Robust hand gesture recognition using rapid transform

He, Jiawei

2015

He, J. (2015). Robust hand gesture recognition using rapid transform. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/65521>

<https://doi.org/10.32657/10356/65521>

NANYANG
TECHNOLOGICAL
UNIVERSITY

Robust Hand Gesture Recognition Using Rapid Transform

HE JIAWEI

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
2015

Acknowledgments

I would like to express my most sincere gratitude to my supervisor, Professor Kai-Kuang Ma, for his instruction, encouragement and support during my master degree study. Professor Kai-Kuang Ma led me into such a fantastic and beautiful research area of computer vision, and he has been giving me kind support and detailed instruction on my research skills, writing skills, and so on. Besides, I would like to thank my dearest friends, Wei Mingwei, Ye Wei, and Yang Mengfei, for their help during my stay in Singapore. Especially, whenever I feel frustrated, they gave me a lot of encouragement and cheered me up. My gratitude also goes to my beloved family for their endless love; my life remains colorful because of you.

Lastly, I want to express my gratitude to the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, for providing me a passionate and happy research environment and equipment to complete my study.

Abstract

In this thesis, a robust and fast hand gesture recognition system is proposed by using the *rapid transform* (RT) to extract the salient features inherited in various hand gestures. The proposed system is ready to be realized in any mobile device (e.g., smart phones, Google glass TM) that has been equipped with a typical RGB camera. The proposed system has an extremely-low computational and algorithmic complexities (hence, incurring low power consumptions). It also delivers highly accurate hand gesture recognition with real-time performance. So far as the hand gesture pattern variations are concerned, the RT exploited in our system is completely invariant to *shift* (or *translation*) and highly invariant to a wide range of rotations, approximately $[-40^\circ, 40^\circ]$ as demonstrated in our simulation results. Although the RT is *scale* sensitive, this issue can be easily overcome by applying a simple hand-size normalization operation on the detected hand region as part of the preprocessing stage. All these attractive merits greatly facilitate the use of hand gesture recognition in various real-life applications. The processing flow of the proposed system is described as follows. Initially, one template for each gesture needs to be created and pre-stored by the user in advance. This processing flow is the same as that in the hand gesture recognition stage. When a hand gesture is posed by the user and to be recognized by the system in the recognition stage, the hand gesture image performed by the user needs to be reasonably captured by a camera, say, from the user's mobile device. The hand region will be detected based on skin color. The segmented hand region will be treated as the foreground object, followed by converting it to a binary image version with further hand-size normalization to address the above-mentioned *scale* issue. Finally, the computed RT is applied to the normalized binary image for extracting salient features of the hand gesture. The RT results are then discriminated against a small set of pre-stored template database based on the *nearest-neighbor* classifier. Extensive simulation

results have clearly shown that our developed RT-based hand gesture recognition system achieves a fairly high recognition rate and is quite robust against practical hand gesture variations, such as translation, scale, orientation, and tilt.

Contents

Acknowledgments	i
Abstract	ii
List of Figures	vi
List of Tables	ix
1 Introduction	1
1.1 Motivation and Objective	1
1.2 System Definition and Requirements	4
1.3 Contribution and Organization	5
2 Literature Review	6
2.1 Single-camera Based HGR	6
2.2 Multiple-camera Based HGR	9
3 Overview of the Rapid Transform (RT)	12
3.1 The Rapid Transform (RT) Algorithm	12
3.1.1 RT Algorithm A	12
3.1.2 RT Algorithm B	13
3.2 Properties of the RT	15
3.2.1 Translation (Shift) Invariant	16
3.2.2 Reflection Invariant	18
3.2.3 Periodicity of the Output Function	20
4 The Proposed RT-based HGR System	23
4.1 Skin Detection and Image Binarization	23
4.2 Hand Detection	26

4.2.1	Forearm Removal	27
4.2.2	Hand Size Normalization	29
4.3	RT-based Feature Extraction and Classification	31
5	Simulation Results	33
5.1	Recognition Performance Under Various Pose Variations	34
5.2	Evaluation of User-independent RT-based Approach	40
5.3	Current Limitations	44
6	Conclusion and Future Work	47
6.1	Conclusion	47
6.2	Future Work	48
	References	50

List of Figures

1.1	(a) An example of data-glove-based hand gesture recognition system [28]; (b) An example of wrist-band hand gesture recognition system [8].	2
1.2	(a) An example of the Kinect-based hand gesture recognition system [29]; (b) An example of a hand gesture recognition system invented by the Leap Motion Controller [30].	3
1.3	Block diagram of a typical visual recognition system.	4
2.1	Vision based hand gesture recognition can be classified into two categories, single-camera based and multiple camera-based hand gesture recognition.	7
2.2	An example of the hierarchical hand model method [14] to identify fingers and palm region respectively. As shown, each part of the hand region is segmented and identified to facilitate further recognition.	8
2.3	An example of hand gesture recognition based on the orientation histogram [23]. As shown, low-level feature-based methods use the whole hand region as the input, and extract features from the whole region.	9
2.4	An example of using the kinect sensor to detect and segment hand region from complicated background [18], as shown in this figure.	11
3.1	The basic computation kernel of the RT. The solid and dashed lines denote addition and absolute-valued difference operations, respectively.	13
3.2	Algorithm A flow chart of eight-point RT	14
3.3	Algorithm B for flow chart of eight-point RT	15
3.4	A demonstration of the translation invariant property of the RT. That is, the RT of each 8×8 matrix in sub-figures (a), (b) and (c) will yield the same transform result as shown in (d)	19

3.5	A demonstration of the reflection invariant property of the RT. That is, the RT of each 8×8 matrix in sub-figures (a), (b) and (c) will yield the same transform result as shown in (d)	21
3.6	A demonstration of the periodicity property of the RT. (a) and (c) are two input patterns, (b) and (d) are their corresponding output patterns respectively.	22
4.1	The block diagram of the proposed RT-based hand gesture recognition system	24
4.2	An illustration of the using the depth map to segment the hand gesture region from the background [30]. The depth map is acquired by Kinect sensor. The sub-figure on the top right corner is the acquired image, the sub-figure on the top left is the depth map, and the sub-figure on the second row is the segmented hand region.	26
4.3	The three methods tried in this work are given here. Column (e) represents the results of the three methods. The first one with only palm region, the second one with palm region and little part of the forearm region, and the third one without palm region.	28
4.4	The Figure in Ren's work [18] to demonstrate the forearm removal phase. As can be seen, the wrist position is indicated by the black belt and the forearm can thus be easily removed.	29
4.5	Some demonstrations of skin detection, binarization and forearm removal process. Column (a) represents the initial image acquired by the camera, and column (b) represents the detected skin region. By applying the Otsu binarization algorithm [44], Column (c) represents the binarization result (including fingers, palm and forearm region). Column (d) shows the largest circle identified over the palm region, and column (e) represents the hand region with forearm removed.	30
5.1	The twelve buttons on every mobile phone (numbers 0-9, "*", and "#"). The 12 hand gesture patterns in the test set are exploited to replace these buttons in this work.	34

