2	3	4	5	6	7	8	9
1	97	2	0	0	0	0	0	0	1
2	3	94	2	0	0	0	0	0	1
3	0	2	95	2	0	0	1	0	0
4	0	0	1	92	2	5	0	0	0
5	0	0	0	5	91	3	1	0	0
6	1	0	0	0	5	91	4	0	0
7	1	0	0	0	0	2	97	0	0
8	2	0	0	0	0	0	1	96	1
9	1	0	0	0	0	2	3	0	94

Figure 17 The confusion matrix of the testing results with gestures used in [17].

is performed. A few indicative recognizing results of the proposed method is shown in Fig. 11, which is similar to Fig. 10. The results demonstrate that hand shape decomposition and skeleton extraction are performed as we want.

5.3 Quantitative Evaluation

We have defined a simple hand gesture dataset shown in Fig. 12 to test our method. To test the performance of proposed method, more than 2000 frames were recorded in experiments. We first evaluate the concavity ε. The recognition accuracy with changing concavity is demonstrated in Fig. 13. The configuration of $\varepsilon = 8mm$ is the best choice and used in other experiments. Then we evaluate our recognition method with the best configuration. Table 1 shows the detail results of the hand gesture recognition experiments (We use 'b-right' as a abbreviation of 'bottom-right', 'b-left' as a abbreviation of 'bottom-left', 't-right' as a abbreviation of 'top-right' and 't-left' as a abbreviation of 'top-left'). The average accuracy of recognition in this experiment is about 0.912. The low accuracy of the gestures 'top-left' and 'top-right' is caused by the difficulty of posing these gestures. That is to say, the two angles α_x and α_y between hand and camera plane become large, which significantly affect the performance. Qualitatively, α_z does not affect the performance since the projective shape is constant with changing α_z. So, we just evaluate α_x and α_y using synthetic data. The testing results are shown in Fig. 14, where the optimal angle of α_x and α_y is from $-20°$ to $20°$. Simultaneously changing α_x and α_y is not tested since it is similar to single change qualitatively.

The effect of varying the distance of the hand from the depth sensor is considered in Fig. 15. The experiments are performed with real-world sequences. The accuracy increases as the depth increases until the average depth is about 1.5 m. Then, it declines with an increasing speed. From the plot, the effective distance is from 0.5m to 3.5 m and the optimal distance is from 0.5 m to 2.5 m, in which region the accuracy is higher than 0.9.

5.4 Comparisons

5.4.1 Comparison to Geometry-based Method

We also compare the proposed method with the previous work. This experiments are based on the same dataset provided in [8], which includes 10 gestures denoted as '1','2',$\cdots$,'10' and 100 cases of each gesture. Although each of the cases consists of a color image and a depth map, we just use the depth map which is the only requirement in our method. The confusion matrix of the testing results based on the dataset using the proposed method is shown in Fig. 16. The mean accuracy is about 91.9 %. Compared with [8], our method is much more efficient for real-time applications, which is shown in Table 2.

5.4.2 Comparison to Classification-based Method

ToF camera provides range data which is similar to the depth image from Kinect sensor. In [17], hand features are extracted after segmentation from range data and used for training and classification. We employ real-world depth sequences from Kinect sensor to test the gestures (Gestures are denoted by IDs from 1 to 9, which are EnumOne, EnumTwo, EnumThree, EnumFour, EnumFive, Stop, Fist, OkLeft, OkRight.) used in [17] and the results is shown in Fig. 17. Each gesture is given 100 cases. The overall accuracy is about 0.941 which is very close to the accuracy 0.939 provided in [17]. However, their method

Table 3 Comparative recognition results of the method in [10] and proposed method.

Method	Gesture	Left	Right	Up	Down	Open	Close	All
	Total	215	242	233	236	146	127	1199
Method in [10]	Correct	192	221	217	203	131	119	1083
	Accuracy	0.893	0.913	0.933	0.860	0.897	0.937	0.903
Our method	Correct	203	227	213	218	141	123	1125
	Accuracy	0.944	0.938	0.914	0.924	0.965	0.969	0.938

needs training process, which is more complex. In general, classification-based methods usually employ appearance features for training and classification. So, these methods are not easy to extend. On the contrary, our system is easy to add new gestures by providing the template gesture skeletons.

