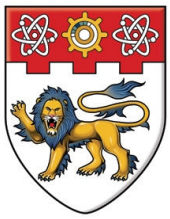
SHAPE BASED HAND GESTURE RECOGNITION



NANYANG TECHNOLOGICAL UNIVERSITY

REN ZHOU

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Master of Engineering

2012

Acknowledgements

I would like to express my most sincere gratitude to my supervisor, Professor Junsong Yuan, for his support, encouragement and instruction in the past two years. Professor Junsong Yuan directed me into such a beautiful and fantastic research realm of computer vision, and supported me with well-equipped environment and active research atmosphere. Besides, Dr. Zhengyou Zhang from Microsoft Research, Redmond, is a wonderful mentor for me. I thank him for his direction, valuable feedbacks and warm encouragement on my research progress. I would also like to thank all the members in my research group. Because of the energetic research style we cultivated, I was passionate and happy. My gratitude goes to my family and my friends, life is beautiful because of you!

Lastly, I want to thank the School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore, for providing me nice research environment and equipment to complete my M.Eng study.

Contents

Summary

According to Siddiqi et al.[45], "Part-based representations allow for recognition that is robust in the presence of occlusion, movement, deletion, or growth of portions of an object. In the task of forming high-level object-centered models from low-level image-based features, parts serve as an intermediate representation". Shape decomposition is a fundamental problem in part-based shape representation. I propose the Minimum Near-Convex Decomposition (MNCD) to decompose arbitrary 2D and 3D shapes into the minimum number of "near-convex" parts. Visual naturalness is important for shape representation [36]. To improve the visual naturalness of the decomposition, two perception rules are considered and the shape decomposition is formulated as a combinatorial optimization problem by minimizing the number of non-intersection cuts. With the degree of near-convexity as a user specified parameter, my decomposition is robust to local distortions and shape deformations.

To justify the advantages of my shape decomposition, I show its superiority in the application of hand gesture recognition. The recently developed depth sensors, e.g., the Kinect sensor, have provided new opportunities for human-computer-interaction (HCI). Although great progress has been made by leveraging the Kinect sensor, e.g. in human body tracking and body gesture recognition, robust hand gesture recognition remains an open problem. Compared to the entire human body, the hand is a smaller object with more complex articulations and more easily affected by segmentation errors. It is thus a very challenging problem to recognize hand gestures. I aim at building a robust hand gesture recognition system from the shape feature, using the Kinect sensor. To handle the noisy hand shape obtained from the Kinect sensor, I propose a novel distance metric, called Finger-Earth Mover's Distance (FEMD), to measure the dissimilarity between hand shapes. As it only matches fingers while not the whole hand shape, it can better distinguish hand gestures of slight differences. In order to accurately detect the fingers, the proposed near-convex shape decomposition method MNCD is employed.

Both theoretical analysis and experimental results show that my shape decomposition outperforms the state-of-the-art methods without introducing redundant parts. My decomposition is robust to shape

distortions and deformations, and it is applicable to 3D shapes. Meanwhile, extensive experiments with the Kinect sensor [1] demonstrate that my hand gesture recognition system is accurate (93.9% mean accuracy on a challenging 10-gesture dataset), efficient (0.0750s per frame), and robust to hand gesture variations (orientation, scale or articulation differences) and can work in uncontrolled environments (with cluttered backgrounds and arbitrary lighting conditions). Finally, my hand gesture recognition system is demonstrated in three HCI demos.

List of Figures

List of Tables

Chapter 1

Introduction

Hand gesture recognition is of great importance to human-computer interaction (HCI), because of its extensive applications in virtual reality, sign language recognition, and computer games [52]. Despite lots of previous work, traditional vision-based hand gesture recognition methods [11] [42] [50] are still far from satisfactory for real-life applications. Because of the nature of optical sensing and the scene complexity, the quality of the images captured by optical sensors is affected by lighting conditions and cluttered backgrounds, thus it is usually unable to detect and track the hands robustly, which largely affects the performance of hand gesture recognition.

First, I give a brief overview of traditional vision-based hand gesture recognition approaches, see [15] [37] [38]for more complete reviews.

From the perspective of extracted features, vision-based hand gesture recognition methods can be classified into three types:

1. The first type is *High-level feature based approaches*: High-level feature based approaches attempt to infer the joint angles of the hand and the pose of the palm from high-level features, such as the joint locations, fingertip and some anchor points on the palm [11]. Colored markers are usually used for feature extraction.

   A common problem with the high-level feature based approaches is in feature extraction. Point features are susceptible to occlusions, thus it is very difficult to track the markers on the image plane because of frequent occlusions or collisions [19]. Non-point features, for instance, protrusions of hand silhouettes [42], were sensitive to hand segmentation performance. Moreover, none of the proposed approaches of this type, including the ones with colored markers, work in cluttered backgrounds.

2. The second type is *3D feature based approaches*: In [8], the 3D depth data is acquired using structured light; however, skin color was a clue used for hand segmentation, which requires homogeneous and high contrast background relative to the hand. Another study [13] proposed to track some interest points on the hand surface using a stereo camera. The 3D trajectories of these interest points was used to augment the range data. Multiple views clues can also be used to create a full 3D reconstruction of the hand surface. However, although 3D data can provide valuable information which help reduce the ambiguities because of self-occlusions, which are inherent in 2D feature-based approaches, a robust, accurate, and efficient 3D hand reconstruction is very difficult. Besides, the additional computational cost hinders its application in real-life systems.

3. The third type is *Low-level feature based approaches*: In many hand gesture recognition applications, all that is needed is a mapping between the hand gesture and input video. Thus, many researchers argued that it is not necessary to reconstruct a full 3D hand model. Instead, many algorithms utilized the low-level features to represent hand gesture that can be extracted efficiently and are fairly robust to distortions. In [49], the principle axes that are defined as an elliptical bounding region of the hand was applied for hand gesture recognition. [54] proposed to use the optical flow and the affine flow of the hand region as the low-level feature. Besides, edges and contours are universal low-level features that are frequently used in model-based hand gesture recognition techniques [34].

However, low-level feature based measures are not effective in cluttered backgrounds. In [50], skin color model was employed to increase robustness, while also restricted the background setting.

From the perspective of problem solving scheme, vision-based hand gesture recognition methods can be classified into two categories:

1. The first category is *Machine Learning based approaches*: For a dynamic gesture, by treating it as the output of a stochastic process, the hand gesture recognition can be addressed based on statistical modeling, such as PCA, HMMs [26] [53], and more advanced particle filtering [24] and condensation algorithms [14].

2. The second category is *Rule based approaches*: Rule based approaches propose a set of pre-encoded rules between input features, which are applicable for both dynamic gestures and static gestures. When testing an hand gesture, a set of features are extracted and compared with the encoded rules, the gesture with the rule that best matches the test input is outputted as the recognized gesture [51].

Traditional hand gesture recognition methods seldom use the hand shape feature because it is difficult to extract a robust hand shape from the optical sensors. As a result, traditional hand gesture recognition methods all apply restrictions on the user or environment, which greatly hinders its widespread use in our daily life. However, the shape feature is shown to be more robust for successful hand gesture recognition than color, texture, shading, or context information [16].

To enable a more robust hand gesture recognition, one effective way is to use other sensors to capture the shape information of hand gestures, e.g. the Kinect sensor. In this thesis, I will present a novel shape based hand gesture recognition algorithm, which uses the part-based representation of the hand shape to recognize the gestures. Now I present the motivation and overview.

1.1 Motivation and Overview

The study of shape is an important theme in computer vision. For example, shape provides one of the major sources of information for recognition. And in image analysis shape plays an essential role, such as in medical image analysis researchers use the shape of an organ to diagnose diseases. It is natural to represent a shape by its parts and there has been strong evidence for part-based representations in human vision [45]. According to Siddiqi et al.[45],

> "Part-based representations allow for recognition that is robust in the presence of occlusion, movement, deletion, or growth of portions of an object. In the task of forming high-level object-centered models from low-level image-based features, parts serve as an intermediate representation."

Given an arbitrary shape, it is thus of great interest to decompose it into a number of natural parts, where each part satisfies certain geometric constraint. The most popular constraint is convexity constraint, because (1) a convex part is visually natural and geometrically simple [7] [47], and thus can serve as a satisfactory primitive for recognition; (2) many operators, which are too complicated to be applied on the original objects, can be easily applied to its convex parts [10] [35]. To this end, strict convex decomposition has been a well studied problem in computational geometry [21] [22].

However, in practice, strict convex decomposition is not robust because it is sensitive to small variations of the shape, such as the local distortions on the contour, which are commonly caused by imperfect image segmentation and shape deformations. In such cases, to satisfy the strict convexity requirement, it usually results in a large number of redundant small parts, thus does not lead to consistent representation.

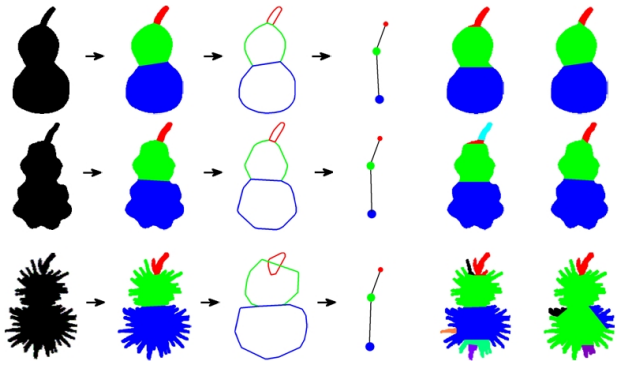


Figure 1.1: A robust shape representation using my method. The first column shows the same objects with different degrees of local distortions. The near-convex decomposition results using my method are shown in the second column. In the third column, the part-based shape representations are illustrated by replacing each part with its convex hull. The fourth column shows the node-graph representations by replacing each part with a node, in which the local information can be imposed. Despite severe local distortions, as my method decomposes a shape into minimum number of near-convex parts, it avoids introducing redundant parts and thus brings consistent decomposition results. The last two columns are the results of existing near-convex decomposition methods: [28] and [33], respectively.

To handle this problem, near-convex decomposition has been proposed. As illustrated in Fig.1.1, instead of requiring each part to be strictly convex, it allows near-convex parts. In [28] [29], Lien et al. proposed a greedy strategy for near-convex decomposition, which exhaustively partitions the most concave feature in the shape until all the parts satisfy the convexity constraint. A recent method proposed by Liu et al.[33] formalized the near-convex decomposition as a linear programming problem by minimizing the total length of cuts, and obtained an approximate optimal solution. Generally, by tolerating local non-convex distortions, near-convex decomposition leads to more robust shape representation.

Despite previous works in near-convex shape decomposition, there still remain two unsolved problems. First, the existing methods cannot avoid introducing redundant parts. For example, the greedy algorithm proposed in [28] [29] inevitably results in redundant parts. Also, by only optimizing the total cut length, the decomposition method of [33] results in redundant parts as well. Thus, these methods cannot generate robust shape decomposition, as illustrated in the last two columns of Fig.1.1. Secondly, without any prior knowledge of the object, it is difficult to obtain visually natural parts through unsupervised decomposition.

To handle these two problems, I present a novel near-convex decomposition method called Minimum Near-Convex Decomposition (MNCD) to decompose arbitrary 2D and 3D shapes. After finding a collection of candidate cuts to partition the shape into near-convex parts, I formulate the shape decomposition problem as a combinatorial optimization problem by selecting the best subset of candidate cuts that has

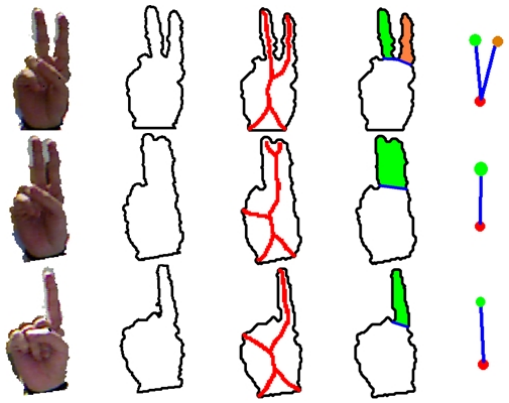


Figure 1.2: The first column illustrates three challenging cases for hand gesture recognition, where the first two hands have the same gesture while the third hand confuses the recognition. Using the skeleton representation shown in red in the third column [4], the last two skeletons are very similar. In the fourth column the decomposition results using my method are illustrated, whose node-graphs are shown in the last column. Although the global structure of the last two node-graphs are more similar, with the help of local information imposed in each node, the dissimilarity distance of the last two gestures is bigger than the first two (using the proposed Finger-Earth Mover's Distance metric, see Section 3.2.3). Namely my part-based representation is helpful for recognizing the cases with local noises.

both a minimum size and high visual naturalness. Two major perception rules, the minima rule [18] and the short cut rule [48], are also imposed to improve the visual naturalness of the decomposition. The optimal solution of this challenging discrete optimization problem can be found efficiently by transforming the problem into a linear programming problem.

My shape decomposition method provides a compact and effective way to represent shapes. As shown in the third column of Fig.1.1, I can approximately represent the original shapes by replacing each part with its convex hull. Further by replacing each part by a node, I obtain a more compact representation of shape, as shown in the fourth column of Fig.1.1. As each node represents a part, the global geometric and topological information is preserved. On the other hand, the information of each part could be imposed into the corresponding node, thus the local geometric information can be preserved as well.

Comparing with the skeleton representation, I can preserve more detailed local information in each node. As illustrated in Fig.1.2, the structure of the skeletons (the third column) of the last two hand shapes are very similar, which confuses the recognition. In the last column, I show the node-graphs obtained by my decomposition method. Similar as the skeleton, the global structure of the last two node-graphs are alike. However, by imposing local information into each node, the dissimilarity distance of the last two graphs are bigger than the first two (I illustrate such distance metric in Section 3.2.3).

Therefore, using the part-based representation I can better handle shapes with local distortions, which is a common problem in many applications. The first two columns of Fig.1.2 illustrate the

distortion problem of hand gesture recognition using Kinect sensor. It can be seen that the contours have significant local distortions in addition to pose variations, where the first two hands have the same gesture while the third hand confuses the recognition. Due to the low resolution and inaccuracy of the Kinect sensor, the two fingers in the second hand of Fig.1.2 are indistinguishable as they are close to each other. Unfortunately, classic shape recognition methods, such as shape contexts [6] and skeleton matching [3] (as shown in the third column), cannot robustly recognize the shape contour with severe distortions. Clearly, recognizing noisy shapes is very challenging, especially if there are many gestures to recognize.

As mentioned before, the part-based representation can well handle the shapes with local distortions by combining the global and local information. In order to address this recognition challenge, I propose a novel distance metric based on the part-representation, called the Finger-Earth Mover's Distance (FEMD). FEMD considers each finger (part) as a cluster and penalizes unmatched fingers.

Extensive experiments on both 2D and 3D shapes show that my decomposition algorithm, MNCD, is robust to local distortions and shape deformation. The comparisons with the state-of-the-art results validate the advantages of my algorithm in terms of reducing the number of redundant parts. With the part-based distance metric, FEMD, I demonstrate the accuracy, efficiency and robustness of my hand gesture recognition system in a 10-gesture dataset, and validate the advantage of part-based representation comparing to skeleton matching and shape context.

1.2 Thesis Contributions

The main contributions of this thesis are as follows:

- ⋆ I propose a novel near-convex decomposition method which decomposes an arbitrary shape into minimum number of near-convex parts, and it can be easily extended to decompose 3D shapes. My decomposition method can well handle local shape distortions and shape deformation, and it is visually more natural.
- ⋆ I present a novel distance metric, Finger Earth Mover Distance (FEMD), for part-based hand gesture recognition. It is efficient, accurate, and robust to articulations, local distortions and orientation or scale changes of the hand shapes. To the best of my knowledge, this is the first attempt in part-based hand gesture recognition using Kinect sensor.
- ⋆ I apply my part-based hand gesture recognition algorithm in HCI applications, and demonstrate its benefit to human life.