5.2	Examples of the twelve gesture types used in the experiments, in which all gestures are from the same user	35
5.3	An illustration of tilted gesture patterns and gestures with orientation variation (better viewed in color).	36
5.4	Examples of the generated gesture patterns with different orientation using Matlab.	37
5.5	Recognition accuracy of gestures with different orientations. Approximately within the range of $[-40^\circ, 40^\circ]$, the hand gesture patterns can be recognized correctly and efficiently.	38
5.6	Examples of severe tilted gestures the proposed method failed to recognize.	39
5.7	Details of the user-dependent experiment results with a set of templates for each user. The results of the five users are marked with different colors, respectively.	41
5.8	The 12 gesture types used in [31]. As shown, 11 out of 12 gesture patterns are the same with our test gesture patterns.	42
5.9	The confusion matrix of 12 gesture types. Each row represents the recognition accuracy of one type of gesture patterns.	43
5.10	The 4 gesture types selected in Jochen Triesch Static Hand Posture Dataset. These gesture types corresponds with gesture 10, 1, 2, 8 in our test set respectively.	44
5.11	A comparison of the 5 features on Jochen's dataset. The proposed RT-based method outperforms other methods.	46
6.1	Examples of the hand gesture images acquired under complicated background.	48
6.2	Some examples of the hand gesture patterns that the RT-based method cannot recognize correctly. Note that the palm region in these hand gesture patterns cannot be represented by the largest circle inscribed in the hand region.	49
6.3	An example of some sign language gestures [4].	49

List of Tables

5.1	Recognition accuracy of different users	40
5.2	Performance comparison of each feature representation	42
5.3	Performance comparison on Jochen's dataset	45
5.4	Performance with different sets of templates	45

Chapter 1

Introduction

1.1 Motivation and Objective

Hand gesture is one of popular and effective ways for human beings to communicate or interact with a machine through a remote distance without incurring any physical contact. Therefore, *hand gesture recognition* (HGR) by machine has been investigated in computer vision research to facilitate human-computer interaction, human-robot interaction, sign language recognition, to name a few [1, 2, 3]. However, the complicated patterns of hand gestures that can be performed by the user and their possible variations, such as scale, translation, orientation, lighting condition, occlusion, and so on, make automatic HGR a very challenging task.

In general, hand gestures can be classified into *static gestures* and *dynamic gestures* [4]. Static gestures do not change with time; that is, a gesture can be represented by one image. Automatic recognition of static gestures by machine mainly relies on the shape of the hand gesture under recognition. On the other hand, a dynamic gesture involves a sequence of temporally-acquired images that are varied along with time. A typical example of dynamic HGR is sign language recognition. Obviously, automatic recognition of such dynamic gesture is much more difficult, compared with that static HGR since complicated information (e.g., trajectory of hand movement) are required to be generated, processed and interpreted along the way. In this thesis, our focus is on the

Figure 1.1: (a) An example of data-glove-based hand gesture recognition system [28]; (b) An example of wrist-band hand gesture recognition system [8].

recognition of static hand gesture using the RT to extract salient features inherited in each hand gesture.

Automatic recognition of static hand gesture has long been one of the major research areas in computer vision and pattern recognition. Earlier hand gesture recognition works exploit data glove or wrist band that contains electric sensors to acquire the hand gesture data by tracking the position of finger joints, palm or wrist in real time [32]. An example of a glove-based hand gesture recognition system is shown in Fig. 1.1 (a). Although these methods can acquire accurate information about hand gestures, such hand gesture recognition system is expensive to set up for practical applications. Furthermore, it is inconvenient for users to wear such device every time when they intend to use it.

Recently, along with the quick development of virtual reality (VR), glove or wrist band-based hand gesture recognition methods have become popular for those applications that require to track and recognize gesture movement in real time. For example, the action of "pulling the trigger" in a VR shooting game (as demonstrated in Fig. 1.1 (b)) can be achieved by wearing a wrist band on the user's arm [8]. In this way, the precise movement of the arm's muscles can be acquired and then used to recognize the hand gesture.

Figure 1.2: (a) An example of the Kinect-based hand gesture recognition system [29]; (b) An example of a hand gesture recognition system invented by the Leap Motion Controller [30].

Compared with data glove or wrist band, *vision-based* hand gesture recognition techniques do not require such equipments other than deploying a low-cost RGB camera, which is commonly available as a built-in component in a large number of electronic devices. Therefore, vision-based techniques have been dominating the mainstream of hand gesture recognition research, which can be classified into two categories: *single-camera-based* HGR and *multiple-camera-based* HGR.

A typical single-camera based hand gesture recognition method utilizes one camera to acquire the input gesture image. Based on the features used to represent the hand gesture patterns, these methods can be further classified into two groups, *high-level feature-based methods* and *low-level feature-based methods*.

With the inception of depth camera, the RGB-D camera has recently been deployed for conducting the HGR task [18, 19, 20]. The most famous RGB-D camera is the Kinect sensor (see Fig. 1.2 (a)). Besides, it is worth to mention another RGB-D camera which is introduced by Leap Motion Controller [30]. Unlike the Kinect, the Leap Motion Controller is much smaller and can be easily connected to a computer with a simple USB interface as shown in Fig. 1.2 (b).

Figure 1.3: Block diagram of a typical visual recognition system.

1.2 System Definition and Requirements

Hand gesture recognition is a typical task encountered in visual pattern recognition, which has a block diagram as shown in Fig. 1.3. It consists of image acquisition, image feature extraction, feature selection, and pattern classification. Since various hand gesture patterns involved in real-life applications could be quite complicated, how to extract salient features from the hand gesture becomes a very challenging task and instrumental to yield highly accurate recognition performance. Furthermore, there are several practical considerations for hand gesture recognition that are needed as system requirements:

1) In real-life applications, it is expected that the input hand gestures could be easily involving various types of variations caused by the pose of hand gesture, such as translation, scale, orientation, and tilt.

2) Besides the variations caused by the user, variations could be also caused by the external interferences, such as complicated background, lighting conditions, partial occlusion, etc. These variations increase difficulty on the segmentation of hand region from the acquired image.

3) It is crucial that hand gesture recognition algorithm can be executed in real time for maintaining the fast response of machine to human intentions. Real-time performance requires low algorithm computational complexity, especially for mobile devices which have inferior processors and low power supply.

4) Nowadays, the use of a mobile device tends to be quite personal. The HGR performance can leverage this aspect to become user-dependent. User-dependent HGR allows users to create their own hand gesture templates. Therefore, the proposed HGR system can be used to recognize

1.3 Contribution and Organization

Motivated by these observations, a novel real-time hand gesture recognition system is proposed in this thesis. It is ready to be incorporated in any mobile device, such as smart phones, Google glass TM. As a typical mobile device is normally equipped with only one RGB camera, the proposed system is targeted to be a single camera approach. The proposed hand gesture recognition system exploits the rapid transform (RT) to extract the salient features of the imposed hand gesture. The RT-based method is able to cope with practical hand gesture variations, such as translation, scale, orientation, and tilt. Furthermore, its computational and algorithmic complexities are fairly low, which is quite attractive for delivering real-time performance. The mobile device equipped with the proposed hand gesture recognition system can be used to assist blind people on dialing phone numbers or performing other tasks through the instruction issued by hand gesture. Last but not least, a thorough study of the RT is provided in this thesis, including the algorithms of RT and some useful properties.

The rest of the thesis is organized as follows. Chapter 2 provides a literature survey of existing state-of-the-art hand gesture recognition methods. Chapter 3 gives an overview of the RT and discusses its properties. The proposed RT-based robust hand gesture recognition system is then described in detail in Chapter 4, followed by conducting extensive simulation results as documented in Chapter 5. The final chapter concludes the thesis with a recommendation of future works.

Chapter 2

Literature Review

In this chapter, an overview of existing vision-based hand gesture recognition approaches is provided, more complete reviews are given by [4, 5, 6, 7]. As discussed before, vision-based hand gesture recognition techniques do not require equipments other than one or several cameras, which is commonly available in a large number of electronic devices. Therefore, vision-based techniques have the potential to be used in more situations than data-glove based methods. Based on the equipment used, vision-based hand gesture recognition techniques can be classified into two categories, *single-camera based hand gesture recognition* and *multiple-camera based hand gesture recognition*. The diagram of this classification is given in Fig. 2.1.