5.4.3 Comparison to Color-based Method

Because there is no ground truth data to compare the proposed depth-based method with color-based methods, depth and color sequences which are synchronously generated from Kinect sensor are used in further experiments. We implement the method proposed in [10] where scale-space feature detection is integrated into gesture recognition and six gestures in Table 3 are used for quantitative evaluation. In Table 3 we can find that we achieve an accuracy of 0.938 compared with the accuracy of 0.903 obtained by the method in [10].

6 Conclusions

Real-time markerless hand gesture recognition has a wide range of applications, such as virtual interaction, robot control and other kinds of electrical applications. In this paper, we have proposed an efficient method for hand gesture recognition. Hand shapes are detected and segmented from low-resolution depth images which are obtained from a depth sensor. An radius based convex shape decomposition method is introduced at the same time to decompose hand shapes. With the shape decomposition result, fingertips are easily detected. The shape decomposition and fingertips detection, combined with the skeleton extraction, have address the accuracy and efficiency problems of hand gesture recognition to a certain extent. Extensive experimental results demonstrate accuracy, efficiency and robustness of our method.

Acknowledgments This work is supported by CASIA - Beijing CAS X-Vision Digital Technology Co., Ltd. Joint Lab of Information Visualization.

References

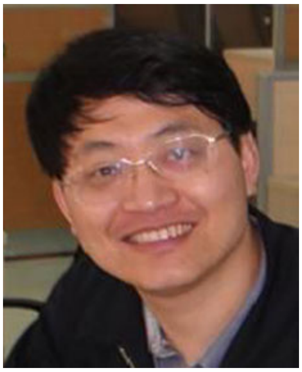
1. Wachs, J.P., Kölsch, M., Stern, H., Edan, Y. (2011). Vision-based hand-gesture applications. *Communications of the ACM*, *54*(2), 60–71.
2. Wang, R.Y., & Popović, J. (2009). Real-time hand-tracking with a color glove. *ACM Transactions on Graphics (TOG)*, *28*(63), 1–8.
3. Bretzner, L., Laptev, I., Lindeberg, T. (2002). Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In *IEEE international conference on automatic face and gesture recognition* (pp. 423–428). IEEE.
4. Argyros, A., & Lourakis, M. (2004). Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *European conference on computer vision* (pp. 368–379). Springer.
5. Argyros, A., & Lourakis, M. (2006). Vision-based interpretation of hand gestures for remote control of a computer mouse. In *Computer vision in human-computer interaction* (pp. 40–51). Springer.
6. Lee, T., Hollerer, T., Handy, A.R. (2007). Markerless inspection of augmented reality objects using fingertip tracking. In *IEEE international symposium on wearable computers* (pp. 83–90). IEEE.
7. Lee, T., & Hollerer, T. (2009). Multithreaded hybrid feature tracking for markerless augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, *15*(3), 355–368.
8. Ren, Z., Yuan, J., Zhang, Z. (2011). Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. In *ACM international conference on multimedia* (pp. 1093–1096). ACM.
9. Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal of Computer Vision*, *30*(2), 79–116.
10. Fang, Y., Cheng, J., Wang, K., Lu, H. (2007). Hand gesture recognition using fast multi-scale analysis. In *International conference on image and graphics* (pp. 694–6980). IEEE.
11. Fang, Y., Wang, K., Cheng, J., Lu, H. (2007). A real-time hand gesture recognition method. In *IEEE international conference on multimedia and expo* (pp 995–998). IEEE.
12. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *IEEE conference on computer vision and pattern recognition* (Vol. 2, pp. 1297–1304).
13. Cai, Q., Gallup, D., Zhang, C., Zhang, Z. (2010). 3d deformable face tracking with a commodity depth camera. In *European conference on computer vision* (pp. 229–242). Springer.
14. Oikonomidis, I., Kyriazis, N., Argyros, A. (2011). Efficient model-based 3d tracking of hand articulations using kinect. In *British machine vision conference* (pp. 101:1–101:11).
15. Van den Bergh, M., & Van Gool, L. (2011). Combining rgb and tof cameras for real-time 3d hand gesture interaction. In *IEEE workshop on applications of computer vision* (pp. 66–72). IEEE.
16. Hackenberg, G., McCall, R., Broll, W. (2011). Lightweight palm and finger tracking for real-time 3d gesture control. *In IEEE virtual reality conference (VR)* (pp. 19–26). IEEE.
17. Molina, J., Escudero-Viñolo, M., Signoriello, A., Pardàs, M., Ferrán, C., Bescós, J., Marqués, F., Martínez, J.M. (2011). Real-time user independent hand gesture recognition from time-of-flight camera video using static and dynamic models. *Machine Vision and Applications*, *24*(1), 187–204.
18. Schwarz, L.A., Mkhitaryan, A., Mateus, D., Navab, N. (2011). Estimating human 3d pose from time-of-flight images based on geodesic distances and optical flow. In *IEEE international conference on automatic face and gesture recognition* (pp. 700–706). IEEE.
19. Zhu, J., Wang, L., Yang, R., Davis, J. (2008). Fusion of time-of-flight depth and stereo for high accuracy depth maps. In *IEEE conference on computer vision and pattern recognition* (pp. 1–8). IEEE.
20. Daribo, I., & Saito, H. (2011). A novel inpainting-based layered depth video for 3dtv. *IEEE Transactions on Broadcasting*, *57*(2), 533–541.
21. Qin, S., Zhu, X., Yu, H., Ge, S., Yang, Y., Jiang, Y. (2012). Real-time markerless hand gesture recognition with depth camera. *In Advances in multimedia information processing – PCM 2012, lecture notes in computer science* (Vol. 7674, pp. 186–197). Springer.