1.3 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, I present the formulation, properties, and solution of my decomposition method MNCD. In Chapter 3, I propose the scheme of my part-based hand gesture recognition system on top of the MNCD part-based representation. Then in Chapter 4, experiments of MNCD on 2D and 3D shapes in terms of the parts number, visual naturalness and decomposition robustness are demonstrated. And I also evaluate my hand gesture recognition method on a 10-gesture dataset in terms of robustness, accuracy, and efficiency. In Chapter 5, three HCI applications of my hand gesture recognition system are demonstrated. Finally I draw some conclusion remarks in Chapter 6.

Chapter 2

Minimum Near-Convex Decomposition

2.1 Introduction

Shape decomposition is a fundamental step towards shape analysis and understanding[7]. Such representation method is widely used in shape retrieval [55], skeleton extraction [20] [2], and motion planning [27] [30].

I can classify most shape decomposition methods into two categories. One category is based on geometric constraints. The other category, motivated by psychological studies, aims to decompose shapes into natural components.

In the first category, the most popular geometric constraint is convexity constraint. This is not only because convex components have nice topological and geometric properties that allow for certain operations and improve the efficiency of algorithms, but also because convexity plays an important role in human perception [7]. There are two main indices to evaluate the performance of convex decomposition methods. One index is the time complexity. In this area, Keil et al. proved the time bound to $O(n + r^2\min(r^2, n))$, where n is the number of vertices and r is the number of notches [21]. The other index is the number of decomposed components. In this area, Snoeyink proposed minimum convex decomposition that can decompose 2D shapes into minimum number of strict convex components [22]. However, strict convex decomposition always produces an unmanageable number of components and is very time consuming. Besides, there is no need to find the strict convex components; a certain degree of approximation is enough to satisfy practical processes and is a much more robust representation. Lien and Amato proposed Approximate Convex Decomposition in [28] [29], which decomposes 2D and 3D shapes into approximately convex components. Hairong proposed Convex Shape Decomposition in [33]

with the minimum length of cuts. In these methods, they ignored small concave features and made the decomposition more robust and efficient.

In the second category, the meaning of "natural components" depends on human perception and thus has no objective definition. However, there are some basic perception rules from cognitive science. In [18], Hoffman proposed the *minima rule*, which pointed out that human visual system is interested in boundaries at negative minima of principal curvature or concave creases. Another major perception rule is the *Short cut rule*, proposed by Singh, Seyranian and Hoffman [48], which stated that human preferred the shortest possible cuts for decomposition.

The aim of my method is to decompose an object into minimum number of near-convex parts. And the two major perception rules are incorporated to guide the decomposition, and ensure high visual naturalness. My decomposition is robust to local distortions and shape deformation, which is helpful for many applications, such as hand gesture recognition.

2.2 Problem Formulation for Near Convex Decomposition

2.2.1 Overview

In near-convex decomposition, each decomposed part may not be strictly convex, thus the user has to specify a parameter ψ which indicates the near-convex tolerance of the decomposed parts. Formally, a ψ-near-convex decomposition of a shape S, $\mathrm{D}_\psi(S)$, is defined as a decomposition that only contains ψ-near-convex non-overlapping parts, i.e.:

$$\mathrm{D}_\psi(S) = \{P_i | \bigcup_i P_i = S, \forall_{i \neq j} P_i \cap P_j = \varnothing, concave(P_i) \leq \psi\}, \tag{2.1}$$

where P_i denotes the decomposed part; $concave(P_i)$ is the concavity of P_i. We say P_i is ψ-near-convex if $concave(P_i) \leq \psi$. P_i is strictly convex if $concave(P_i)$=0. According to the definition, near-convex decomposition has two constraints: *the non-overlapping constraint*, $\forall_{i \neq j} P_i \cap P_j = \varnothing$; *the convexity constraint*, $\forall P_i$, $concave(P_i) \leq \psi$.

The partition $\{P_i\}$ is formed by some cuts. For any two vertices p, q on the contour, if the line connecting p and q locates inside the shape, line pq is a cut. As shown in Fig.2.1(b), the red lines are some example cuts. I denote the complete set of all possible cuts in shape S as the candidate cut set, $C(S)$. Therefore, as shown in Fig.2.1, a near-convex decomposition of S is to select a subset of cuts from

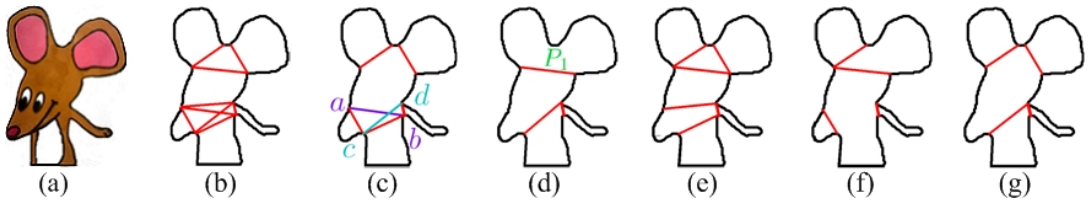


Figure 2.1: Illustration of near-convex decomposition (better viewed in color). (a) The original image. (b) The extracted shape with some sampled candidate cuts inside. (c) An incorrect near-convex decomposition which does not satisfy *the non-overlapping constraint*, as the purple line *ab* intersects with the cyan line *cd* causing the part *abc* to overlap with the part *bcd*. (d) An incorrect near-convex decomposition which does not satisfy *the convexity constraint*, as $concave(P_1) > \psi$. (e) A near-convex decomposition of 7 parts. (f) A minimum near-convex decomposition of 5 parts. (g) Another minimum near-convex decomposition of 5 parts, but looks more natural.

$C(S)$ to form $\{P_i\}$ such that the two constraints in Eq.2.1 are satisfied: 1) as illustrated in Fig.2.1(c), to ensure the non-overlapping constraint, the selected cuts cannot intersect with each other; and 2) as illustrated in Fig.2.1(d), to ensure the convexity constraint, I restrict $\forall P_i$, $concave(P_i) \leq \psi$.

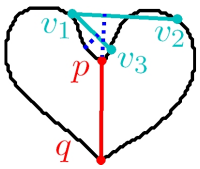


Figure 2.2: At the concave contour, some lines (such as v_1v_2, v_1v_3) intersect with the contour or locate outside the contour, which form the mutex pairs; while vertices v_2, v_3 are not a mutex pair (better viewed in color).

In order to measure $concave(P_i)$, I apply the shape feature *mutex pair* in [33]: for any two vertices on a shape contour, v_1 and v_2, if the connecting line between v_1 and v_2 intersects with the contour or locates outside the contour, (v_1, v_2) is a mutex pair. As shown in Fig.2.2, (v_1, v_2) and (v_1, v_3) are two mutex pairs. The concavity of a part P_i is defined as the maximal concavity of the mutex pairs in the part:

$$concave(P_i) = \max_{(v_1,v_2)\in P_i} \{concave_m(v_1, v_2)\}, \tag{2.2}$$

where (v_1, v_2) denotes the mutex pair in P_i; $concave(P_i)$ denotes the concavity of P_i; $concave_m(v_1, v_2)$ is the concavity of mutex pair (v_1, v_2).

Hence, I can measure $concave(P_i)$ by measuring all $concave_m(v_1, v_2)$ in P_i. I use the same method proposed in [33] to measure $concave_m(v_1, v_2)$: by projecting the shape contour in multiple Morse functions, the concavity of a mutex pair is defined as the maximal perpendicular distance between line v_1v_2

and the corresponding concave contour. As in Fig.2.2, $concave_m(v_1, v_2)$, $concave_m(v_1, v_3)$ are shown as the blue dotted lines, and $concave_m(v_1, v_2) > concave_m(v_1, v_3)$.

To ensure the convexity constraint: $\forall P_i$, $concave(P_i) \leq \psi$, according to Eq.2.2, the concavities of all the mutex pairs in each part P_i must be smaller than ψ. Therefore, for a ψ-near-convex decomposition, I need to separate all the mutex pairs in S whose concavities are greater than ψ into different parts to ensure $concave(P_i) \leq \psi$. As illustrated in Fig.2.2, cut pq separates the heart shape into two parts, and the mutex pair (v_1, v_2) as well as (v_1, v_3) are separated. Thus $concave_m(v_1, v_2)$ and $concave_m(v_1, v_3)$ will not affect the concavities of these two parts.

2.2.2 Minimum Near-Convex Decomposition

As illustrated in Fig.2.1(e), Fig.2.1(f) and Fig.2.1(g), in order to decompose a shape into minimum number of parts with high visual naturalness, I need to optimize the selection of cuts. Assume there are in total n possible cuts in a shape S, namely $C(S) = \{cut_1, ..., cut_n\}$. The final decomposition consists of a subset of the cuts from $C(S)$, denoted by $C'(S) \subseteq C(S)$. I assign a binary variable x_i to each cut_i in $C(S)$ where:

$$x_i = \begin{cases} 1 & cut_i \in C'(S), \\ 0 & cut_i \notin C'(S). \end{cases} \tag{2.3}$$

Thus $\mathbf{x}_{n\times 1} = (x_1, x_2, ..., x_n)^\top$ is a binary vector indicating the selection/rejection of cuts from $C(S)$.

With the two constraints in Eq.2.1, by minimizing the number of cuts and imposing perception rules, I formulate the ψ-MNCD as follows:

$$\begin{aligned} &\textbf{min} \quad \| \mathbf{x} \|_\mathbf{0} + \lambda \mathbf{w}^\top \mathbf{x}, \\ &s.t. \quad \mathbf{Ax} \geq \mathbf{1}, \quad \mathbf{x}^\top \mathbf{Bx} = 0, \quad \mathbf{x} \in \{0, 1\}^n, \end{aligned} \tag{2.4}$$

where $\|\mathbf{x}\|_0$ is the zero-norm of vector $\mathbf{x}$, which counts the number of the selected cuts in $C'(S)$. $\lambda \geq 0$ is a parameter introducing the visual naturalness regularization $\mathbf{w}^\top \mathbf{x}$ to the decomposition, in order to regularize the cuts selection by favoring the cuts with high visual naturalness. I will discuss λ in Section 2.3.1. Now I explain my formulation.

The visual naturalness regularization: $\mathbf{w}^\top\mathbf{x}$

I employ both the minima rule [18] and the short cut rule [48] to ensure high visual naturalness of the decomposition. A cost is assigned to each $cut_i \in C(S)$ to evaluate its own visual naturalness, and a smaller cost means a higher visual naturalness:

$$w_{pq} = \frac{dist(pq)}{1 + \beta \cdot |\min\{cur(p), 0\} + \min\{cur(q), 0\}|} \quad , \tag{2.5}$$

where cut pq is a candidate cut in $C(S)$; $dist(pq)$ is the normalized distance between vertices p and q. This corresponds to the short cut rule: a shorter cut has a smaller cost. $cur(p)$ denotes the normalized curvature of the vertex p, which corresponds to the minima rule: a cut resolving at positions with negative curvatures of larger absolute values has a smaller cost. I normalize the negative curvature among concave vertices and ignore the convex vertices. β is a parameter balancing these two rules. As both rules are critical for natural decomposition, I set $\beta = 1$ in my experiments.

I denote $\mathbf{w}_{n\times 1}=(w_1, w_2, ..., w_n)^\top$ as the costs of n candidate cuts. From Eq.2.5, we know that the cuts separating at positions with negative curvatures of larger absolute values and with shorter lengths have smaller costs. Thus by minimizing $\mathbf{w}^\top\mathbf{x}$, those cuts with higher visual naturalness are more likely to be selected.

The convexity constraint: $\mathbf{Ax}\geq\mathbf{1}$

As mentioned in Section 2.1, to ensure the convexity constraint: $\forall P_i$, $concave(P_i) \leq \psi$, I need to separate all the mutex pairs whose concavities are greater than ψ into different parts. So I first obtain the ψ-mutex set of S, $M^\psi(S)$, which is defined as the set of mutex pairs whose concavities are greater than ψ. Then I separate all the mutex pairs in $M^\psi(S)$ with the selected cuts from $C(S)$. A candidate cut may separate several mutex pairs, such as the cut pq in Fig.2.2. For every candidate cut in $C(S)$, cut_i, the mutex pairs it can separate form a subset of $M^\psi(S)$, denoted by M'_i. In this way, we obtain $\{M'_i,\ i = 1, ..., n\}$.

Suppose there are m mutex pairs in the ψ-mutex set, $M^\psi(S) = \{mp_1, ..., mp_m\}$. For each mutex pair in $M^\psi(S)$, mp_i, among all the cuts that can separate it, at least one cut must be in set $C'(S)$. Thus, for each mp_i, this gives a constraint:

$$\sum_{j=1}^{n} a_{ij}x_j \geq 1, \text{ where } a_{ij} = \begin{cases} 1 & mp_i \in M'_j, \\ 0 & mp_i \notin M'_j. \end{cases} \tag{2.6}$$

Let us denote $\mathbf{A}_{m\times n}=(a_{ij}|i = 1, ..., m;\ j = 1, ..., n)$, $\mathbf{1}_{m\times 1}=(1, ..., 1)^\top$. Consider all the m mutex pairs in $M^\psi(S)$, we have the convexity constraint: $\mathbf{Ax}\geq\mathbf{1}$, which is also used in [33].

The non-overlapping constraint: $\mathbf{x}^\top\mathbf{Bx}=\mathbf{0}$

As for two cuts in $C(S)$, cut_i and cut_j, they may intersect with each other. I define an intersection matrix, $\mathbf{B}_{n\times n}$, to indicate the intersection relations in $C(S)$:

$$b_{ij} = \begin{cases} 0 & cut_i \text{ does not intersect with } cut_j, \text{ and } i \neq j, \\ 1 & cut_i \text{ intersect with } cut_j, \text{ and } i \neq j, \\ 0 & i = j. \end{cases} \tag{2.7}$$

As mentioned in Section 2.1, to ensure the non-overlapping constraint $\forall_{i\neq j}P_i \cap P_j= \varnothing$, the selected cuts in $C'(S)$ cannot intersect with each other, namely $\forall\ x_i, x_j \in \mathbf{x}$, $x_i \times b_{ij} \times x_j = 0$. Thus we have the intersection constraint: $\mathbf{x}^\top\mathbf{Bx}=0$.

2.3 Solution

2.3.1 Selection of parameter λ

As mentioned earlier, λ is an important parameter introducing the visual naturalness regularization to the decomposition. If we do not consider the visual naturalness of the decomposition, while only focus on the minimum number of parts, the problem can be reformulated by setting $\lambda = 0$, i.e.:

$$\mathbf{min}\ \ \| \mathbf{x} \|_\mathbf{0}\ \ s.t.\ \mathbf{Ax} \geq \mathbf{1},\ \mathbf{x}^\top\mathbf{Bx} = 0,\ \mathbf{x} \in \{0, 1\}^n. \tag{2.8}$$

The solution $\mathbf{x}$ of this formulation is not unique, but it ensures exactly minimum number of parts. Although with different objective functions, I can prove that my formulation in Eq.2.4 can obtain the same minimum number of parts as Eq.2.8 if λ is selected appropriately. Theorem 1 tells the relationship between Eq.2.8 and my formulation in Eq.2.4:

Theorem 1 minimum decomposition rule

We consider two objective functions as follows:

$$\begin{cases} f(\mathbf{x}) = \| \mathbf{x} \|_0 + \lambda \mathbf{w}^\top \mathbf{x}, & s.t.\ \mathbf{A}\mathbf{x} \geq \mathbf{1},\ \mathbf{x}^\top \mathbf{B}\mathbf{x} = 0,\ \mathbf{x} \in \{0,\ 1\}^n, \\ g(\mathbf{x}) = \| \mathbf{x} \|_0, & s.t.\ \mathbf{A}\mathbf{x} \geq \mathbf{1},\ \mathbf{x}^\top \mathbf{B}\mathbf{x} = 0,\ \mathbf{x} \in \{0,\ 1\}^n, \end{cases}$$

Let:

$$\mathbf{x}' = \arg\min_{\mathbf{x}} f(\mathbf{x}), \qquad \mathbf{x}'' = \arg\min_{\mathbf{x}} g(\mathbf{x}).$$

We have $\| \mathbf{x}' \|_0 = \| \mathbf{x}'' \|_0$ *when* $0 \leq \lambda \leq 1/\sum_{i=1}^{n} w_i$.