2.1 Single-camera Based HGR

A typical single-camera based hand gesture recognition method utilizes one camera to acquire the input gesture image. In other words, one hand gesture is represented by one image only. As shown in Fig. 1.3, one input hand gesture image is acquired, preprocessed and represented by different features. After that, these features are used for recognition. Based on the features used to represent the hand gestures, single-camera based hand gesture recognition methods can be further classified into two groups, *high-level feature-based methods* and *low-level feature based methods*.

Figure 2.1: Vision based hand gesture recognition can be classified into two categories, single-camera based and multiple camera-based hand gesture recognition.

High-level feature-based methods High-level feature-based methods, also known as deformable part-based methods, usually first identify certain parts of the hand gesture, such as the joint position, fingertips, or some anchor points on the palm , then utilize these information to represent different hand gesture patterns. [12, 13] used peaks to identify fingers from the gesture region, then simply count finger numbers and relative locations and [14] proposed a hierarchical hand model to identify fingers and palm region respectively. This method is shown in Fig. 2.2.

High-level feature-based methods can be fairly robust in user-independent scenarios, because specific parts of the hand gesture are identified before the recognition process. However, this additional identification stage usually results in high computational cost, which could be a challenge in some real-time hand gesture recognition systems.

Low-level feature-based methods In fact, in many hand gesture recognition applications, the hand gesture patterns are not so complicated which requires deformable

Figure 2.2: An example of the hierarchical hand model method [14] to identify fingers and palm region respectively. As shown, each part of the hand region is segmented and identified to facilitate further recognition.

methods. Thus, it is not necessary to reconstruct a full 3D hand model or to extract complicated high-level features in these cases. Some approaches use simple low-level features that can be extracted efficiently and are fairly robust to distortions to represent hand gestures. One prominent advantage of these methods is that the computational complexity is usually low, because the identification of certain parts in high-level feature-based methods is not necessary. Edges and contours are common low-level features frequently used in hand gesture recognition techniques [17]. localized contour sequence (LCS) based feature in [26] and modified census transform (MCT) in [24] were utilized to represent the hand gesture contours. Besides contours, other methods regarded the whole hand region as the low-level feature. These methods include moment functions [25], orientation histogram [23], etc. An example of the orientation histogram based hand gesture recognition method is shown in Fig. 2.3. In these methods, certain features are extracted from the whole hand region, therefore, the computation complexities of these methods are theoretically the lowest among all hand gesture recognition methods.

Low-level feature-based methods usually can achieve real-time performance because of the low computational complexity. However, these methods are not effective under complicated backgrounds. In other words, hand region is difficult to be segmented from

Figure 2.3: An example of hand gesture recognition based on the orientation histogram [23]. As shown, low-level feature-based methods use the whole hand region as the input, and extract features from the whole region.

the background. Therefore, other more advanced segmentation methods need to be introduced to facilitate the low-level feature based methods.

2.2 Multiple-camera Based HGR

On the other hand, some researchers utilize more than one camera to acquire the input hand gesture data. Stereo camera, for instance, is used to capture the hand gesture pattern from multiple viewpoint. The acquired data is then utilized to reconstruct a full 3-D hand model [9, 10]. The 3-D hand model usually contain more information than the 2-D methods. They can be invariant under viewpoint variation or occlusion. However, the reconstruction of a full 3-D hand model takes a lot of effort, and the high computational complexity hinders its application.

Another try is to combine a common RGB camera with a depth camera. The most famous RGB-D camera is Kinect. In Ren's work [18], the Kinect sensor is used to capture the depth map, which contains the depth information of different objects in the image. Depth map can be used to segment the hand region easily, as shown in Fig. 2.4. The segmented hand gesture is then represented by decompose it into several natural parts and counts the distance between the palm region and each fingertip. A review of RGB-D camera-based hand gesture recognition is given in [19]. Another RGB-D camera worth-mentioning is the Leap Motion Controller [30]. Unlike Kinect, Leap Motion Controller is quite small, and can connect to any computer with a simple USB interface as shown in Fig. 1.2.

Utilizing the information acquired by the additional cameras, multiple-camera based methods usually have potential to achieve high recognition accuracy. However, the disadvantage of these methods is also obvious: the system is expensive and hard to set up. Therefore, the using situation is limited, for example, these methods can not be used on mobile devices or a common laptop, which only has one common RGB camera.

In this thesis, a robust hand gesture recognition system is developed to be used on any mobile device. As common mobile device only has one RGB camera, the proposed system belongs to the category of single-camera based hand gesture recognition. The proposed hand gesture recognition system is based on the *rapid transform* (RT) [34], which is used to extract salient features inherited in various hand gestures. The RT has been used in different recognition tasks [34, 36, 37]. Reitboeck and Brody [34] first suggested a method to recognize handwritten letters, Wang et al. [35] used RT to recognize printed Chinese characters, Ma et al. [36] utilized RT to recognize tactile, Muller et al. [38] suggested a system for speaker-independent speech recognition system. And lately, Li et al. [39] introduced RT into texture classification area. However it is still a much less known transform method. In fact, the RT is a very powerful feature extraction method with

Figure 2.4: An example of using the kinect sensor to detect and segment hand region from complicated background [18], as shown in this figure.

great potentials for any pattern-recognition oriented application. Foremost, it is *shift* (or *translation*) invariant [34], which is highly desirable on facilitating pattern recognition tasks, including the hand gesture recognition investigated in this thesis. Furthermore, the computational complexities are probably the lowest among all the known transformation techniques; this is quite attractive for being used in real-time systems since the power consumption and VLSI implementation are of great concerns.

Chapter 3

Overview of the Rapid Transform (RT)

In this chapter, the RT algorithm is firstly introduced, as well as some examples of the RT process, then some of its properties and how these properties can be an advantage in hand gesture recognition are given.

3.1 The Rapid Transform (RT) Algorithm

3.1.1 RT Algorithm A

For an input sequence with 2^M numbers, it takes M iterations to complete the RT process. In each iteration, a new set of 2^M numbers will be yielded by conducting the following steps. First, the numbers in any iteration (k) will be equally divided into 2^{k-1} non-overlapping sub-groups with the same size. That is, the numbers in any sub-group of iteration (k) are the results generated from the numbers in the preceding iteration ($k-1$) by computing

$$X_{m,2n}^{(k)} = X_{m,n}^{(k-1)} + X_{m+s,n}^{(k-1)} \quad (\text{Eq. 3.1})$$

and

$$X_{m,2n+1}^{(k)} = \left| X_{m,n}^{(k-1)} - X_{m+s,n}^{(k-1)} \right| \quad (\text{Eq. 3.2})$$

Figure 3.1: The basic computation kernel of the RT. The solid and dashed lines denote addition and absolute-valued difference operations, respectively.

where $s = 2^{M-k}$, $t = 2^{k-1}$, $m \in [0, s - 1]$, and $n \in [0, t - 1]$. In this notation, the first subscript represents the sample index of the sub-group as indicated in the second subscript.

The basic computation kernel of the RT with only two inputs, is illustrated in Fig. 3.1. The solid line denotes the operation of addition as expressed in (1) and the dashed line denotes the absolute-valued difference as expressed in (2). An RT example with eight-variable input sequence is illustrated in Fig. 3.2. Note that the basic computation kernel as shown in Fig. 3.1 is performed in all transform iterations, which makes it very simple to implement in software and in VLSI circuit design with low cost.

3.1.2 RT Algorithm B

A second algorithm, giving the same results is shown in Fig. 3.3. The numbers of the iterations required for an input sequence with 2^M numbers is again M . In each iteration generated by the last one, the numbers are calculated by

$$X_{2n}^{(k)} = X_{2n}^{(k-1)} + X_{2n+s}^{(k-1)} \quad (\text{Eq. 3.3})$$

and

$$X_{2n+1}^{(k)} = \left| X_{2n+1}^{(k-1)} - X_{2n+1+s}^{(k-1)} \right| \quad (\text{Eq. 3.4})$$

Figure 3.2: Algorithm A flow chart of eight-point RT

where $s = 2^{M-1}$, , and $n \in [0, 2^{M-1} - 1]$.