22. Telea, A. (2004). An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, *9*(1), 23–34.
23. Kopf, J., Cohen, M.F., Lischinski, D., Uyttendaele, M. (2007). Joint bilateral upsampling. *ACM Transactions on Graphics*, *26*(96), 1–8.
24. Liu, H., Liu, W., Latecki, L.J. (2010). Convex shape decomposition. In *IEEE conference on computer vision and pattern recognition* (pp. 97–104). IEEE.
25. Lien, J.M., & Amato, N.M. (2006). Approximate convex decomposition of polygons. *Computational Geometry*, *35*(1), 100–123.
26. Mi, X., & DeCarlo, D. (2007). Separating parts from 2d shapes using relatability. In *IEEE international conference on computer vision* (pp. 1–8). IEEE.
27. Bai, X., & Latecki, L. (2007). Discrete skeleton evolution. In *Energy minimization methods in computer vision and pattern recognition* (pp. 362–374). Springer.
28. Schwarz, L.A., Mkhitaryan, A., Mateus, D., Navab, N. (2011). Human skeleton tracking from depth data using geodesic distances and optical flow. *Image and Vision Computing*, *30*(3), 217–226.
29. Bai X., & Latecki, L.J. (2008). Path similarity skeleton graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *30*(7), 1282–1292.

Shuxin Qin received his B.S. degree in computer science and technology from Central South University, China, in 2009. Currently, he is currently pursuing a Ph.D. degree in computer graphics at Institute of Automation, Chinese Academy of Sciences. His research interests include computer graphics and computer vision.

Xiaoyang Zhu received his B.S. degree from Hebei University of Technology, China, in 2008 and the Ph.D. degree from Institute of Automation, Chinese Academy of Sciences in 2013. His research interests include computer graphics and computer animation.

Yiping Yang received his B.S. degree from Xi'an Jiaotong University in 1982 and M.S. degree from Institute of Automation, Chinese Academy of Sciences in 1988. He is currently a Professor and Deputy Director of Institute of Automation, Chinese Academy of Sciences. Prof. Yang's current research interests include intelligent systems and information processing.



Yongshi Jiang received the B.S. degree from Tsinghua University, China, in 1986 and the M.S. degree from Institute of Computing Technology, Chinese Academy of Sciences in 1996. He is currently a Professor in Institute of Automation, Chinese Academy of Sciences. Prof. Jiang's current research interests include information visualization, virtual reality and computer graphics.