The proof of Theorem 1 is in the Appendix. $\mathbf{x}''$ is the solution of Eq.2.8 whose zero-norm is minimized, and $\mathbf{x}'$ is the solution of my formulation in Eq.2.4. Therefore, my formulation can decompose a shape into minimum number of parts when $0 \leq \lambda \leq 1/\sum_{i=1}^{n} w_i$. It is worth mentioning that although Eq.2.4 and Eq.2.8 both minimize the number of parts, their cuts are not necessarily the same subset from $C(S)$, as Eq.2.4 favors visually more natural cuts.

2.3.2 Efficient Solution by Formulation Transform

The shape decomposition problem formulated in Eq.2.4 is a NP-hard combinatorial optimization problem, as the solution space is of size $O(2^n)$. It is hard to solve because it is a quadratic programming problem.

Now I introduce an efficient solution to this problem by converting it into a linear programming problem. The quadratic component in Eq.2.4 is $\mathbf{x}^\top \mathbf{B}\mathbf{x}$=0. From Eq.2.7, $\mathbf{B}$ is a real, symmetric and sparse $n \times n$ matrix. Therefore, $\mathbf{x}^\top \mathbf{B}\mathbf{x} = \sum_{i=1}^{n} b_{ii} x_i + 2\sum_{i=1}^{n}\sum_{j=i+1}^{n} b_{ij} x_i x_j$. By defining a variable y_{ij}=$x_i x_j$, we have $\mathbf{x}^\top \mathbf{B}\mathbf{x} = \sum_{i=1}^{n} b_{ii} x_i + 2\sum_{i=1}^{n}\sum_{j=i+1}^{n} b_{ij} y_{ij}$. Note that vector $\mathbf{x}$ is binary, therefore the objective function in Eq.2.4 can be expressed as a linear form $\|\mathbf{x}\|_0 + \lambda \mathbf{w}^\top \mathbf{x} = (\mathbf{u}^\top + \lambda\ \mathbf{w}^\top)\mathbf{x}$, where $\mathbf{u}$ is a unit vector. The original quadratic programming problem in Eq.2.4 can be converted to the following linear programming problem:

$$\begin{aligned} &\mathbf{min} \quad (\mathbf{u}^\top + \lambda \mathbf{w}^\top)\mathbf{x}, \\ s.t. &\begin{cases} \mathbf{A}\mathbf{x} \geq \mathbf{1}, \quad \mathbf{x} \in \{0,1\}^n, \\ \sum_{i=1}^{n} b_{ii} x_i + 2\sum_{i=1}^{n}\sum_{j=i+1}^{n} b_{ij} y_{ij} = 0, \\ y_{ij} \leq x_i, \forall i, \\ y_{ij} \leq x_j, \forall j, \\ x_i + x_j - y_{ij} \leq 1, \forall (i,j), \\ y_{ij} \geq 0, \forall (i,j), \end{cases} \end{aligned} \tag{2.9}$$

where the last four constraints impose $y_{ij} = x_i x_j$ for any binary vector $\mathbf{x}$. There are many techniques to solve this linear programming problem efficiently, such as CPLEX, Lingo, and constraint relaxation.

Algorithm 1: MNCD(S, ψ)

Input: A shape, S, and a concavity tolerance, ψ;
Output: ψ-MNCD of S, $\{P_i\}$.

```
◇ compute the candidate cut set, C(S);
◇ compute ψ-mutex set of S→M^ψ(S);
foreach mp_i in M^ψ(S) do
    foreach cut_j in C(S) do
        check whether cut_j separates mp_i → a_ij;
foreach cut_i in C(S) do
    compute its cost → w_i;
    foreach cut_j in C(S) do
        check whether cut_i intersects with cut_j → b_ij;
◇ obtain the optimized solution by solving Eq.2.9→ {P_i}.
```

2.3.3 Implemental Details and Time Complexity

Algorithm 1 shows the overall procedure of my method. In the implementation of computing the $C(S)$, to save the memory, I discard invalid cuts such as those whose endpoints are both convex points. As I consider all pairs of vertices on the contour, the time complexity of computing $C(S)$ is $O(v^2)$, where v is the number of vertices. According to [33], the time complexity of computing $M^{\psi}(S)$ is $O(tvr)$, where r is the number of notches in the shape, and t is the number of Morse functions we used to compute $M^{\psi}(S)$. With $C(S)$ and $M^{\psi}(S)$, I can obtain the value of matrixes A, B in $O(mn+n^2)$ time, where m is the number of mutex pairs in $M^{\psi}(S)$ and n is the number of candidate cuts in $C(S)$, where $n \gg m, t, v$, and r. Thus the total time complexity of formulating the problem is $O(v^2 + tvr + mn + n^2)$=$O(n^2)$. Solving the problem is the most time-consuming part of the decomposition, which takes several seconds.

2.3.4 Comparison with Other Methods

Table 2.1 presents a comparison between MNCD and the state-of-the-art methods: ACD [28] and CSD [33]. My method aims at the minimum number of parts with high visual naturalness for robust shape representation.

Specifically, CSD is a special case of my formulation in Eq.2.4 if discarding the $\|\mathbf{x}\|_0$ term and setting $\beta = 0$ in Eq.2.5. The $\|\mathbf{x}\|_0$ term in my formulation guarantees the minimum number of decomposed parts, which eliminates all the redundant parts in near-convex decomposition. This point is essential for robust shape representation and can improve the efficiency of further processes, as shown in Fig.1.1. Parameter β in Eq.2.5 imposes the minima rule and short cut rule on my near-convex decomposition scheme. Setting $\beta = 0$ means discarding the minima rule. This point is essential as well because these

	ACD [28]	CSD [33]	MNCD
Objective	a NCD without optimization	a NCD with the minimum length of cuts	a NCD with minimum number of parts and high visual naturalness
Candidate cut set	complete set of all possible cuts	incomplete set from Reeb graph	complete set of all possible cuts
Perception rules	minima rule and short cut rule	short cut rule	minima rule and short cut rule
Constraints	non-overlapping constraint convexity constraint	convexity constraint	non-overlapping constraint convexity constraint

Table 2.1: The comparison among ACD, CSD and MNCD, where NCD denotes near-convex decomposition.

two perception rules are introduced for high visual naturalness which guarantees better recognition primitives. And the minima rule inhibits cuts at positions with small negative curvatures or even at convex points.

2.4 Experiments of Shape Decomposition

2.4.1 2D Shape Decomposition

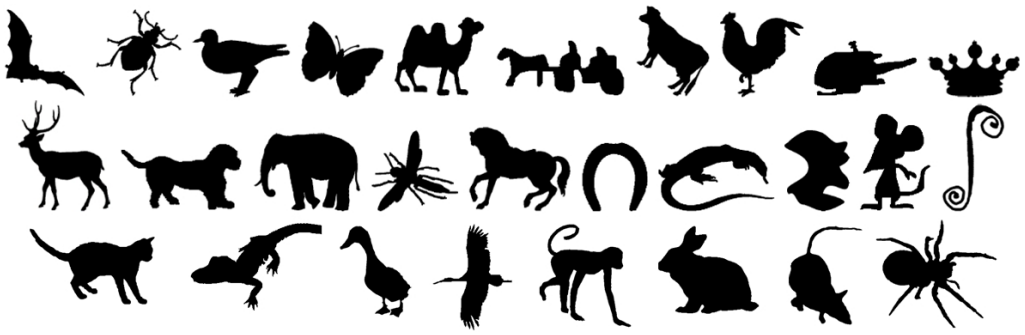


Figure 2.3: An example of each shape category selected from the MPEG-7 dataset [25] (the first two rows) and the Animal dataset [5] (the third row) is displayed.

In order to evaluate my shape decomposition method Minimum Near-Convex Decomposition (MNCD) on 2D shapes, I test the MPEG-7 shape dataset [25] and the Animal dataset [5]. Excluding simple shapes such as the heart shape that can be easily decomposed, I select 20 complex shape categories from MPEG-7 dataset, in which each category has 20 shapes (20×20=400 shapes), and 8 complex shape categories from Animal dataset, in which each category has 100 shapes (8×100=800 shapes). Fig.2.3 shows an image for each selected category.

Evaluation of parameters

In my algorithm, there are 2 parameters, ψ and λ, where ψ is the user specified concavity tolerance for near-convex decomposition; λ is the parameter introducing the visual naturalness.

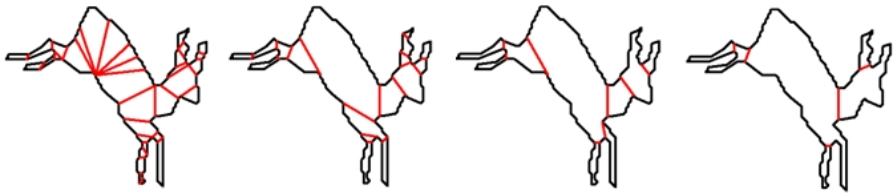


Figure 2.4: The decomposition results by MNCD, with ψ= 0.005R, ψ=0.01R, ψ=0.03R and ψ=0.06R, from left to right, respectively, where R is the radius of the shape's minimum enclosing disk.

The parameter ψ tells how small degree of concave features the user want to ignore in near-convex decomposition. Fig.2.4 shows the decomposition results at different value of ψ. A very small ψ means that the decomposed parts are almost strictly convex, which will introduce a large number of small parts to ensure the convexity constraint, thus is not robust to local distortions. When ψ increases, the decomposition can tolerate more severe distortions.

Figure 2.5: The decomposition results of MNCD when ψ=0.03R, with λ= 0, λ=0.5/$\sum_{i=1}^{n} w_i$, λ=1/$\sum_{i=1}^{n} w_i$, from left to right, respectively.

The parameter λ introduces the visual naturalness to the decomposition in Eq.2.4. Fig.2.5 shows the decomposition results of MNCD at different values of λ. If $0 \leq \lambda \leq 1/\sum_{i=1}^{n} w_i$, the number of parts by MNCD is minimized. But a larger λ brings a more natural decomposition as it counts more weight of the visual naturalness term in Eq.2.4. In my experiments below, I use λ=1/$\sum_{i=1}^{n} w_i$.

Evaluation of the number of parts

One advantage of my method is that it does not introduce redundant part as it decomposes the shape into minimum number of parts. In terms of the number of parts, table 2.2 presents the *average reduction rate* comparing my method with ACD [28] and CSD [33] at 4 different ψ, on MPEG-7 dataset and the Animal dataset respectively. The average reduction rate scores are defined as:

MPEG-7 dataset	ψ=0.005R		ψ=0.01R		ψ=0.03R		ψ=0.06R	
	ACD↓	CSD↓	ACD↓	CSD↓	ACD↓	CSD↓	ACD↓	CSD↓
bat	14.3%	8.9%	20.8%	11.3%	16.2%	6.8%	8.6%	6.5%
beetle	23.8%	10.3%	22.9%	9.0%	21.9%	16.0%	19.3%	14.4%
bird	18.5%	13.6%	23.8%	12.5%	12.8%	7.6%	17.4%	10.6%
butterfly	4.4%	5.8%	13.1%	7.2%	16.9%	8.8%	32.7%	12.9%
camel	16.1%	10.5%	15.2%	3.3%	21.1%	9.5%	21.3%	4.8%
carriage	5.5%	3.7%	13.8%	9.2%	15.6%	9.5%	18.4%	13.3%
cattle	24.9%	14.6%	24.5%	10.7%	27.4%	8.9%	23.0%	12.3%
chicken	19.0%	10.0%	23.1%	15.2%	24.0%	10.5%	3.1%	5.2%
chopper	8.9%	7.7%	16.2%	10.4%	22.1%	10.7%	17.4%	11.3%
crown	16.0%	9.2%	20.7%	11.9%	27.8%	14.6%	19.4%	16.7%
deer	18.0%	14.5%	24.2%	10.5%	15.3%	4.2%	22.6%	13.3%
dog	23.8%	15.4%	18.8%	7.6%	24.5%	9.2%	15.7%	10.5%
elephant	24.1%	12.0%	24.0%	8.9%	24.9%	9.7%	25.2%	7.8%
fly	11.9%	9.2%	8.9%	5.6%	4.2%	3.9%	10.6%	8.4%
horse	20.1%	8.0%	23.8%	5.1%	19.8%	1.1%	18.8%	6.1%
horseshoe	26.1%	18.6%	21.9%	11.7%	23.5%	14.8%	12.2%	12.2%
lizard	18.2%	10.4%	15.9%	10.0%	27.5%	15.2%	11.7%	7.3%
Misk	29.8%	30.7%	24.2%	11.9%	25.8%	20.3%	13.2%	15.4%
Mickey	24.6%	13.4%	14.0%	10.5%	19.8%	12.9%	17.3%	8.5%
spring	22.6%	12.6%	25.1%	13.7%	24.5%	15.8%	25.7%	6.9%

Table 2.2: The average reduction rate of MNCD comparing with ACD [28] and CSD [33], on the MPEG-7 dataset, where R is the radius of the shape's minimum enclosing disk.

Animal dataset	ψ=0.005R		ψ=0.01R		ψ=0.03R		ψ=0.06R	
	ACD↓	CSD↓	ACD↓	CSD↓	ACD↓	CSD↓	ACD↓	CSD↓
cat	16.7%	8.7%	24.4%	12.6%	21.9%	10.2%	22.8%	11.1%
crocodile	15.7%	14.4%	22.3%	15.9%	21.7%	11.2%	22.5%	5.9%
duck	21.1%	12.5%	25.2%	13.2%	19.0%	6.4%	15.1%	8.0%
flyingbird	7.1%	3.7%	13.2%	7.9%	21.1%	11.8%	15.6%	8.2%
monkey	18.5%	9.9%	25.5%	4.5%	21.2%	12.7%	30.5%	7.0%
rabbit	24.0%	9.4%	23.9%	6.3%	20.0%	4.9%	18.0%	11.1%
rat	24.2%	7.3%	23.5%	10.7%	12.2%	8.6%	15.3%	12.9%
spider	11.4%	5.1%	11.3%	6.2%	4.7%	5.2%	3.1%	4.1%

Table 2.3: The average reduction rate of MNCD comparing with ACD [28] and CSD [33], on the Animal dataset, where R is the radius of the shape's minimum enclosing disk.

$$\text{ACD} \downarrow = (\#\text{ACD} - \#\text{MNCD})/\#\text{ACD} \ ,$$
$$\text{CSD} \downarrow = (\#\text{CSD} - \#\text{MNCD})/\#\text{CSD} \ .$$

As it shows, my algorithm produce the least number of parts. Comparing with ACD [28], up to 32.7% of redundant parts are eliminated, and up to 30.7% of redundant parts are eliminated compared with CSD [33]. On average, compared with ACD 18.99% of parts are eliminated and 10.15% compared with CSD. Thus, the efficiency of further applications on the decomposed parts can be highly improved. On the other hand, from the table, we notice that all the ACD↓ and CSD↓ scores are greater than 0 on every shape category and every ψ, which means that MNCD always produces minimum number of parts, as proved in Theorem 1.