The output pattern of algorithm B is exactly the same as algorithm A. Detailed proof is given by [34]. The advantage of this algorithm is that the exactly same operations are performed in every transform iteration.

Furthermore, the one-dimensional (1-D) RT can be directly extended and applied to conduct the two-dimensional (2-D) RT through a separable approach. That is, given a grey-scale image, the 1-D RT will be performed to each row independently, and the obtained RT result will replace the input data at the same row index. A collection of all row-based RT results is then subjected to perform 1-D RT, column by column,

Figure 3.3: Algorithm B for flow chart of eight-point RT

respectively. Note that the order of the two stages can be reversed (i.e., performing all columns one by one first, followed by the same 1-D RT operation for each row, individually), and the RT result should remain the same.

3.2 Properties of the RT

The RT has several unique and attractive properties, such as translation invariant, reflection invariant, which are highly desirable to pattern recognition applications, including the hand gesture recognition focused in this report.

3.2.1 Translation (Shift) Invariant

For 1-D case, the RT transform is invariant under cyclic shift of the input sequence. With an input sequence of N numbers,

$$\begin{aligned} & RT[X_{(i \bmod N)}, X_{(i+1 \bmod N)}, \dots, X_{(i+N-1 \bmod N)}] \\ &= RT[X_{(j \bmod N)}, X_{(j+1 \bmod N)}, \dots, X_{(j+N-1 \bmod N)}], \end{aligned} \quad (\text{Eq. 3.5})$$

where i and j are any positive integers and $\bmod$ performs modulo operation.

This property makes RT invariant of the pattern translation as demonstrated in Fig. 3.4, where the three triangle-shaped input patterns at different locations will yield the same RT result as shown in the sub-figure (d) despite the fact that their spatial positions are different.

This shift invariant property is mathematically proved in [34]. It will be proved again as follows. The flow chart of algorithm 1 consists of an input sequence of 2^M variables, followed by M iterations of operation, performed on pairs of the variables of the preceding iteration. For a given iteration $(k - 1)$, the calculation of the 2^M numbers in the next iteration (k) are defined by Eq. 3.1 and Eq. 3.2. From the flow chart and equations, it is evident that the operations in the flow chart follow symmetrical patterns, so that every iteration $(k > 0)$ can be divided into 2^k groups, with each group containing 2^{M-k} numbers.

The input numbers are defined as being cyclic with a period of $N = 2^M$, so that

$$X_{m,0}^{(0)} = X_{m+2^M,0}^{(0)}. \quad (\text{Eq. 3.6})$$

We now consider a 2^{M-1} -fold cyclic shift of the numbers in the input sequence, defined as

$$Y_{i,0}^{(0)} = X_{i+2^{M-1},0}^{(0)}, \quad (\text{Eq. 3.7})$$

which corresponds to a shift of the input pattern by 2^{M-1} places, that is, to an interchange of the top and bottom halves of the variables of the input sequence. From Eq. 3.1 and Eq. 3.2, the numbers in iteration (1) can be obtained.

$$Y_{i,0}^{(1)} = Y_{i,0}^{(0)} + Y_{i+2^{M-1},0}^{(0)} = X_{i+2^{M-1},0}^{(0)} + X_{i+2^M,0}^{(0)} = X_{i,0}^{(1)} \quad (\text{Eq. 3.8})$$

and

$$Y_{i,1}^{(1)} = \left| Y_{i,0}^{(0)} - Y_{i+2^{M-1},0}^{(0)} \right| = \left| X_{i+2^{M-1},0}^{(0)} - X_{i+2^M,0}^{(0)} \right| = X_{i,1}^{(1)} \quad (\text{Eq. 3.9})$$

i.e. the numbers in iteration (1) remains unchanged.

By a similar argument, it can be seen that the numbers in iteration (2) are invariant to a 2^{M-2} -fold cyclic shift of the numbers in the input sequence, and further, in general, we state the following theorem.

Theorem: *The numbers in any group (n) of a given iteration (k) are cyclic with a period of 2^{M-k} , and therefore invariant to a 2^{M-k} -fold cyclic shift of the numbers in the input sequence, i.e.*

$$Y_{m,n}^{(k)} = X_{m+2^{M-k},n}^{(k)} = X_{m,n}^{(k)}, \quad (\text{Eq. 3.10})$$

for $Y_{m,0}^{(0)} = X_{m+2^{M-k},0}^{(0)}$.

Proof. The theorem is proved by mathematical induction. First, assume the truth of the proposition for group (n) in iteration (k-1), i.e.

$$Y_{m,n}^{(k-1)} = X_{m+2^{M-(k-1)},n}^{(k-1)} = X_{m,n}^{(k-1)} \quad (\text{Eq. 3.11})$$

for $Y_{m,0}^{(0)} = X_{m+2^{M-(k-1)},0}^{(0)}$, and then calculate the numbers in column (k) for a s^{M-k} -fold cyclic permutation of the numbers in the input sequence,

$$Y_{m,0}^{(0)} = X_{m+2^{M-k},0}^{(0)} \quad (\text{Eq. 3.12})$$

From Eq. (1) and (2), the numbers in group $(2n)$ of iteration (k) can be obtained:

$$Y_{m,2n}^{(k)} = Y_{m,n}^{(k-1)} + Y_{m+s,n}^{(k-1)} = X_{m+2^{M-k},n}^{(k-1)} + X_{m+2^{M-k+1},n}^{(k-1)} = X_{m+2^{M-k},n}^{(k-1)} + X_{m,n}^{(k-1)} = X_{m,2n}^{(k)} \quad (\text{Eq. 3.13})$$

A similar argument could show the truth of the proposition for the variables in group $(2n+1)$ of iteration (k) :

$$Y_{m,2n+1}^{(k)} = X_{m,2n+1}^{(k)} \quad (\text{Eq. 3.14})$$

From the assumption that the theorem is true for $(k-1)$, it is also true for (k) . And since it is true for $k=1$ as given before, it is true for all $k \in [1, M]$. Therefore, for the output pattern, that is the iteration (k) , it is shown that that numbers $X_{0,n}^M$ remain unchanged if the input is shifted by $s = 2^{M-M} = 1$ place, i.e.,

$$Y_{0,n}^{(M)} = X_{0,n}^{(M)} \quad (\text{Eq. 3.15})$$

for $Y_{m,0}^{(0)} = X_{m+1,0}^{(0)}$, and therefore, by any integer of places. The invariance under cyclic shift of the RT is proven.

3.2.2 Reflection Invariant

For 1-D case, the RT transform is invariant under reflection of the input sequence. For an input sequence with 2^M numbers $(X_0, X_1, X_2, \dots, X_{2^M-1})$,

$$RT(X_0, X_1, X_2, \dots, X_{2^M-1}) = RT(X_{2^M-1}, X_{2^M-2}, X_{2^M-3}, \dots, X_1, X_0) \quad (\text{Eq. 3.16})$$

For example, the input sequence is given as an 8 by 1 column vector,

$$X^{(0)} = [0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1]^T,$$

Figure 3.4: A demonstration of the translation invariant property of the RT. That is, the RT of each 8×8 matrix in sub-figures (a), (b) and (c) will yield the same transform result as shown in (d)

it needs 3 iterations to perform the RT, and the final result is

$$RT\{X\} = X^{(3)} = [4 \quad 2 \quad 2 \quad 0 \quad 2 \quad 0 \quad 2 \quad 0]^T.$$

The reflection of the input sequence is

$$Y^{(0)} = [1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0]^T,$$

and the final result $RT\{Y\}$ is the same as $RT\{X\}$.

For 2-D case, the reflection of vector in the 1-D case is equivalent to the mirror image of array in the 2-D case. That means any pair of mirror images will result in the same RT result. An example is shown in Fig. 3.5, where the input patterns (b) and (c) are the mirror image of pattern (a) and they will yield the same RT result as shown in sub-figure (d).

A formal proof of this property is given as follows. First, the reflection of the input pattern is defined as

$$Y_{m,0}^{(0)} = X_{2^M-1-m,0}^{(0)} \quad (\text{Eq. 3.17})$$

Using Eq. 3.1 , we can get the numbers of first group in iteration (1)

$$Y_{m,0}^{(1)} = Y_{m,0}^{(0)} + Y_{m+2^{M-1},0}^{(0)} \quad (\text{Eq. 3.18})$$

and with Eq. 3.17,we can further obtain

$$Y_{m,0}^{(1)} = X_{2^M-1-m,0}^{(0)} + X_{2^{M-1}-1-m,0}^{(0)} \quad (\text{Eq. 3.19})$$

hence, $Y_{m,0}^{(1)}$ is the reflection of $X_{m,0}^{(1)}$. With a similar argument, the second group of iteration (1) $Y_{m,1}^{(1)}$ is the reflection of $X_{m,1}^{(1)}$.

With a mathematical induction proof similar to the one used in the proof of shift invariant property, it can be shown that in any column (k), each group is the reflection of the group for the original input. Thus in iteration ($M-1$) consisting only two numbers, the reflection becomes equivalent to a unit cyclic shift. This, by previous proof, leaves the output invariant.

3.2.3 Periodicity of the Output Function

The transform of a one-dimensional sequence of N input numbers is periodic over its basic interval if $N/2$ or more adjacent numbers are zero. An example of this property is shown in Fig. 3.6.

Figure 3.5: A demonstration of the reflection invariant property of the RT. That is, the RT of each 8×8 matrix in sub-figures (a), (b) and (c) will yield the same transform result as shown in (d)

Figure 3.6: A demonstration of the periodicity property of the RT. (a) and (c) are two input patterns, (b) and (d) are their corresponding output patterns respectively.

Chapter 4

The Proposed RT-based HGR System

The proposed RT-based static hand gesture recognition algorithm (as shown in Fig. 4.1) consists of three stages: 1) *skin detection and image binarization*, 2) *hand detection*, and 3) *RT-based feature extraction and hand gesture recognition*. The former two stages together is a ‘*preprocessing*’ with an aim to prepare the input data of the hand gesture under recognition. The last stage is the heart of the recognition system. A description of each stage is detailed as follows.

4.1 Skin Detection and Image Binarization

Before applying the RT to the hand gesture image data, the region of the hand must be first detected and segmented from the initially acquired image to facilitate the recognition task, since other image contents are irrelevant and will affect recognition result. Many techniques have been proposed to detect and segment the hand from a complicated background [40, 41, 42]; some are based on skin color information [40], and some utilize the shape information [41] or a combination of several clues, such as shape, context and skin color information [42]. With a reasonable assumption that the hand gesture should be acquired with a not-so-complicated background, the color of skin can be used to

Figure 4.1: The block diagram of the proposed RT-based hand gesture recognition system

detect the hand region effectively. For that, a simple and yet relatively reliable method for conducting skin detection as described in [43] is adopted in this work, where skin pixels are detected by combining the features obtained from the YCbCr color space and the YIQ color space. The hue value is estimated from the chromatic components Cb and Cr by $\theta = \arctan(Cb/Cr)$, and the in-phase color component I is calculated from the RGB component as $I = 0.596R - 0.274G - 0.322B$. According to [43], the range of θ and that of the in-phase color component I for the ‘skin pixel’ are: $105 \leq \theta \leq 150$ and $30 \leq I \leq 100$. These ranges are suggested regardless of different races of people. Any pixel other than the skin pixel is regarded as the background pixel. Otsu’s method [44] is then used to binarize the image to separate and denote these two classes as the foreground (representing the hand region in white) and the background (in black), respectively. Some demonstrations of skin detection and the resulted binarized images are shown in Fig. 4.5

column (a) to column (c). Column (a) represents the initial image acquired by the camera, and column (b) represents the detected skin region. By applying the Otsu binarization algorithm [44], column (c) represents the binarization result.

Note that if other more complicated hand detection methods like [42] are introduced to extract hand region from the acquired image, the RT-based algorithm can easily be extended to situations with complicated background. In these situations, the background is unconstrained with large regions in the background also be in skin color.

To enable more robust hand gesture recognition in these situations, another effective way is to use other sensors to capture the hand gesture regions, for example, data glove [32]. 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 while performing the hand gestures, it is very inconvenient. 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. Recently, thanks to the recent development of inexpensive Kinect sensor, many techniques utilize the Kinect sensor to capture the depth map, which contains the depth information of different objects in the image. Depth map can be used to segment the hand region easily. An illustration of using the depth map to segment the hand gesture region from the background is given in Fig. 4.2 [30]. The depth map is acquired by Kinect sensor. The sub-figure on the top right corner is the initial acquired image, the sub-figure on the top left is the depth map, and the sub-figure on the second row is the segmented hand region.

As the goal of this thesis is to investigate the efficiency and efficacy of the RT in hand gesture recognition, segmentation improvement is put in the future work.

Figure 4.2: An illustration of the using the depth map to segment the hand gesture region from the background [30]. The depth map is acquired by Kinect sensor. The sub-figure on the top right corner is the acquired image, the sub-figure on the top left is the depth map, and the sub-figure on the second row is the segmented hand region.

4.2 Hand Detection

This stage consists of two major steps: *forearm removal* and *palm size normalization*. Forearm has to be removed from the hand gesture image because it has no relevance to the hand gesture at all and can affect recognition result. After that, palm size need to be normalized because the RT is scale sensitive. So, generally speaking, the purpose of this stage is to prepare the segmented skin region from any distraction to perform the RT. These steps are detailed as follows.

4.2.1 Forearm Removal

According to the skin pixel detection as described previously, the pixels located in the arm region (provided no sleeve covers on it) will be detected and regarded as a part of the hand. Obviously, the arm region has no relevance to the hand gesture at all and it should be removed. To accomplish this goal, the palm region is detected first, since the generic shape of a palm is close to a circle and thus it can be more easily detected by simply finding the largest circle inscribed inside the palm region. This process can be done by utilizing a morphological operation, *open*, iteratively. To be more specific, the initial opening element is a circle with the radius of one pixel. The radius of the circle (opening element) is then extended by one pixel at a time in each iteration. This process will be terminated once every pixel of the morphological opening result becomes background, which indicates that the extended opening element is one pixel larger than the largest circle inscribed inside the palm region. Once this case is encountered, the palm region is identified, which is based on the result obtained in the previous opening iteration. With the identification of the palm region, all the pixels below the lower boundary of the palm region are considered as forearm region and removed. For illustration, refer to Fig. 4.5 (column (c) - column (e)), where column (c) represents the binarized skin region (including the fingers, palm, and forearm region) obtained from the previous stage. Column (d) shows the largest circle identified over the palm region, and column (e) represents the hand region with forearm removed.

In this work, three forms of the forearm removal methods are tested, shown as Fig. 4.3. Method 1 removes every pixel below the palm region, as in the first row of Fig. 4.3. Method 2 uses the lower boundary of the palm region to identify the wrist position, and every pixel below the wrist position are removed, resulting in the second row of Fig. 4.3. Method 3 removes the palm region as well as the forearm region, because palm region is the same in different hand gesture patterns.