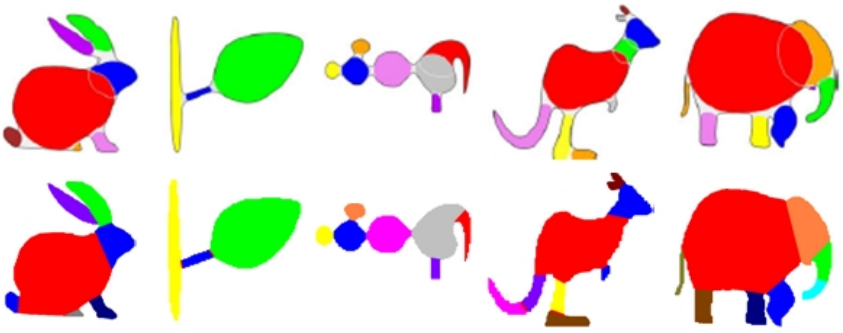


Figure 2.6: The first row shows the decomposition results of [36], and the second row shows the results of MNCD.

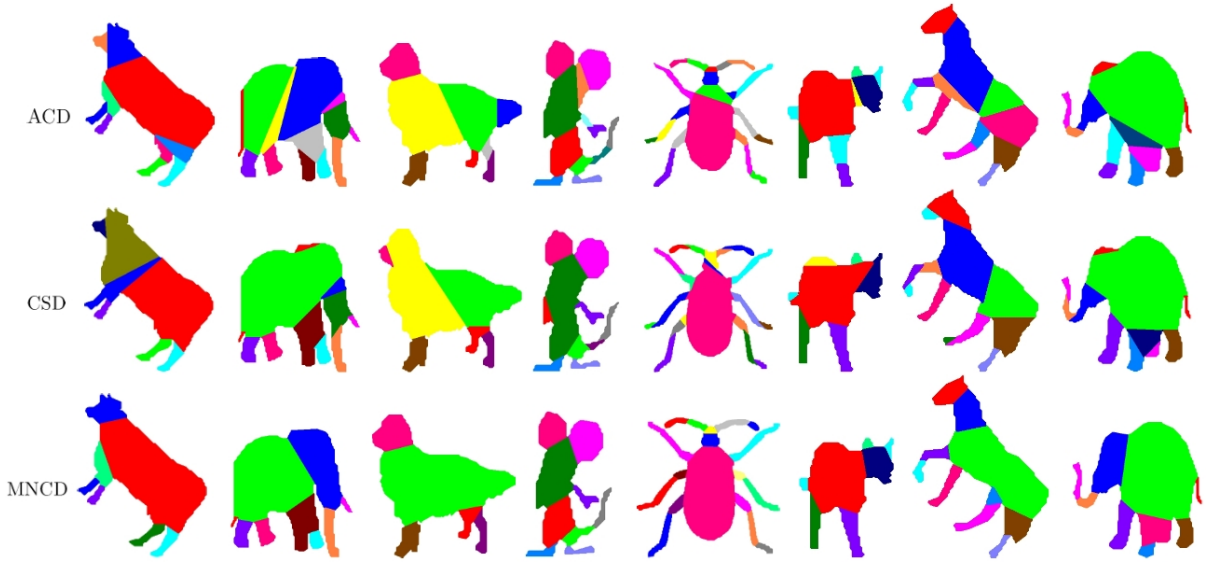


Figure 2.7: Some decomposition results of ACD [28], CSD [33] and MNCD. The MNCD method produces the least number of near-convex parts and the decompositions are visually more natural.

Decomposition results

To further evaluate the visual naturalness of my decomposition, Fig.2.6 compares my method with the method proposed by Mi and Decarlo [36]. Mi's method is specifically designed to decompose 2D shapes into natural parts. The first row are the decomposition results of their method, and the second row are the results of MNCD. As we can see, when considering the minima rule and short cut rule in my formulation, my method decomposes shapes into parts with high visual naturalness comparable to [36], such as the legs, head and body of the animal, the leaf and stem of the tree, etc.

In Fig.2.7, more comparisons among ACD [28], CSD [33] and my method are provided, with ψ=0.03R. The decompositions using my method produce the minimum and most natural recognition primitives.

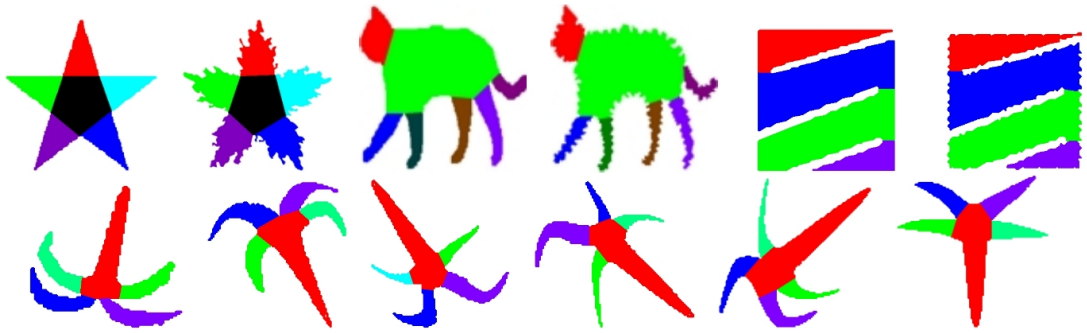


Figure 2.8: The robust decomposition results of MNCD. The first row is the results of shapes with local distortions; the second row is the results of shapes with deformation. Without introducing redundant parts and by considering perception rules, MNCD is robust to local distortions and shape deformation.

At this concavity tolerance, MNCD decomposes the animals into primitives such as head, body, legs and tail, and avoids decomposing them into redundant parts as in [28] [33].

Without introducing redundant parts, MNCD is robust to local distortions, as shown in the first row of Fig.2.8. The robustness of my method is more obvious when there are large local distortions as shown in the last row of Fig.1.1, while the existing decomposition methods produce many redundant noisy parts. Besides, my MNCD algorithm imposes two perception rules to guide the decomposition, thus it produces more natural parts, which makes MNCD robust to shape deformation, as illustrated in the second row of Fig.2.8.

2.4.2 3D Shape Decomposition

For 3D shapes, the decomposition is formulated the same way as for 2D shapes in Eq.2.4. Therefore, to decompose a 3D shape, we also need to compute the ψ-mutex set $M^{\psi}(S)$ and the candidate cut set $C(S)$. I follow the way proposed in [33], which proposed to obtain the 3D shape features by projecting the 3D shape into 2D planes multiple times. With t height planes evenly sampled in the space, the 3D shape S can be projected into t point clusters O_t. For each O_t, $M^{\psi}(O_t)$ and $C(O_t)$ can be computed the same way as the 2D cases. Finally we can obtain $M^{\psi}(S)$ and $C(S)$ from $M^{\psi}(O_t)$ and $C(O_t)$, and decompose the shape S by Eq.2.4.

Fig.2.9 shows some decomposition results of the shapes from Mcgill 3D Shape Benchmark [46]. As it shows, MNCD can decompose 3D shapes into parts with high visual naturalness, such as the fin of the whale, the legs of the bear, the lenses of the glasses, the fingers of the hand, etc. The last row in Fig.2.9 illustrates the robustness of MNCD decompositions for 3D human postures. Although the human body varies significantly with different postures, the MNCD decomposition results are stable.

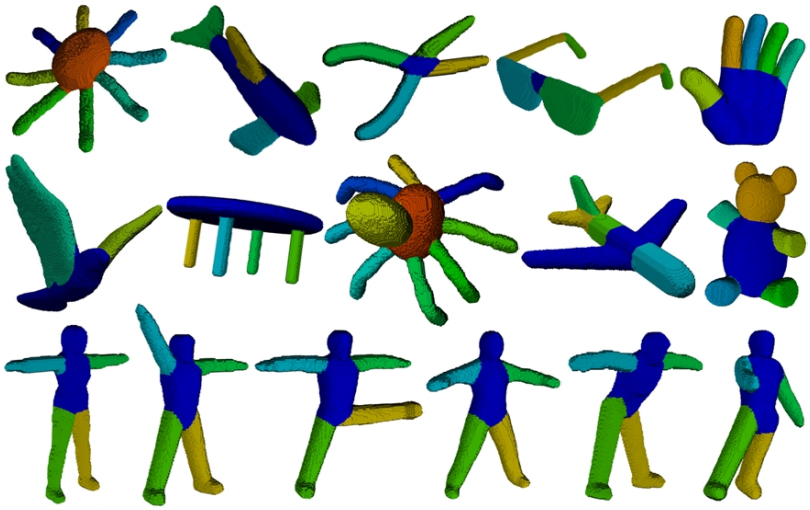


Figure 2.9: The 3D shape decompositions results of MNCD. The last row illustrates the robustness of my method to shape deformation.

In the next section, I present a novel hand gesture recognition system based on my part-based hand shape representation, using the Kinect sensor.

Chapter 3

Part-based Hand Gesture Recognition on top of Finger-Earth Mover's Distance

3.1 Introduction

Despite lots of previous work, traditional vision-based hand gesture recognition methods [11] [50] [15] [37] are still far from satisfactory for real-life applications. Because of the nature of optical sensors and the scene complexity, the quality of the captured images is affected by lighting conditions and cluttered backgrounds, thus these methods usually cannot detect and track the hands robustly, which largely affects the performance of hand gesture recognition. All existing vision-based hand gesture recognition methods have constraints on the users or the environment, which greatly hinders its widespread use in real-life applications. On one hand, to infer the pose of the palm and angles of joints, many methods use colored markers to extract high-level features, such as the fingertip, joint locations or some anchor points on the palm [11] [42] [50]. However, a common problem with these methods is the inaccurate hand segmentation: none of these methods operates well in cluttered environments due to the sensitivity of colored markers (skin color model) to the background. On the other hand, a few studies try to first fully reconstruct 3D hand surfaces [8] [13] [31]. Even though the 3D data provides valuable information that can handle problems like self-occlusion, an accurate, real-time and robust 3D reconstruction is still very difficult. Furthermore, the high computational cost forbids its widespread adoption.

To enable more robust hand gesture recognition, one effective way is to use other sensors to capture the hand gesture and motion, e.g. through the data glove [17]. Unlike optical sensors, such sensors are usually more reliable and are not affected by lighting conditions or cluttered backgrounds. However, as it requires the user to wear a data glove and sometimes requires calibration, it is inconvenient for the

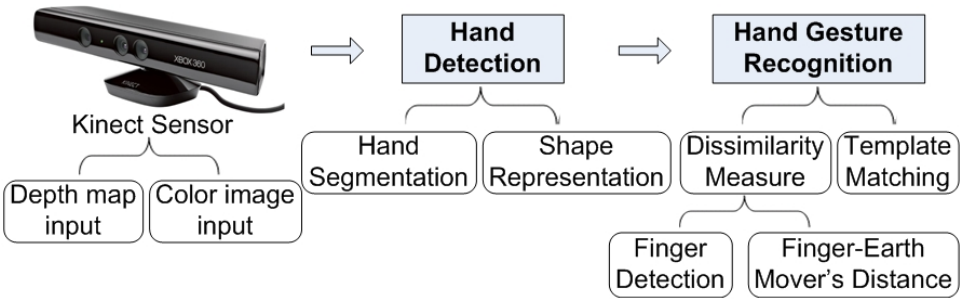


Figure 3.1: The framework of my part-based hand gesture recognition system.

user and may hinder the natural articulation of hand gesture. Also, such data gloves are usually much more expensive than other sensors. As a result, it is not a very popular way for hand gesture recognition.

Thanks to the recent development of inexpensive Kinect sensor [43], new opportunities for hand gesture recognition emerge. In spite of many recent successes in applying the Kinect sensor to face recognition [9] and human body tracking [43], it is still an open problem to use Kinect for hand gesture recognition. Due to the low-resolution of the Kinect depth map, at only 640×480, although Kinect works well to track a large object, e.g. the human body, it is difficult to detect and segment a small object from an image with this resolution, e.g., a human hand which occupies a very small portion of the image with more complex articulations. In such a case, the segmentation of the hand is usually inaccurate and noisy, thus may significantly affect the recognition step. However, robust shape recognition under distortions is challenging. Classic shape recognition methods are not robust to severe distortions in hand shapes. For instance, contour-based recognition approaches, such as moments, are not robust when the contour is disturbed by local distortions. Skeleton-based recognition methods [44] also suffer from contour distortions, because even little noise or slight variations in the contour often severely perturb the topology of its skeletal representation. Bai *et al*. propose a skeleton pruning method in [4], which makes skeleton robust to contour noises. However, skeleton-based methods still cannot deal with the ambiguity problem as shown in Fig.1.2, since the second and the third skeleton have more similar structure than the first two shapes. As for the correspondence-based shape recognition methods such as shape contexts [6] and inner-distance [32], they are not effective in solving the ambiguity in Fig.1.2 either, because the correspondences of the second and the last hands have more similar contexts than the first and the second one do.

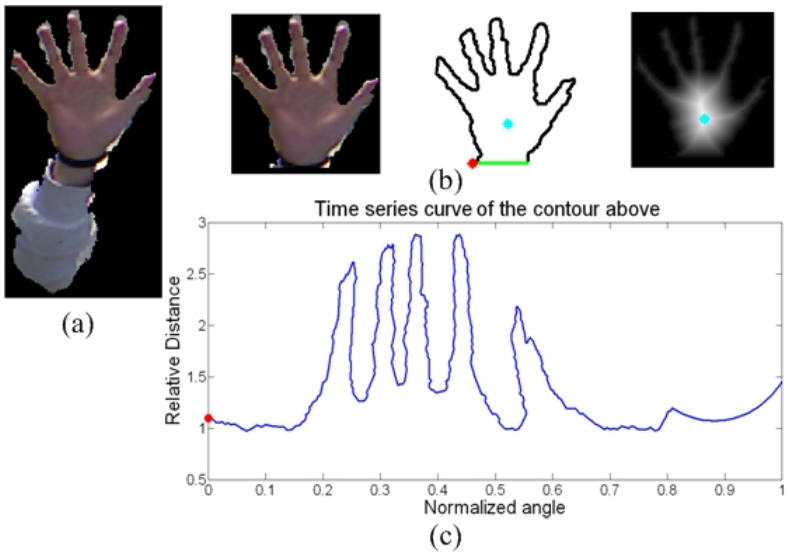


Figure 3.2: Hand detection (better viewed in color). (a) The rough hand segmented by depth thresholding; (b) A more accurate hand detected with black belt (the green line), the initial point (the red point) and the center point (the cyan point); (c) Its time-series curve representation.

3.2 Part-based Hand Gesture Recognition

I propose to use the part-based representation for hand gesture recognition, which combines the global structure and local information. As discussed before, the global structure of the node-graphs in Fig.1.2 may confuse the recognition. However, when I add local information into each node, such challenges can be solved. The core of my hand gesture recognition scheme is the dissimilarity distance metric that combines the global information and local information, called Finger-Earth Mover's Distance (FEMD).

Now I introduce the scheme of my part-based hand gesture recognition system with the Kinect sensor. Fig.3.1 illustrates the framework, which consists of two major modules: hand detection and hand gesture recognition.

3.2.1 Hand Detection and Representation

As shown in Fig.3.1, I use the Kinect sensor as the input device, which captures the color image and the depth map at 640×480 resolution. Generally the depth information derived from Kinect sensor is usable but not very accurate in details.