Figure 4.3: The three methods tried in this work are given here. Column (e) represents the results of the three methods. The first one with only palm region, the second one with palm region and little part of the forearm region, and the third one without palm region.

The three methods are tested on the test set given in Chapter 5, and as a result, the recognition rate remain unchanged. The reason is that no matter the palm region is removed or not, the unique features of each gesture class, which is the finger number, location and position of each gesture, remains unchanged. As method 2 is the most intuitive one, it is selected for forearm removal in this work.

In Ren's work [18] shown in Fig. 4.4, black belt has to be worn by the user in order

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.

Figure 4.4: The Figure in Ren’s work [18] to demonstrate the forearm removal phase. As can be seen, the wrist position is indicated by the black belt and the forearm can thus be easily removed.

to identify the wrist of the hand. Pixels below the wrist is removed. This method is straightforward, and can get good forearm removal performance. However, similar to the discussion about data gloves before, it is inconvenient for users to wear the wrist band anytime he uses the hand gesture recognition system. Other works require user to wear long sleeves or have a clear color mark on the wrist position, which are also not practical solutions. Therefore, comparatively, the proposed method can achieve good forearm removal performance while the user performs hand gestures naturally.

4.2.2 Hand Size Normalization

Generally speaking, the hand region in an acquired image can have different sizes due to various distances that might be imposed between the hand and the camera during the hand gesture image acquisition phase. Besides, the size of user’s hand also contributes

to this effect. Because the RT is a scale sensitive transform, any input pattern need to be normalized to a fixed size before they can be represented by the RT. Since the the largest circle inscribed inside the palm region identified in the previous stage can be used to represent the size of the hand region, the radius of the circle is normalized to a fixed size 35, which is empirically determined in this work.

Figure 4.5: Some demonstrations of skin detection, binarization and forearm removal process. Column (a) represents the initial image acquired by the camera, and column (b) represents the detected skin region. By applying the Otsu binarization algorithm [44], Column (c) represents the binarization result (including fingers, palm and forearm region). Column (d) shows the largest circle identified over the palm region, and column (e) represents the hand region with forearm removed.

4.3 RT-based Feature Extraction and Classification

The normalized binary hand gesture patterns are then processed by the RT via Eq. 3.1 and Eq. 3.2. In this work, a separable approach as discussed in Chapter 3 is exploited. The computed RT results are treated as the salient features extracted for conducting hand gesture recognition.

After that, the *nearest-neighbor* (N-N) classification is considered in this work to facilitate the recognition task, since the number of template gesture classes will be quite limited (typically, about 10-20 classes). The similarity between the input pattern and each template pattern stored in the database will be measured to find the best match. From the viewpoint of practical usage, it would be very difficult, and unreasonable, to ask the user to perform each of his/her hand gesture to be exactly the same as the one initially stored in the system template database each time. To address this concern, some variations of the initially-performed hand gesture pattern have been automatically created by the system and added into the template database for pattern recognition. With this design, the system becomes more user-friendly and more robust on recognition performance. In other words, each hand gesture pattern has a *set* of template patterns, not just one pattern only. For one user, the proposed hand gesture recognition system achieves 98.0% recognition rate in this experiment. Details of these experiments are given in the simulation results part. In this work, the N-N classifier is used for conducting template matching. The N-N classifier assigns an input gesture to the class of its nearest neighbor in the feature space. To be more specific, consider z as the feature vector generated by combining the numbers in the RT result column by column. And z_s and z_t represent the feature vector of the input image and the template image, respectively. The N-N classifier will compute the distance between z_s and z_t as

$$d_t(z_s, z_t) = \sqrt{\sum_{j=1}^M (z_{sj} - z_{tj})^2}, \quad (\text{Eq. 4.1})$$

with M being the dimension of the feature vector and t being the index of the template gesture in the database. The input gesture pattern is then assigned to the gesture class t as

$$t = \underset{t}{\operatorname{argmin}} d_t(z_s, z_t), \quad (\text{Eq. 4.2})$$

which is closest to the input gesture in the feature space.

Chapter 5

Simulation Results

In simulation, 12 hand gestures are used to replace the 12 buttons on any mobile phone (Fig. 5.1). On the use of the proposed static hand gesture recognition algorithm in real-life applications, it is expected that the input hand gesture acquired by a mobile device might easily involve certain kinds of variations during the acquisition process, such as translation, scale, orientation, and tilt. Therefore, extensive simulation experiments of hand gesture recognition under these variations have been conducted for the performance evaluation of the proposed RT-based approach, and the simulation results (in terms of recognition rate) are documented in this chapter. Consider, nowadays, the use of mobile device becomes quite personal, thus this evaluation is conducted from the standpoint of *user-dependent* scenario.

As the proposed user-dependent hand gesture recognition algorithm is targeted to be deployed in personal mobile devices (e.g., smart phones), the user is required to establish his own gesture template set at the very beginning of using the mobile device. Thus this user-dependent hand gesture recognition algorithm can be easily used on other gestures not initially included in the system, which in turn provides user-customize property. In this RT-based approach, a set of templates consisting of one hand-gesture pattern together with some variations are established to represent that hand-gesture pattern. So there are 12 gesture templates in total under simulation that represent the 12 buttons

Figure 5.1: The twelve buttons on every mobile phone (numbers 0-9, “*”, and “#”). The 12 hand gesture patterns in the test set are exploited to replace these buttons in this work.

(numbers 0-9, “*”, and “#”) commonly available in all phone sets (refer to Fig. 5.2 for example, in which all gestures are from the same user). Furthermore, since this system is targeted for the use in a mobile environment, several most likely incurred variations such as translation, scale, orientation and tilt, will be inevitably imposed on a query gesture pattern under recognition are thoroughly studied in this chapter.

5.1 Recognition Performance Under Various Pose Variations

1) Translation variation: As discussed in Chapter 3, the RT is shift invariant. That is, the same gesture appeared at different positions will, in fact, yield the same RT result as shown in Fig. 3.4. So the translation variation become no issue at all in the RT-based hand gesture recognition system.

2) Scale variation: Note that the RT is scale sensitive; it means the size of the detected hand region matters. To consider this factor, the size of the detected hand

Figure 5.2: Examples of the twelve gesture types used in the experiments, in which all gestures are from the same user

region is normalized in the hand detection phase before the pattern is processed by the RT for extracting its features. Due to this normalization process, the recognition accuracy influenced by the scale variations is minimized and becomes no issue.

3) Orientation variation: In real applications, it is reasonable to assume that hand gesture can be in various poses with different orientations (see Fig. 5.3). To evaluate the recognition performance of the RT-based method against hand gestures with different

Figure 5.3: An illustration of tilted gesture patterns and gestures with orientation variation (better viewed in color).

orientations, a test set of gestures with orientation variation is created. This test set contains 12 types of gestures from a single user and each gesture type is performed 20 times by the same user. These patterns are then subjected to rotation with 1° or -1° each time over the range of $[-90^\circ, 90^\circ]$ using Matlab (thus 180 gesture patterns are generated). Some examples of the generated gesture patterns with different orientation is shown in Fig. 5.4.

In total, this test set has $12 \times 20 \times 180 = 43200$ gesture patterns and it is used to identify the range of pattern orientation that RT-based algorithm can recognize accurately. Note that all test images in this work are acquired by a popular mobile device,

Figure 5.4: Examples of the generated gesture patterns with different orientation using Matlab.

HTC Sensation (G14) mobile phone. The choice of this kind of device is to demonstrate that this algorithm can be exploited in any real mobile device as long as it is equipped with a camera, such as hand phone or Google glass.

The recognition accuracy is shown in Fig. 5.5. From Fig. 5.5, It can be seen that the RT-based method can be used to recognize gestures with orientation variation within a certain degree accurately. To be more specific, gestures with orientation ranging from $[-41^\circ, 43^\circ]$ can reach a recognition rate higher than 90%, which is sufficient for practical applications. With the orientation angle of the hand gestures becomes bigger, the recognition accuracy drops gradually.

4) Tilt variation: Besides orientation, it is reasonable to expect that the input

Figure 5.5: Recognition accuracy of gestures with different orientations. Approximately within the range of $[-40^\circ, 40^\circ]$, the hand gesture patterns can be recognized correctly and efficiently.

gestures imposed by the user might have some light degree of tilt-up or tilt-down on the gesture pose, since the user's hand might lean toward to or away from the camera lens along its axial direction as shown in Fig. 5.3. To our best knowledge, other state-of-the-art hand gesture recognition methods have not considered such practical scenarios and the resulted performance. To evaluate the recognition performance of the RT-based algorithm against tilted gestures, a set of tilted hand gesture test images are acquired. In this test set, a user was asked to perform 10 tilted-up gestures and 10 tilted-down gestures individually for each of the 12 gesture pattern types. As it is not practical to measure the angles of the tilted gesture, the user was asked to perform reasonable tilted

Figure 5.6: Examples of severe tilted gestures the proposed method failed to recognize.

gesture as if he would do in real-life situation, as well as some severe tilted gestures. In total, this test set has $(10 + 10) \times 12 = 240$ patterns. All of the reasonable tilted gestures are recognized correctly, while a few of the severe tilted gestures failed to be recognized. The performance shows that the RT-based algorithm can deal with reasonable tilted gestures effectively. When the tilt-angle is severe (not practical in real-life situation), this algorithm begins to mis-classify gesture patterns. Several severe tilted gesture patterns failed to be recognized is shown in Fig. 5.6.

5) Performance evaluation: In this part, the RT-based approach is tested in real-life situation, which means that the query gesture patterns can suffer from different variations, namely scale, position, orientation and tilt. To simulate the real-life situation, a test set consisting of 1200 gesture patterns is collected. These gesture patterns are collected from 5 different users, and it contains the same 12 types of gestures as before. Each user performed 20 different poses for each type of gesture as if he was in real-life situations. Since there are 12 types of hand gesture patterns, thus there are $5 \times 20 \times 12 = 1200$ patterns in total. Note that these patterns are performed by different users practically, so it covers reasonable translation, scale, orientation and tilt variations to

Table 5.1: Recognition accuracy of different users

Subject	Recognition Rate
User 1	97.91%
User 2	97.91%
User 3	98.75%
User 4	97.91%
User 5	97.50%
Average	98.00%

simulate real-life situations.

The result of this experiment is given in Fig. 5.7 and Table 5.1. The experiment result is analyzed in two aspects. 1) In Fig. 5.7, It can be seen that for one user, the lowest recognition rate of any gesture pattern is 90%. This means that out of the 20 gestures an user performed, at most 2 of them are not recognized correctly. Note that in average, 9 of the 12 gesture categories reach 100% recognition rate. 2) From Table 5.1, we can see the RT-based algorithm can provide high recognition accuracy with stable performance. The five users tested reaches an average recognition rate of 98.0%. The lowest user's recognition accuracy is 97.5%, which is much acceptable in real-life applications. Therefore, the RT-based approach is proved to be reliable as an user-dependent hand gesture recognition algorithm.

5.2 Evaluation of User-independent RT-based Approach

In some applications such as remote controlling a smart TV or interaction with a vending machine using HGR, user-independent hand gesture recognition is of great importance.

For each gesture type, as discussed before, a set of templates consisting of one hand-gesture pattern together with some variations are established and pre-stored in the tem-

Figure 5.7: Details of the user-dependent experiment results with a set of templates for each user. The results of the five users are marked with different colors, respectively.

plate database. In the user-independent scenario, different users share the same pre-stored templates in the system to recognize hand-gesture patterns.

The test set in this experiment is the same with the one in the performance evaluation section. The performance is compared with existing state-of-the-art feature types, including HOG [27], MCT [24], Shape Context [45] and MSBNM [31]. The performance of these four features are tested by Hu [31] in a very similar test set. In his work, twelve types of static hand gestures as shown in Fig. 5.8 are acquired in a laboratory environment with background to be not so complicated, which is very similar to our assumption. Furthermore, 11 out of the 12 gesture pattern types in our test set are the same as his. And the last one from our test set is more difficult to be recognized than his, because it is more similar to other hand gesture patterns in our test set. So it is reasonable to compare results of our experiments with his.

In Hu's work, MSBNM, MCT and HOG features are trained on 3 different classifiers, namely, *Decision Tree* (DT), *Bayesian Network* (BN) and *Classification via Regression* (CR), respectively. The performance (shown in Table 5.2) of each feature is represented

Figure 5.8: The 12 gesture types used in [31]. As shown, 11 out of 12 gesture patterns are the same with our test gesture patterns.

Table 5.2: Performance comparison of each feature representation

Features	Classifier	Performance
Shape Context [45]	N-N	85.0%
MCT [24]	BN/CR	88.6%
HOG [27]	CR	90.1%
MSBNM [31]	DT	94.8%
RT (proposed)	N-N	96.7%

by the optimum accuracy across all classifiers. The performance of Shape Context is evaluated using the N-N classifier suggested in its reference. Our RT-based method is also evaluated using the N-N classifier as discussed before. The accuracy of each feature representation is given in Table 5.2. From Table 5.2, we can see the RT-based approach outperforms other state-of-the-art methods. The confusion matrix of the recognition result of the RT-based method is shown in Fig. 5.9. The average recognition rate is 96.7%, 1.9% higher than the method proposed by Hu.

		Template Gesture Patterns											
		1	2	3	4	5	6	7	8	9	10	11	12
Input Gesture Patterns	1	1.0											
	2		0.99										0.01
	3		0.04	0.91	0.01			0.02					0.02
	4				1.0								
	5				0.03	0.95				0.02			
	6		0.02				0.98						
	7							1.0					
	8	0.01	0.01					0.02	0.96				
	9					0.03		0.05		0.92			
	10										1.0		
	11							0.01			0.02	0.97	
	12		0.03	0.03				0.01	0.01				0.92

Figure 5.9: The confusion matrix of 12 gesture types. Each row represents the recognition accuracy of one type of gesture patterns.

For further comparison with the state-of-the-art methods on the exactly same test data set, the RT-based method is also tested on a dataset of Jochen Triesch Static Hand Posture Dataset [33]. In this simulation, 4 gesture types which are included in our test set are selected from their data set. The four gesture types are given in Fig. 5.10. In total, 96 test images, collected from 24 different people were used for testing.

Since the hand gesture images in Jochen Triesch Dataset is in grey scale, the input test images are directly binarized without the skin color detection phase. The accuracies of classification on this test set are shown in Table 5.3.

Figure 5.10: The 4 gesture types selected in Jochen Triesch Static Hand Posture Dataset. These gesture types corresponds with gesture 10, 1, 2, 8 in our test set respectively.

From Table 5.3, it can be seen that the proposed RT-based method still outperforms all of the other methods on Jochen’s dataset. Furthermore, in Hu’s experiment, 1200 training hand gesture images were acquired for this task. On the other hand, in the proposed RT-based algorithm, the training phase is not required. Hand gesture templates are created using the gesture images in Jochen’s data set. As it is for the simulation of an user-independent scenario, only one image of each gesture type is randomly selected from the data set and regarded as the gesture pattern template. This process is done 10 times to investigate the recognition accuracy with different sets of templates. The accuracies of each set of pattern templates are documented in Table 5.4. As shown in Fig. 5.11, the lowest recognition rate of the proposed RT-based method is 0.5% less than MSBNM, and the highest is 8.2% higher than MSBNM. The average recognition rate of the proposed RT-based method is 87.8%, which is 3.4% higher than MSBNM.