In order to segment the hand shape, I require the user to cooperate in two aspects (both are reasonable requirements in HCI): first, the user need to make sure that the hand is the frontmost object facing the sensor. Thus, by thresholding from the nearest depth position with a certain gap, a rough hand region

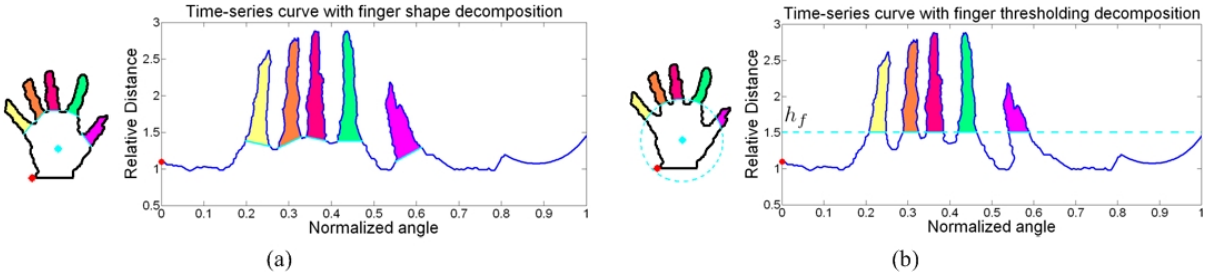


Figure 3.3: Illustration of two finger detection methods in hand shape and its time-series curve. (a) is near-convex decomposition, (b) is thresholding decomposition.

can be obtained, as shown in Fig.3.2(a). Second, the user need to wear a black belt on the gesturing hand's wrist. I use RANSAC to locate the position of the black belt, and thus, a more precise hand shape can be detected, as shown in Fig.3.2(b). The hand shape is generally of around 100×100 pixel resolution, with possibly severe distortions.

After detecting the hand shape, I represent it as a *time-series curve*, as shown in Fig.3.2(c). Such a shape representation has been successfully used for the classification and clustering of shapes [23]. The time-series curve records the relative distance between each contour vertex and a center point. I define the center point as the point with the maximal distance after Distance Transform on the shape (the cyan point), as shown in Fig.3.2(b); and the initial point (the red point) is defined according to the RANSAC line detected from the black belt (the green line).

In my time-series representation, the horizontal axis denotes the angle between each contour vertex and the initial point relative to the center point, normalized by 360°. The vertical axis denotes the Euclidean distance between the contour vertices and the center point, normalized by the radius of the maximal inscribed circle. As shown in Fig.3.2, the time-series curve captures nice topological properties of the hand, such as the fingers.

3.2.2 Finger Detection

In order to measure the dissimilarity distance between two hand shapes, I represent the hand shape as a signature with each finger as a cluster. In Fig.3.3, I propose two finger detection methods to obtain the finger clusters from the hand shapes. Now I introduce these two algorithms:

3.2.2.1 Near-convex shape decomposition

I note that the fingers have geometric properties. They are near-convex parts of the hand shape. Therefore, I adjust the Minimum Near-Convex Decomposition (MNCD) described in Section 2.2.2 to a finger detection method, which is illustrated in Fig.3.3(a):

$$\begin{aligned} \min \quad & \alpha \parallel \mathbf{x} \parallel_{\mathbf{0}} + (\mathbf{1} - \alpha)\ \mathbf{w}^{\top}\mathbf{x}, \\ s.t. \quad & \mathbf{Ax} \geq \mathbf{1},\ \mathbf{x}^{\top}\mathbf{Bx} = 0,\ \mathbf{x} \in \{0,1\}^{n}. \end{aligned} \tag{3.1}$$

The goal of the first term in the objective function is to reduce the redundant parts that are not fingers, and the second term is to improve the visual naturalness of the decomposition. Parameter α balances the influence between the first and the second term. This problem can be solved in the same way as in Eq.2.4 with $\lambda = (1-\alpha)/\alpha$. I will investigate the effects of α in Section 3.3.3.

3.2.2.2 Thresholding decomposition

Although it is accurate to obtain the part information using near-convex shape decomposition method, Eq.3.1 is complex to solve. I propose an alternative finger detection method. As mentioned before, the time-series curve reveals a hand's topological information well. As shown in Fig.3.3(b), each finger corresponds to a peak in the curve. Therefore, I can apply the height information in time-series curve to decompose the fingers. Specifically, I define a finger as a segment in the time-series curve, whose height is greater than a threshold h_f. In this way, I can detect the fingers fast. However, choosing a good height threshold h_f is essential. I will investigate the effects of h_f in Section 3.3.3.

3.2.3 Hand Gesture Recognition

After representing each hand shape as a signature with each finger as a cluster, I use template matching for robust recognition, i.e., the input hand is recognized as the class with which it has the minimum dissimilarity distance: $c = \arg\min_{c} \text{FEMD}(H,\ T_c)$, where H is the input hand; T_c is the template of class c; $\text{FEMD}(H,\ T_c)$ denotes the proposed Finger-Earth Mover's Distance between the input hand and each template. Now I introduce the main idea of FEMD.

3.2.3.1 Finger-Earth Mover's Distance

In [41], Rubner et al. presented a general and flexible metric, called Earth Mover's Distance (EMD), to measure the distance between signatures or histograms. EMD is widely used in many problems such as content-based image retrieval and pattern recognition.

EMD is a measure of the distance between two probability distributions. It is named after a physical analogy that is drawn from the process of moving piles of earth spread around one set of locations into another set of holes in the same space. The locations of earth piles and holes denotes the mean of each cluster in the signatures, the size of each earth pile or hole is the weight of cluster, and the ground

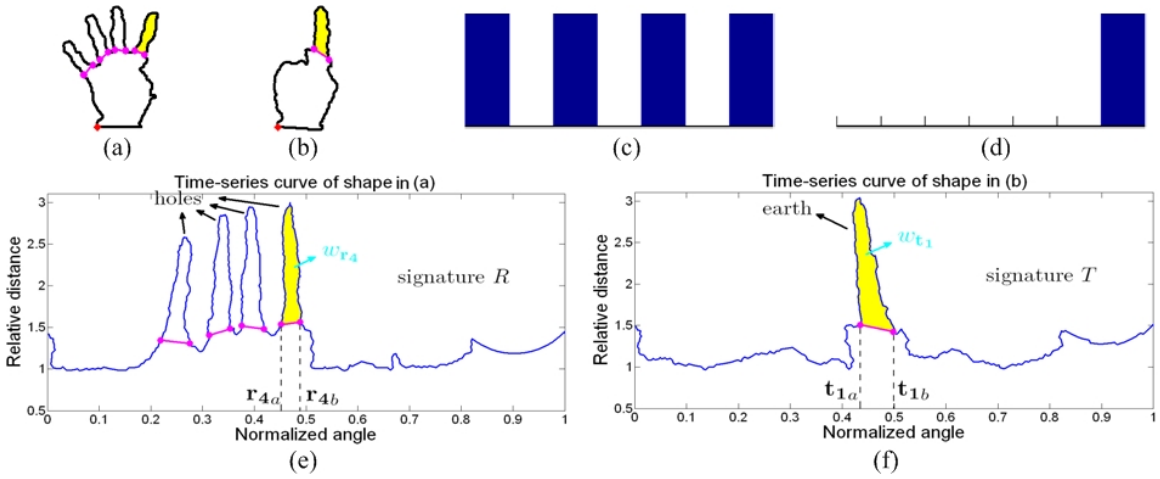


Figure 3.4: (a) (b): two hand shapes whose time-series curves are shown in (e) (f). (c) (d): two signatures that partially match, whose EMD cost is 0. (e) (f): illustration of the signature representations of time-series curves.

distance between a pile and a hole is the amount of work needed to move a unit of earth. To use this transportation problem as a distance measure, i.e., a measure of dissimilarity, one seeks the least costly transportation — the movement of earth that requires the least amount of work.

Grauman and Darrell applied EMD to contour matching and contour retrieval [12], which represents the contour by a set of local descriptive features and computes the set of correspondences with minimum EMD costs between the local features. However, the existing EMD-based contour matching algorithms have two deficiencies when applied to hand gesture recognition:

1. Two hand shapes differ mainly in global features while not local features. As shown in Fig.3.4(a)(b), the fingers (global features) are their major difference. Besides, the large number of local features slows down the speed of contour matching. Therefore, it is better to consider global features in contour matching.

2. EMD allows for partial matching, i.e., a signature and its subset are considered to be the same in EMD measure: as in Fig.3.4(c)(d), the EMD distance of these two signatures is zero because the signature in Fig.3.4(d) is a subset of Fig.3.4(c). However, in many situations partial matching is illogical, such as in the case of Fig.3.4(a)(b), where the finger in Fig.3.4(b) is a partial set of the fingers in Fig.3.4(a). Clearly, they should be considered different.

The Finger-Earth Mover's Distance (FEMD) can address these two deficiencies of the contour matching methods using EMD. Different from the EMD-based algorithm which considers each local feature as

a cluster [12], I represent the input hand by global features (the finger clusters). And I add penalty on empty holes to alleviate partial matches on global features.

Formally, let R=$\{(\mathbf{r}_1, w_{\mathbf{r}_1}), ..., (\mathbf{r}_{\overline{m}}, w_{\mathbf{r}_{\overline{m}}})\}$ be the first hand signature with $\overline{m}$ clusters, where $\mathbf{r}_i$ is the cluster representative and $w_{\mathbf{r}_i}$ is the weight of the cluster; T=$\{(\mathbf{t}_1, w_{\mathbf{t}_1}), ..., (\mathbf{t}_{\overline{n}}, w_{\mathbf{t}_{\overline{n}}})\}$ is the second hand signature with $\overline{n}$ clusters. Now I show how to represent a time-series curve as a signature. Fig.3.4(e)(f) show the time-series curves of the hands in Fig.3.4(a)(b) respectively, where each finger corresponds to a segment of the curve. I define each cluster of a signature as the finger segment of the time-series curve: the representative of each cluster $\mathbf{r}_i$ is defined as the angle interval between the endpoints of each segment, $\mathbf{r}_i$=$[\mathbf{r}_{ia},\ \mathbf{r}_{ib}]$, where $0 \leq \mathbf{r}_{ia} < \mathbf{r}_{ib} \leq 1$; and the weight of a cluster, $w_{\mathbf{r}_i} \in (0, 1)$, is defined as the normalized area within the finger segment.

$\mathbf{D}$=$[d_{ij}]$ is the ground distance matrix of signature R and T, where d_{ij} is the ground distance from cluster $\mathbf{r}_i$ to $\mathbf{t}_j$. d_{ij} is defined as the minimum moving distance for interval $[\mathbf{r}_{ia},\ \mathbf{r}_{ib}]$ to totally overlap with $[\mathbf{t}_{ja},\ \mathbf{t}_{jb}]$, i.e.:

$$d_{ij} = \begin{cases} 0, & \mathbf{r}_i \text{ totally overlap with } \mathbf{t}_j, \\ \min(|\mathbf{r}_{ia} - \mathbf{t}_{ja}|,\ |\mathbf{r}_{ib} - \mathbf{t}_{jb}|), & \text{otherwise.} \end{cases}$$

For two signatures, R and T, their FEMD distance is defined as the least work needed to move the earth piles plus the penalty on the empty hole that is not filled with earth:

$$\begin{aligned} \text{FEMD}(R,\ T) &= \beta E_{move} + (1-\beta) E_{empty}, \\ &= \frac{\beta \sum_{i=1}^{\overline{m}} \sum_{j=1}^{\overline{n}} d_{ij} f_{ij} + (1-\beta) |\sum_{i=1}^{\overline{m}} w_{\mathbf{r}_i} - \sum_{j=1}^{\overline{n}} w_{\mathbf{t}_j}|}{\sum_{i=1}^{\overline{m}} \sum_{j=1}^{\overline{n}} f_{ij}}, \end{aligned}$$

where $\sum_{i=1}^{\overline{m}} \sum_{j=1}^{\overline{n}} f_{ij}$ is the normalization factor, f_{ij} is the flow from cluster $\mathbf{r}_i$ to cluster $\mathbf{t}_j$, which constitutes the flow matrix $\mathbf{F}$. Parameter β modulates the importance between the first and the second terms. I will investigate the effects of β in Section 3.3.3. As we can see, E_{empty}, d_{ij} are constants for the two signatures; to compute the FEMD, we need to compute the value of $\mathbf{F}$. $\mathbf{F}$ is defined by minimizing the work needed to move all the earth piles:

$$\mathbf{F} = \arg\min \text{WORK}(R,\ T,\ \mathbf{F}) = \arg\min \sum_{i=1}^{\overline{m}} \sum_{j=1}^{\overline{n}} d_{ij} f_{ij},$$

$$s.t. \begin{cases} f_{ij} \geq 0 \qquad 1 \leq i \leq \overline{m},\ 1 \leq j \leq \overline{n}, \\ \sum_{j=1}^{\overline{n}} f_{ij} \leq w_{\mathbf{r}_i} \quad 1 \leq i \leq \overline{m}, \\ \sum_{i=1}^{\overline{m}} f_{ij} \leq w_{\mathbf{t}_j} \quad 1 \leq j \leq \overline{n}, \\ \sum_{i=1}^{\overline{m}} \sum_{j=1}^{\overline{n}} f_{ij} = \min(\sum_{i=1}^{\overline{m}} w_{\mathbf{r}_i},\ \sum_{j=1}^{\overline{n}} w_{\mathbf{t}_j}). \end{cases}$$

I follow the definition of the flow matrix $\mathbf{F}$ in EMD, as I also intend to find the minimum work needed to move the earth piles. The first constraint restricts the moving flow to one direction: from earth piles to the holes. The last constraint forces the maximum amount of earth possible to be moved. I will demonstrate the superiority of FEMD over EMD for contour matching in Section 3.3.3.

3.3 Experiments of Hand Gesture Recognition

3.3.1 Dataset

I collect a new hand gesture dataset using Kinect sensor which contains both color images and depth maps. My dataset is collected from 10 subjects, and it contains 10 gestures as shown in Fig.3.5. Each subject performs 10 different poses for the same gesture. Thus in total my dataset has 10 people $\times$ 10 gestures/people $\times$ 10 cases/gesture $=$ 1000 cases, each of which consists of a color image and a depth map.

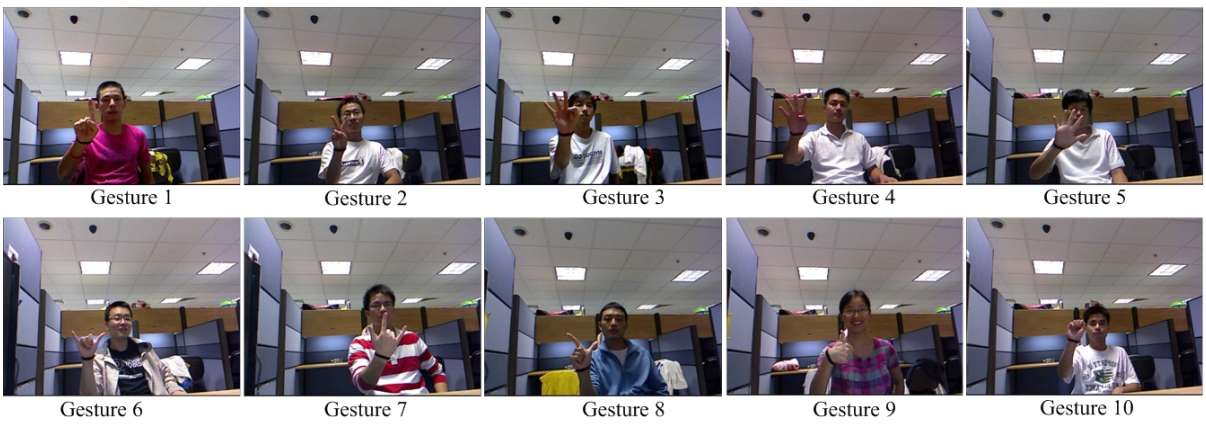
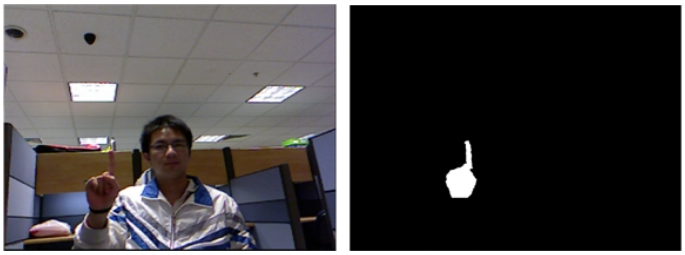
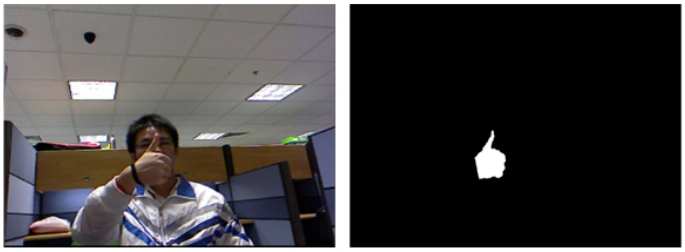


Figure 3.5: The color image examples for the 10 gestures in my dataset.

My dataset is a very challenging real-life dataset, which is collected in cluttered backgrounds. Besides, for each gesture, the subject poses with variations, namely the hand changes in orientation, scale, articulation, etc., as illustrated in Fig.3.5.



(a) The hand that is cluttered by background can be detected accurately;



(b) The hand that is cluttered by face can be detected accurately.

Figure 3.6: My system is robust to cluttered backgrounds.

3.3.2 Performance Evaluation

All experiments were done on a Intel Core^{TM} 2 Quad 2.66 GHz CPU with 3 GB of RAM. Now I evaluate the performance of my system from the following aspects:

3.3.2.1 Robustness to cluttered backgrounds

My hand gesture recognition system is robust to cluttered backgrounds, because the hand shape is detected using the depth information and thus the backgrounds can be easily removed. Fig.3.6(a) illustrates an example when the hand is cluttered by the background. It is hard for other hand gesture recognition methods that use colored markers to detect the hand. In Fig.3.6(b), it shows a difficult case for the skin color-based hand gesture recognition approaches, where the hand is cluttered by the user's face. However, the hand segmentation is very accurate using Kinect sensor, as shown in the right column of Fig.3.6.

3.3.2.2 Robustness to input variations and distortions

In real-life environment, a hand can have variations on orientation, scale and articulation. Besides, because of the limited resolution of the depth map, the hand shapes are always distorted, or ambiguous. However, I can demonstrate that the proposed dissimilarity distance metric, Finger-Earth Mover's Dis-

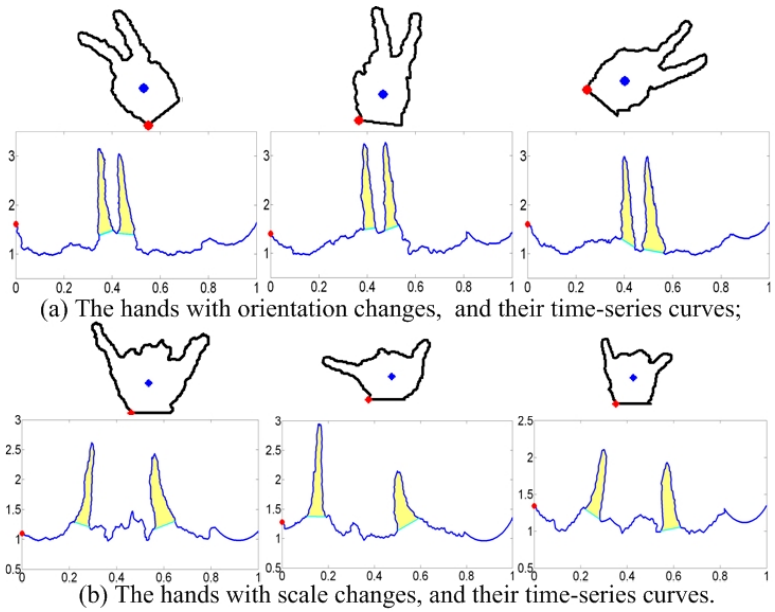


Figure 3.7: My method is robust to orientation and scale changes.

tance (FEMD), is not only robust to the orientation and scale changes of the hand, but also insensitive to distortions and articulations.

Fig.3.7(a) shows 3 hands with different orientations. As we can see, the initial point (the red point on the figure) and the center point (the blue point) are relatively fixed in these shapes. Thus the time-series curves of these hands (the second row in Fig.3.7(a)) are similar, and their distances are very small. In Fig.3.7(b), there are 3 hands of different size. Because the time-series curve and the FEMD distance are normalized, they are correctly recognized as the same gesture. Hence we can conclude that FEMD is robust to orientation and scale changes.

Furthermore, my hand gesture recognition method is robust to the articulations and distortions brought by imperfect hand segmentation. Since the proposed FEMD distance metric uses global features (fingers) to measure the dissimilarity, local distortions are tolerable. As for the articulations, Fig.3.8 shows some examples: the leftmost column shows 4 hand images of the same gesture; the middle column shows the corresponding hand shapes; and the rightmost column shows their time-series curves. As we can see, the hand shapes in Fig.3.8(c)(d) are heavily distorted. However, as illustrated in the rightmost column of Fig.3.8, by detecting the finger parts (the yellow regions), I represent each shape as a signature whose clusters are the finger parts. Particularly, the signatures of Fig.3.8(a)(b) have 2 clusters: $\{(\mathbf{r}_1,\ w_{\mathbf{r}_1}),\ (\mathbf{r}_2,\ w_{\mathbf{r}_2})\}$, and the signatures of Fig.3.8(c)(d) only have 1 cluster: $\{(\mathbf{t}_1,\ w_{\mathbf{t}_1})\}$. From Section 3.2.3.1, we can estimate that $(w_{\mathbf{r}_1}+w_{\mathbf{r}_2}) \approx w_{\mathbf{t}_1}$, and the ground distance $d_{11},\ d_{21} \approx 0$. According to the

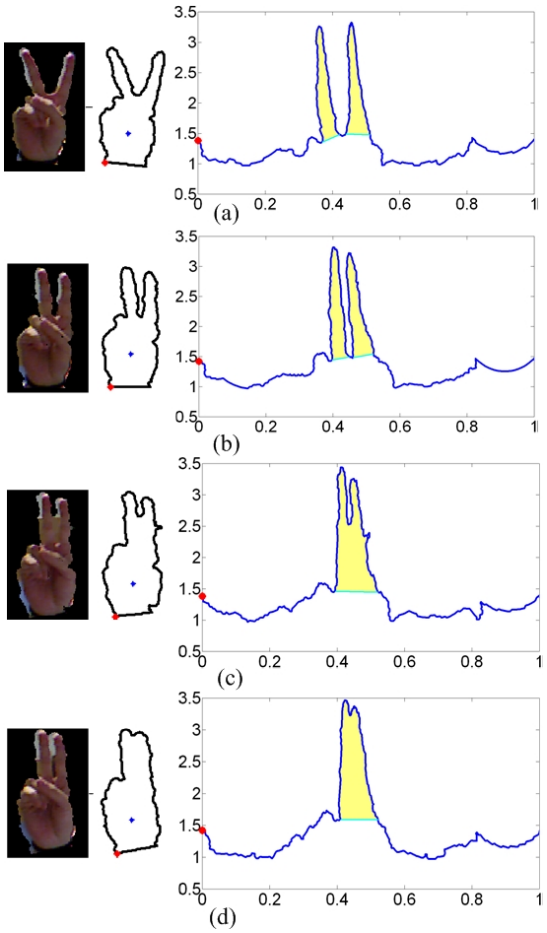


Figure 3.8: My system is insensitive to the distortions and articulation.

definition, we know that the FEMD distances among the 4 shapes ≈ 0. Therefore, my FEMD metric is insensitive to distortions and articulations.

3.3.2.3 Accuracy and efficiency

In order to evaluate the accuracy and efficiency of my system, two experiments are conducted on the new dataset. The first experiment uses thresholding decomposition to detect the finger clusters for FEMD measure, and the second experiment uses near-convex shape decomposition for finger detection.

Experiment I: Thresholding decomposition + FEMD

In experiment I, I fix the height threshold h_f=1.6 and the FEMD parameter β=0.5.

Fig.3.9 is the confusion matrix of experiment I. The mean accuracy is 90.6%. As it shows, the two

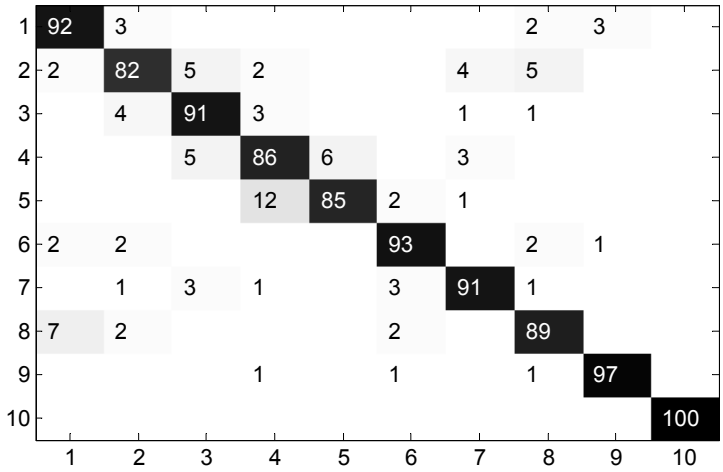


Figure 3.9: The confusion matrix of Experiment I.

most confused gesture categories are gesture 5 and 4, gesture 8 and 1. Fig.3.10 shows two confused cases of these categories.

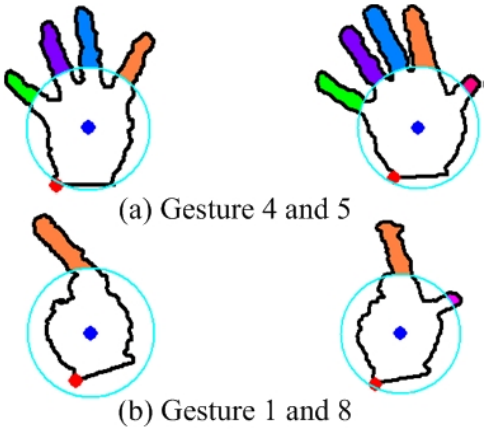


Figure 3.10: Two pairs of confusing gestures in Experiment I.

Because the thumb finger is shorter and smaller, if decomposing the hands only by a height thresholding, important finger regions may be lost in some cases. As shown in Fig.3.10, the thumb fingers are not well decomposed. As a result, the FEMD distances of these two cases are very small, which confuse the recognition.

However, thresholding decomposition is simple and fast. Besides, due to the few number of extracted global features, FEMD operates efficiently. Table 3.1 gives the mean running time of a hand recognition procedure in experiment I, 0.0750s. It should be noted that some parts of my code are coded in Matlab without optimization. As we see, thresholding decomposition based FEMD operates in real time.

	Mean Accuracy	Mean Running Time
Shape Context without bending cost [6]	83.2%	12.346s
Shape Context with bending cost [6]	79.1%	26.777s
Skeleton Matching [3]	78.6%	2.4449s
Thresholding Decomposition+FEMD	90.6%	**0.0750**s
Near-convex Decomposition+FEMD	**93.9**%	4.0012s

Table 3.1: The mean accuracy and the mean running time of Shape Contexts, Skeleton Matching, and my methods. My part-based hand gesture recognition system using FEMD outperforms the traditional shape matching algorithms.

Experiment II: Near-convex shape decomposition + FEMD

Figure 3.11: Finger Detection results of Experiment II using near-convex decomposition algorithm.

In order to more accurately decompose the fingers from the hands, I conduct another experiment which computes FEMD distance based on the proposed near-convex decomposition algorithm in Eq.3.1. Fig.3.11 shows some finger detection results of my near-convex decomposition algorithm. Here I fixed the near-convex decomposition parameter α=0.5 and the FEMD parameter β=0.5. As we see, by formulating the shape decomposition problem as a combinational optimization problem to improve the visual naturalness and inhibit noisy parts, the decomposition results are more accurate than thresholding decomposition.

Fig.3.12 shows the confusion matrix of experiment II. Compared with experiment I, there are no seriously confused categories. And we can see that the accuracy in all the classes are improved.

In the last row of Table 3.1, the mean accuracy and the mean running time of experiment II are given. The mean accuracy of experiment II (93.9%) is higher than that of experiment I (90.6%), owing to more accurate finger decomposition. In terms of the recognition error the proposed near-convex decomposition reduces the error from 9.4% for the thresholding decomposition to 6.1%, which is a 35% of relative error

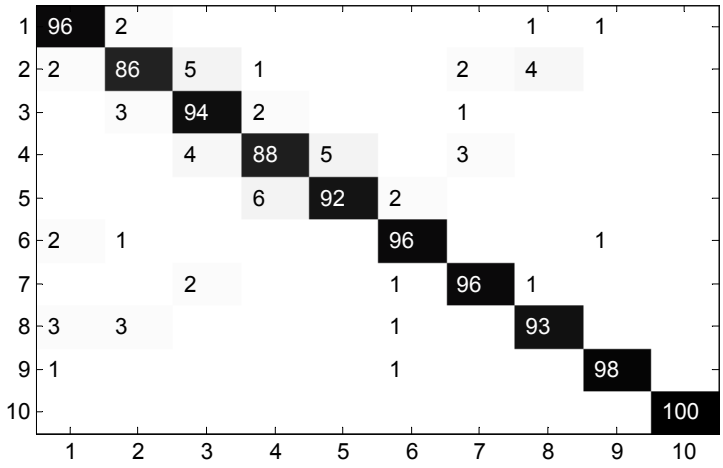


Figure 3.12: The confusion matrix of Experiment II.

reduction. But on the other hand, the speed of the second method is slower than that of the former one, because of the more complex finger detection algorithm.

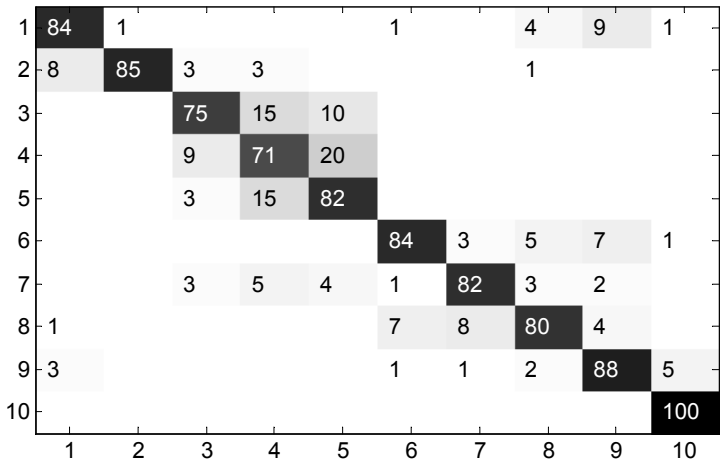
3.3.2.4 Comparison with Other Methods

FEMD is a part-based hand matching metric. I compare it with the traditional correspondence-based matching algorithm, Shape Context [6] and the skeleton-based matching algorithm, Path Similarity [3]. Their mean accuracy and running time are given in Table 3.1. I pre-segment the hand shape using the same method as I used in Section 3.2.1.

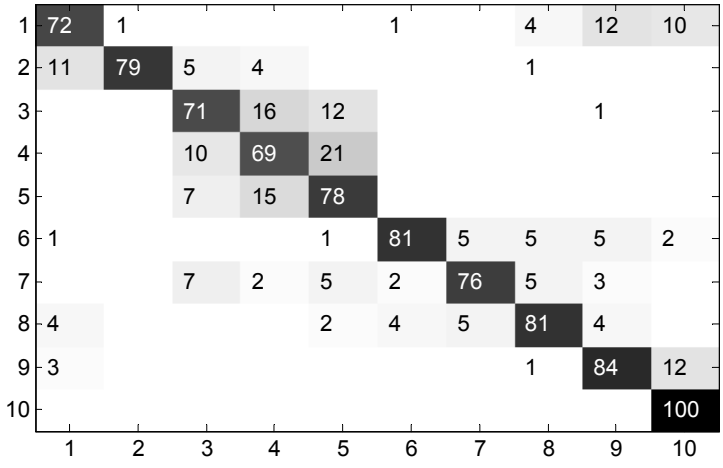
Fig.3.13 illustrates the confusion matrixes of Shape Context [6]. From both Fig.3.13(a) and (b), I find that the most confusing classes are gesture 3, 4, and 5. The reason is that fingers are more easily distorted in these classes, making them indistinguishable, which I have discussed before in Fig.1.2 and Fig.3.8. Fig.3.14 shows some confusing cases for shape context where shapes are locally distorted.

From the first two rows of Table 3.1, I notice that considering the bending cost of TPS transformation worsens the recognition performance. The reason is that in order to be rotation invariant, shape context needs to treat the tangent vector at each point as the positive axis for the log-polar histogram frame. However, since the shape is binary, a small variation on the shape could cause severe change of the tangent vectors at points on the shape. Thus adding TPS bending cost worsens the performance.

Fig.3.15 shows the confusion matrix of skeleton matching. I first prune the noisy skeleton using the method proposed in [4] and match them using Path Similarity proposed in [3]. From the figure, we notice that many gestures are severely confused, such as between gesture 1 and 9, gesture 6 and 8. The



(a)



(b)

Figure 3.13: The confusion matrix of hand gesture recognition using Shape Context [6]. (a) is the result without bending cost, and (b) is the result with bending cost.

Figure 3.14: Some confusing cases for shape context [6], where shapes are locally distorted.

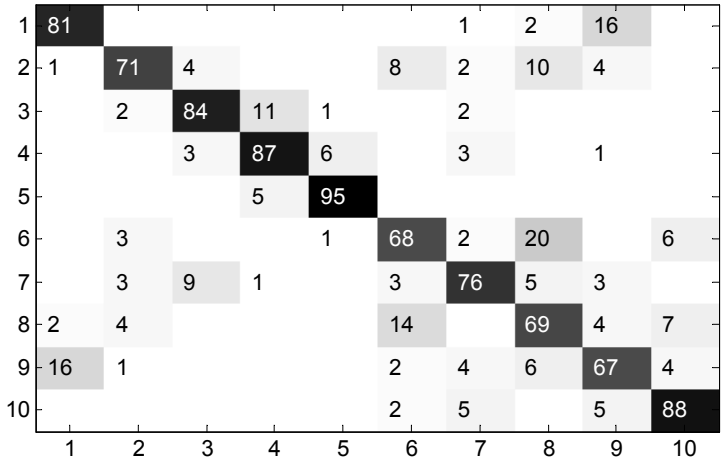


Figure 3.15: The confusion matrix of hand gesture recognition using skeleton matching [3].

Figure 3.16: Some confusing cases for skeleton matching [3], where very different shapes can lead to similar skeletons.

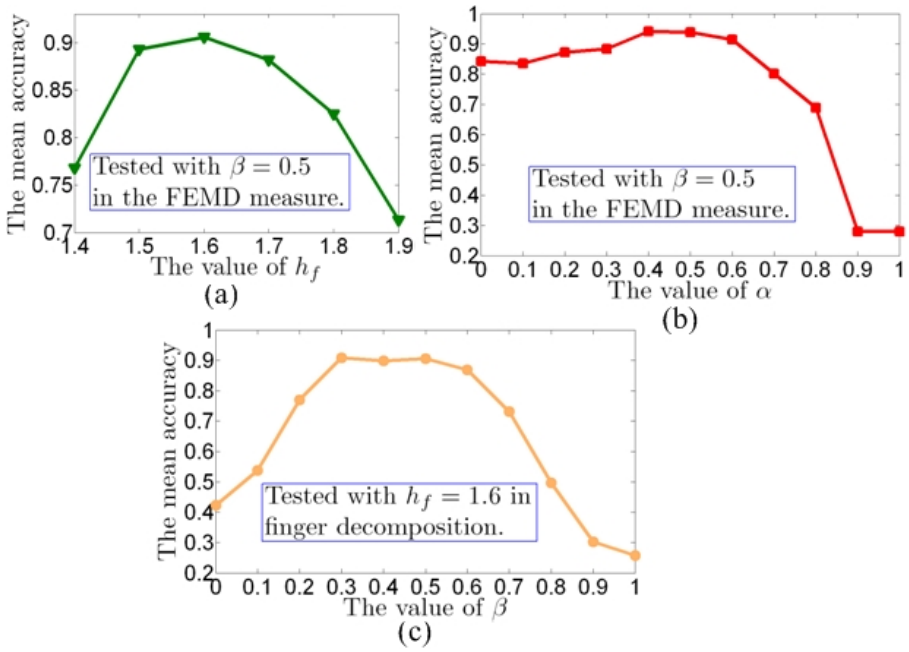


Figure 3.17: Parameter sensitivity on h_f, α and β. $\alpha = 0$ corresponds to the shape decomposition method in [33], and $\beta = 1$ corresponds to the EMD metric [41].

reason is that in those cases, their skeletons have very similar global structure, as shown in Fig.3.16. However, without local information, it is easy to confuse these classes.

3.3.3 Parameter sensitivity

In this section, I evaluate 3 important parameters — the height threshold h_f in thresholding decomposition based finger detection (Section 3.2.2.2), the parameter α in near-convex decomposition based finger detection(Eq.3.1), and the parameter β in FEMD formulation (Section 3.2.3.1).

The results are shown in Fig.3.17. In thresholding decomposition, h_f determines the radius of the decomposing circle (see Fig.3.3(b)). If h_f is too small (i.e., $h_f \leq 1.5$), the fingers cannot be well decomposed; and if h_f is too large (i.e., $h_f \geq 1.7$), essential finger regions will be lost. Fig.3.17(a) shows that we can obtain the best result if setting h_f around 1.6. In finger detection using near-convex decomposition, α balances the impact of the visual naturalness and the number of parts. As shown in Fig.3.17(b), if we only minimize the visual naturalness term (i.e., $\alpha = 0$), we will obtain noisy parts that affect the FEMD measure, which also justifies that my near-convex decomposition method can more accurately detect the fingers than [33] (the special case when $\alpha = 0$). Besides, the curve drops fast after $\alpha > 0.8$ because if minimizing the parts number too much while ignoring the visual naturalness, we may

obtain parts that are not fingers. As a result, the finger-based FEMD measure operates improperly. This also validates the importance of introducing the visual naturalness regularization term in my formulation. In the FEMD measure, β modulates importance between the earth-moving work E_{move} and the empty-hole penalty E_{empty}. Fig.3.17(c) shows that if either only considering E_{move} (i.e., $\beta = 1$) or only considering E_{emtpy} (i.e., $\beta = 0$), FEMD cannot measure correct dissimilarity between hand shapes. This curve also justifies that FEMD is better than EMD (the special case when $\beta = 1$) for dissimilarity measure between hand shapes.

The hand gesture dataset I collected with Kinect sensor (introduced in Section 3.3.1) is available at http://eeeweba.ntu.edu.sg/computervision/people/home/renzhou/HandGesture.htm.

Chapter 4

Applications in Human-Computer-Interaction

Lately there has been a great emphasis on Human-Computer-Interaction (HCI) research to create easy-to-use interfaces by employing natural communication and manipulation skills of humans. Among different human body parts, the hand is the most effective interaction tool, because of its dexterity. Adopting hand gesture as an interface in HCI will not only allow the deployment of a wide range of applications in sophisticated computing environments such as virtual reality systems and interactive gaming platforms, but also benefit our daily life such as providing aids for the hearing impaired, and maintaining absolute sterility in health care environment using touchless interfaces via gestures [52].

Currently, the most effective tools to capture the hand gesture are electro-mechanical or magnetic sensing devices (data gloves) [17]. These methods employ sensors attached to a glove that transduces finger flexions into electrical signals to determine the hand gesture. They deliver the most complete, application-independent set of real-time measurements of the hand in HCI. However, they have several drawbacks (1) they are very expensive for casual use, (2) they hinder the naturalness of hand gesture, and (3) they require complex setup and calibration procedures to obtain precise data.

Vision-based hand gesture recognition serves as a promising alternative to them because of its potential in providing unencumbered, non-contact, and more natural interfaces. However, despite lots of previous work [11] [42] [50], traditional vision-based hand gesture recognition methods are still far from satisfactory for real-life applications. Because of the limitations of the optical sensors, the quality of the captured images is sensitive to lighting conditions and cluttered backgrounds, thus it is usually not able to detect and track the hands robustly, which largely affects the performance of hand gesture recognition. I presented a novel dissimilarity distance metric, Finger-Earth Mover's Distance, for hand

gesture recognition approach using Kinect depth sensor in the last chapter. It considers each finger as a cluster and penalizes unmatched fingers. And I built several HCI applications on top of this novel hand gesture recognition system and demonstrate its potential in other real-life HCI applications.

In the last chapter, I illustrate the accuracy, efficiency, and robustness of FEMD hand gesture recognition on a 10-gesture dataset, and I also validate the superiority of FEMD based hand gesture recognition system over traditional corresponding based matching algorithm, Shape Context [6] and skeleton based method, Path Similarity [3]. In this chapter, I further demonstrate my part-based based hand gesture recognition system on top of FEMD[40] in several real-life HCI applications: Arithmetic computation and Rock-paper-scissors game, and Sudoku game.

My current system is built on an Intel Core TM 2 Quad 2.66 GHz CPU with 3GB of RAM and a Kinect depth camera (driver Sensorkinect version 5.0.0). I developed the system on top of OpenNI open platform for Natural Interaction (version 1.0.0.25 binaries) and OpenNI compliant middleware binaries provided by PrimeSense (NITE version 1.3.0.18). Details of OpenNI can be found at http://www.openni.org/.

4.1 Arithmetic computation

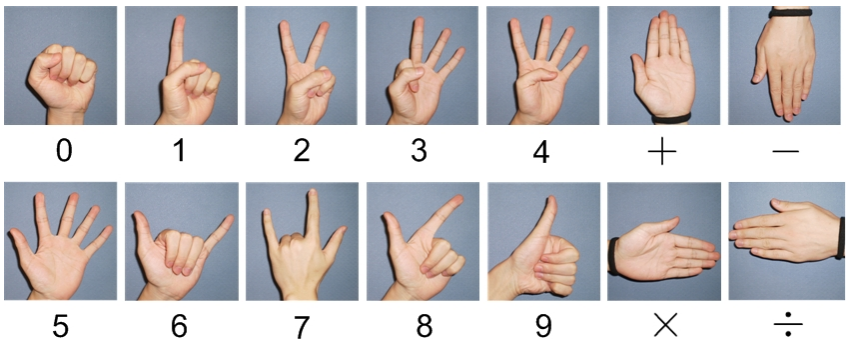


Figure 4.1: The 14 gesture commands in my arithmetic computation system.

Arithmetic computation is an interesting HCI application. Instead of interacting with the computer by the keyboard or mouse, we input arithmetic commands to the computer via hand gestures. As shown in Fig.4.5, 14 hand gestures are employed to represent 14 commands, namely number 0-9 and operator $+$, $-$, $\times$, $\div$, respectively.

By recognizing each input gesture as a command, the computer can perform arithmetic computations instructed by the user. Two examples are shown in Fig.4.2. The key frames are shown as well.

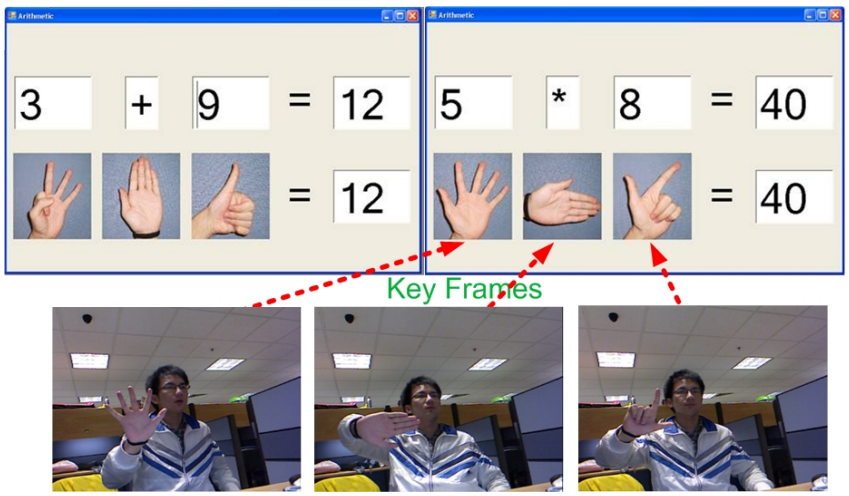


Figure 4.2: Arithmetic computation.

4.2 Rock-paper-scissors game

Rock-paper-scissors is a traditional game. The rule is rock breaks scissors; scissors cut paper; and paper wraps rock. In this demo, I build a Rock-paper-scissors game system played between a human and a computer. Three hand gestures are defined as 3 different weapons in the game, as shown in Fig.4.3.

Figure 4.3: The 3 gesture commands used in Rock-paper-scissors game.

The gesture of the user can be recognized by my system, and the computer just randomly chooses a weapon. Then, according to the game rule, my system can decide the winner between human and computer. Fig.4.4 shows two examples.

4.3 Sudoku game

Sudoku is a popular number-placement puzzle. The objective is to fill a 9×9 grid with digits so that each column, each row, and each of the nine 3×3 sub-grids contain digits from 1 to 9 without repetition. As shown in Fig.4.5, 9 hand gestures are employed to represent 9 commands, namely number from number 1 to number 9.

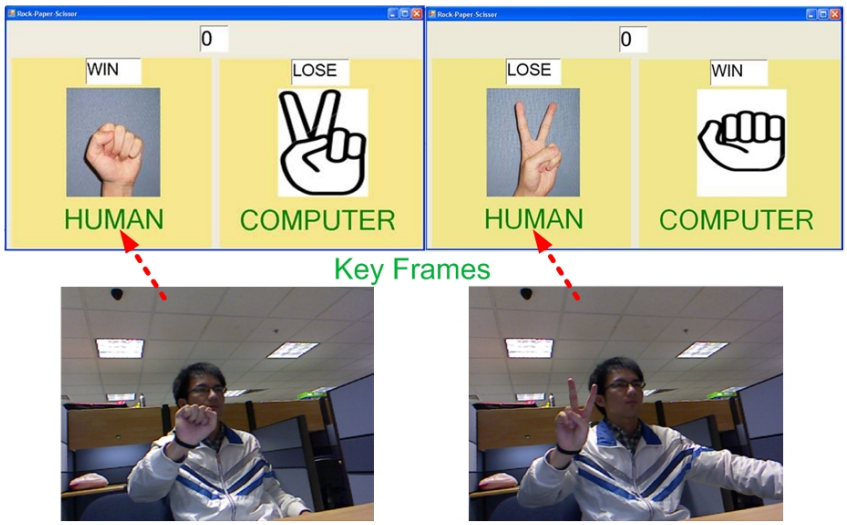


Figure 4.4: Rock-paper-scissors game.

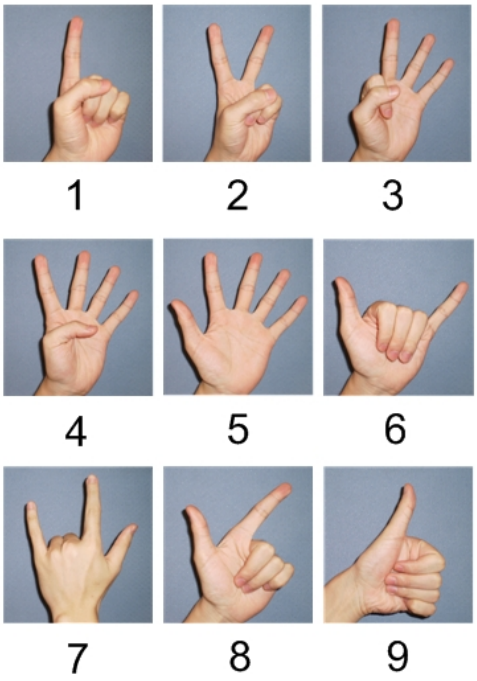


Figure 4.5: The 9 gesture commands adopted in Sudoku game.

The puzzle begins with a partially completed grid and typically has a unique solution. Fig.4.6 shows an example of Sudoku puzzle. The user selects a square by hovering his hand over it and pushes ("clicks") once. He/She then commands a number to be filled into the square by performing the corresponding hand gesture in Fig.4.5. The system recognizes the number and fills it into the square and check whether it is correct in the end.

Figure 4.6: Illustration of Sudoku game

The technical demo showing the human-computer-interaction applications of my part-based hand gesture recognition system [40] [39] is available at http://eeeweba.ntu.edu.sg/computervision/people/home/renzhou/HandGesture.htm.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, I propose a novel near-convex shape decomposition algorithm for robust shape representation, Minimum Near-Convex Decomposition (MNCD), which decomposes 2D and 3D shapes into minimum number of parts with high visual naturalness. I formulate the shape decomposition problem as a combinatorial optimization problem, which have been proved that can decompose shapes into exactly minimum number of near-convex parts. Furthermore, the original problem requiring solving quadratic programming is reformulated into a linear programming problem by introducing auxiliary variables resulting in efficient optimal solution. My decomposition method provides a compact and effective way to represent shapes. I demonstrate the robustness of such a part-based representation in the application of hand gesture recognition. Using Kinect sensor as the input device, I proposed a part-based dissimilarity distance metric, Finger-Earth Mover's Distance (FEMD). As it matches the finger parts and adds penalty to alleviate partial matching, FEMD is fast and can handle noisy hand shapes. Experiments on complex 2D and 3D shape datasets show that my shape decomposition method is robust to local distortions and shape deformation and it outperforms the state-of-the-art methods in terms of the number of decomposed parts. Extensive experiments on a new challenging 10-gesture dataset validate that my part-based hand gesture recognition system is accurate (a 93.9% mean accuracy) and efficient (0.0750s per frame). Furthermore, it is robust in uncontrolled environments and tolerant of shape variations.

5.2 Current Limitations and Future Work

I proposed a shape decomposition method that can produce minimum number of shape parts with high degree of visual naturalness. However, in some cases, the number of decomposed parts is not the most

essential criterion. Thus, it would be an issue to tradeoff between the parts number and their visual naturalness. In the future, I would like to explore shape representation method driven by some specific tasks, for instance, action recognition.

In addition, in my part-based hand gesture recognition system, the current solution relies on the depth cue to separate the hand from cluttered background and it also needs a black belt to locate the hand shape which can be cumbersome. However, in reality, there might be difficult situations with multiple objects in the same depth layer and it may not be sufficient to detect the hand by simple depth thresholding. I would like to explore a more effective hand detection algorithm and better hand location algorithm removing the black belt.

Appendix

I prove Theorem 1 here. In order to prove $\| \mathbf{x}' \|_0 = \| \mathbf{x}'' \|_0$, when $0 \leq \lambda \leq 1/\Sigma_{i=1}^{n} w_i$, first we have:

$$\min_{\mathbf{x}} f(\mathbf{x}) = \| \mathbf{x}' \|_0 + \lambda \mathbf{w}^\top \mathbf{x}' \leq \| \mathbf{x}'' \|_0 + \lambda \mathbf{w}^\top \mathbf{x}'', \tag{5.1}$$

$$\min_{\mathbf{x}} g(\mathbf{x}) = \| \mathbf{x}'' \|_0 \leq \| \mathbf{x}' \|_0, \tag{5.2}$$

As $w_i > 0$, so when $0 \leq \lambda \leq 1/\Sigma_{i=1}^{n} w_i$, $\forall\ \mathbf{x} \in \{0,\ 1\}^n$, $0 \leq \lambda \mathbf{w}^\top \mathbf{x} \leq 1$. Therefore, from Eq.5.1 we further have Eq.5.3, and from Eq.5.2 we further have Eq.5.4:

$$\| \mathbf{x}' \|_0 + \lambda \mathbf{w}^\top \mathbf{x}' \leq \| \mathbf{x}'' \|_0 + 1. \tag{5.3}$$

$$\| \mathbf{x}'' \|_0 \leq \| \mathbf{x}' \|_0 + \lambda \mathbf{w}^\top \mathbf{x}'. \tag{5.4}$$

Combining Eq.5.3 and Eq.5.4, we have:

$$\| \mathbf{x}'' \|_0 \leq \| \mathbf{x}' \|_0 + \lambda \mathbf{w}^\top \mathbf{x}' \leq \| \mathbf{x}'' \|_0 + 1.$$

As $0 \leq \lambda \mathbf{w}^\top \mathbf{x}' \leq 1$, and $\| \mathbf{x}' \|_0$, $\| \mathbf{x}'' \|_0$ are integers, thus $\| \mathbf{x}' \|_0 = \| \mathbf{x}'' \|_0$ when $0 \leq \lambda \leq 1/\Sigma_{i=1}^{n} w_i$.

Author's Publications

(1) **Zhou Ren**, Junsong Yuan, Chunyuan Li, and Wenyu Liu, "Minimum Near-Convex Decomposition for Robust Shape Representation", In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pp. 303-310, Barcelona, Spain, 2011.

(2) **Zhou Ren**, Junsong Yuan, and Zhengyou Zhang, "Robust Hand Gesture Recognition based on Finger-Earth Mover's Distance with a Commodity Depth Camera", In *Proceedings of ACM International Conference on Multimedia (ACM Multimedia)*, pp. 1093-1096, Scottsdale, Arizona, USA, 2011.

(3) **Zhou Ren**, Jingjing Meng, Junsong Yuan, and Zhengyou Zhang, "Robust Hand Gesture Recognition with Kinect Sensor", In *Proceedings of ACM International Conference on Multimedia (ACM Multimedia)*, pp. 759-760, Scottsdale, Arizona, USA, 2011. (Technical Demo)

(4) **Zhou Ren**, Jingjing Meng, and Junsong Yuan, "Depth Camera based Hand Gesture Recognition and its Applications in Human-Computer-Interaction". In *Proceedings of IEEE International Conference on Information, Communication, and Signal Processing (ICICS)*, Singapore, 2011. (Oral)

References

[1] Microsoft Corp. Redmond WA.: Kinect for Xbox 360.

[2] F. Aurenhammer. Weighted skeletons and fixed-share decomposition. In *Computational Geometry*, volume 40, pages 93–101. Elsevier, 2008.

[3] X. Bai and L. J. Latecki. Path similarity skeleton graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1–11, 2008.

[4] X. Bai, L. J. Latecki, and W.-Y. Liu. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:449–462, 2007.

[5] X. Bai, W. Liu, and Z. Tu. Integrating contour and skeleton for shape classification. In *Proc. of IEEE Workshop on NORDIA(in conjunction with ICCV)*, 2009.

[6] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:509–522, 2002.

[7] I. Biederman. Recognition-by-components: A theory of human image understanding. In *Psychological Review*, volume 94, pages 115–147. American Psychological Association, 1987.

[8] M. Bray, E. Koller-Meier, and L. V. Gool. Smart particle filtering for 3D hand tracking. In *Proceedings of IEEE International Conference on Face and Gesture Recognition*, pages 675 – 680, Los Alamitos, CA, USA, 2004.

[9] Q. Cai, D. Gallup, C. Zhang, and Z. Zhang. 3D deformable face tracking with a commodity depth camera. In *Proceedings of European Conference on Computer Vision*, pages 229–242, Crete, Greece, 2010.

[10] T. A. Cass. Robust affine structure matching for 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1265 – 1274, 1998.

[11] C. Chua, H. Guan, and Y. Ho. Model-based 3D hand posture estimation from a single 2D image. In *Image and Vision Computing*, volume 20, pages 191 – 202. Elsevier, 2002.

[12] K. Crauman and T. Darrell. Fast contour matching using approximate earth mover's distance. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 220–227, Washington DC. USA, 2004.

[13] G. Dewaele, F. Devernay, and R. Horaud. Hand motion from 3D point trajectories and a smooth surface model. In *Proceedings of European Conference on Computer Vision*, pages 495 – 507, Prague, Czech Republic, 2004.

[14] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo in Practice*. NewYork: Springer-Verlag, 2001.

[15] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108:52 – 73, 2007.

[16] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. In *Computer Vision and Image Understanding*, volume 108, pages 52–73. Elsevier, 2007.

[17] E. Foxlin. Motion tracking requirements and technologies. *Handbook of Virtual Environment Technology*, pages 163–210, 2002.

[18] D. D. Hoffman and M. Singh. Salience of visual parts. In *Cognition*, volume 14, pages 29–78. Elsevier, 1997.

[19] E. Holden. *Visual recognition of hand motion*. Ph.D thesis, Department of Computer Science, University of Western Australia, 1997.

[20] S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzing clustering and cuts. *ACM Transactions on Graphics*, 2:954–961, 2003.

[21] J. M. Keil and J. Snoeyink. On the time bound for convex decomposition of simple polygons. *International Journal of Computational Geometry and Application*, 12:181–192, 2002.

[22] J. M. Keil and J. Snoeyink. *Minimum Convex Decomposition*. www.cs.ubc.ca/∼snoeyink/demos/convdecomp, 2007.

[23] E. Keogh, L. Wei, X. Xi, S. Lee, and M. Vlachos. Lb_keogh supports exact indexing of shapes under rotation invariance with arbitrary representations and distance measures. In *Proceedings of International Conference on Very Large Databases*, pages 882–893, 2006.

[24] C. Kwok, D. Fox, and M. Meila. Real-time particle filters. In *Proceedings of IEEE*, pages 469 – 484, 2004.

[25] L. J. Latecki, R. Lakamper, and U. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 424–429, 2000.

[26] H. Lee and J. Kim. An hmm-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:961 – 973, 1999.

[27] J.-M. Lien. Hybrid motion planning using minkowski sums. In *Proceedings of Robotics: Science and System V*, 2008.

[28] J.-M. Lien and N. Amato. Approximate convex decomposition of polygons. In *Computational Geometry*, volume 35, pages 100–123. Elsevier, 2006.

[29] J.-M. Lien and N. Amato. Approximate convex decomposition of polyhedra. In *Proceedings of ACM Symposium on Solid and Physical Modeling*, pages 121–131, 2007.

[30] J.-M. Lien and Y. Lu. Planning motion in similar environments. In *Proceedings of Robotics: Science and System V*, 2009.

[31] J. Lin, Y. Wu, and T. Huang. 3D model-based hand tracking ysing stochastic direct search method. In *Proceedings of IEEE International Conference on Face and Gesture Recognition*, pages 693 – 698, Seoul, Korea, 2004.

[32] H. Ling and D. W. Jacobs. Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:286–299, 2007.

[33] H. Liu, L. J. Latecki, and W. Liu. Convex shape decomposition. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 104–124, 2010.

[34] D. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:441 – 450, 1991.

[35] G. Lu and A. Sajjanhar. Region-based shape representation and similarity measure suitable for content-based image retrieval. In *Multimeida Systems*, volume 7, pages 165 – 174. Springer, 1999.

[36] X. Mi and D. Decarlo. Separating parts from 2D shapes using relatability. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[37] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man and Cybernetics-Part C: Application and Review*, 37:311 – 324, 2007.

[38] G. R. S. Murthy and R. S. Jadon. A review of vision based hand gesture recognition. *International Journal of Information Technology and Knowledge Management*, 2:405–410, 2009.

[39] Z. Ren, J. Meng, J. Yuan, and Z. Zhang. Robust hand gesture recognition with kinect sensor. In *Proceedings of ACM International Conference on MultiMedia*, pages 759–760, Scottsdale, Arizona, USA, 2011.

[40] Z. Ren, J. Yuan, and Z. Zhang. Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. In *Proceedings of ACM International Conference on MultiMedia*, pages 1093–1096, Scottsdale, Arizona, USA, 2011.

[41] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. In *International Journal of Computer Vision*, volume 40, pages 99–121. Springer, 2000.

[42] N. Shimada, Y. Shirai, Y. Kuno, and J. Miura. Hand gesture estimation and model refinement using monocular camera-ambiguity limitation by inequality constraints. In *Proceedings of IEEE International Conference on Face and Gesture Recognition*, pages 268 – 273, Nara , Japan, 1998.

[43] J. Shotton, A. Fitzgibbon, M. Cook, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, Colorado, USA, 2011.

[44] K. Siddiqi, S. Bouix, A. R. Tannenbaum, and S. W. Zucker. Hamilton-jacobi skeletons. In *International Journal of Computer Vision*, volume 48, pages 215–231. Springer, 2002.

[45] K. Siddiqi, K. Tresness, and B. B. Kimia. Parts of visual form: Psychophysical aspects. In *Perception*, volume 25, pages 399–424. Pion, 1996.

[46] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, and S. Dickinson. Retrieving articulated 3D models using medial surfaces. In *Machine Vision and Applications*, volume 19, pages 261–274. Springer, 2008.

[47] M. Singh and D. D. Hoffman. Part-based representations of visual shape and implications for visual cognition. In *Advances in Psychology*, volume 130, pages 401–459. Elsevier, 2001.

[48] M. Singh, G. Seyranian, and D. D. Hoffman. Parsing silhouettes: The short-cut rule. *Percept and Psychophys*, 61:636–660, 1999.

[49] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1371 – 1375, 1998.

[50] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Filtering using a tree-based estimator. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 1063 – 1070, Washington, DC, USA, 2003.

[51] M.-C. Su. A fuzzy rule-based approach to spatio-temporal hand gesture recognition. *IEEE Transactions on Systems, Man and Cybernetics-Part C: Application and Review*, 30:276 – 281, 2000.

[52] J. P. Wachs, M. Klsch, H. Stern, and Y. Edan. Vision-based hand-gesture applications. *Communications of the ACM*, 54:60–71, 2011.

[53] A. Wilson and A. Bobick. Parametric hidden markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:884 – 900, 1999.

[54] M. H. Yang, N. Ahuja, and M. Tabb. Extraction of 2D motion trajectories and its application to hand gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:1062 – 1074, 2002.

[55] E. Zuckerberger, A. Tal, and S. Shlafman. Polyhydral surface decomposition with applications. In *Computers & Graphics*, volume 26, pages 733–743. Elsevier, 2002.