5.3 Current Limitations

In this section, the proposed RT-based hand gesture recognition method is tested on several test sets and compared with other state-of-the art methods. As a result, the proposed method is proved to be a robust hand gesture recognition method. However,

Table 5.3: Performance comparison on Jochen’s dataset

Features	Classifier	Performance
Shape Context [45]	N-N	33.3%
MCT [24]	BN/CR	25.0%
HOG [27]	CR	25.0%
MSBNM [31]	DT	84.4%
RT (proposed)	N-N	87.8%

Table 5.4: Performance with different sets of templates

Template Set Number	1	2	3	4	5	6	7	8	9	10
Accuracy	90.3	88.7	84.7	83.9	84.7	86.7	89.9	87.8	92.6	88.7

there are some limitations at this stage, and may need further research in the future.

Hand region segmentation In the simulation, the test set are collected under a controlled background, which means that the background can not be too complicated. With this assumption, hand detection and segmentation using skin color information is adequate. However, this is not always the case. In real-life scenarios, the background could be much more complicated. Skin color will be detected in both foreground (hand region) and background. This will lead to an unsatisfying hand region segmentation result.

Forearm removal In the forearm removal phase described in chapter 4.2.1, the biggest circle inscribed in the hand region is used to identify the palm region, and then pixels below the palm region is regarded as forearm and removed. In practical situation, this forearm removal method requires the input hand gesture pattern to have a clear full palm region. Gestures shown in Fig. 6.2 cannot be recognized in this proposed HGR system because the palm region cannot be detected correctly, and the forearm region cannot be removed accordingly.

Figure 5.11: A comparison of the 5 features on Jochen's dataset. The proposed RT-based method outperforms other methods.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, a novel *hand gesture recognition* (HGR) method is developed that can be deployed in any mobile device equipped with a RGB camera for acquiring the posed hand gesture image under recognition. Since the proposed system has 96.7% recognition accuracy while others only have 94.8% at most as shown in Chapter 5.2, the proposed system can deliver relatively accurate and robust HGR performance in real time. Also, because only $+$ and $|-|$ operators are used in the RT algorithm, the computational complexity is very low compared with other methods. In the proposed method, the hand gesture image is first acquired by a normal RGB camera. Then, the hand region is detected by using the skin color information and further binarized. After that, palm size normalization are conducted as the pre-processing steps before the *rapid transform* (RT) is exploited to the processed binary image for extracting its salient features. The main reasons of choosing the RT are that it is not only shift invariant, but also nearly (or practically) invariant to rotation up to a certain degree.

Extensive simulation experiments have been conducted to test the accuracy and efficiency of the proposed hand gesture recognition system as documented in Chapter 5. As shown in Chapter 5, the proposed system is completely invariant of translation variation and highly robust to a certain degrees of rotation approximately covering the range of

Figure 6.1: Examples of the hand gesture images acquired under complicated background.

$[-40^\circ, 40^\circ]$. Although the RT is *scale* sensitive, however, this issue has been overcome by applying a simple normalization operation on the detected hand region. In conclusion, the proposed RT-based hand gesture recognition system can be used to recognize hand gesture patterns with a common RGB camera robustly and efficiently.

6.2 Future Work

For future work, there are several directions, which are worthy of conducting further investigation. The first one is to continue this topic on hand gesture recognition by considering other possible interferences that might be encountered by the user in real-life scenarios, such as low lighting condition and complicated background. An example of the hand gesture images acquired under complicated background is shown in Fig. 6.1. Under these conditions, the hand region could be more difficult to be segmented from the input image.

Besides, as discussed in Chapter 5, there are several limitations at this stage of work. By continuing focusing on dealing with these limitations, the proposed hand gesture recognition method can be used to recognize more hand gesture patterns. For example,

Figure 6.2: Some examples of the hand gesture patterns that the RT-based method cannot recognize correctly. Note that the palm region in these hand gesture patterns cannot be represented by the largest circle inscribed in the hand region.

Figure 6.3: An example of some sign language gestures [4].

other methods to remove the forearm region could be investigated when the user intends to impose a hand gesture. This will make the proposed hand gesture recognition system able to recognize more hand gestures, such as the ones shown in Fig. 6.2. These hand gesture patterns can not be recognized correctly in the current hand gesture recognition system as discussed in Chapter 5.

Another possible direction could be introducing the RT into sign language recognition. As a typical dynamic hand gesture recognition task, sign language can be viewed as a series of static hand gesture patterns. Examples of sign language recognition [4] can be seen in Fig. 6.3. The RT may also be able to extract the salient features in this task and be further used for recognition.

References

- [1] A. Chaudhary, J. L. Raheja, K. Das, and S. Raheja, “Intelligent approaches to interact with machines using hand gesture recognition in natural way: a survey,” *International Journal of Computer Science and Engineering Survey*, vol. 2, no. 1, pp. 122-133, 2011.
- [2] A. Malima, O. Erol, and C. Mujdat, “A fast algorithm for vision-based hand gesture recognition for robot control,” *Signal Processing and Communications Applications*, pp. 1-4, 2006.
- [3] A. Licsár and T. Szirányi, “User-adaptive hand gesture recognition system with interactive training,” *Image and Vision Computing*, vol. 23, pp. 1102–1114, 2005.
- [4] S. Mitra and T. Acharya, “Gesture Recognition: A Survey,” *IEEE Transactions on systems, Man, and Cybernetics*, vol. 37, no. 3, pp. 311–324, 2007
- [5] K. Derpanis and S. Jadon, “A Review of Vision-Based Hand Gestures,” *International Journal of Information Technology and Knowledge* , vol. 2, no. 2, pp. 405-410, 2009
- [6] Z. Khan, and N. Ibraheem, “Survey on gesture recognition for hand image postures,” *Computer and Information Science* , vol. 5, no. 3, 2012
- [7] V. Pavlovic, , R. Sharma, and T. Huang, “Visual interpretation of hand gestures for human-computer interaction: A review,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 677 - 695, 1997.
- [8] <https://www.thalmic.com/en/myo/>

REFERENCES

- [9] M. Bray, E. Koller-Meier, and L. Van Gool, “Smart particle filtering for 3D hand tracking,” in *Proceeding of International Conference on Automatic Face and Gesture Recognition (FGR)* , pp. 675 - 680, 2004.
- [10] G. Dewaele, F. Devernay, and R. Horaud, “Hand motion from 3D point trajectories and a smooth surface model,” in *Proceeding of European Conference on Computer Vision* , pp. 495 - 507, 2004.
- [11] C. Chua, H. Guan, and Y. Ho, “Model-based 3D hand posture estimation from a single 2D image,” *Image and Vision Computing* , vol. 20, pp. 191 - 202, 2002.
- [12] M. Panwar, “Hand gesture recognition based on shape parameters,” in *Proceeding of International Conference on Computing, Communication and Application* , pp. 22 - 24, 2012.
- [13] Y.-T. Li, P. Juan, “HEGM A hierarchical elastic graph matching for hand gesture recognition,” *Progress in Pattern Recognition, Image Analysis, Computer Vision and Applications* , pp. 308 - 315, 2012.
- [14] L. Bretzner, I. Laptev, and T. Lindeberg, “Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering,” in *Proceeding of International Conference on Automatic Face and Gesture Recognition (FGR)* , pp. 423 - 428, 2002.
- [15] E. Holden, “Visual recognition of hand motion,” Ph.D thesis, Department of Computer Science, University of Western Australia, 1997.
- [16] N. Shimada, Y. Shirai, Y. Kuno, and J. Miura, “Hand gesture estimation and model refinement using monocular camera-ambiguity limitation by inequity constraints,” in *Proceeding of International Conference on Automatic Face and Gesture Recognition (FGR)*, pp. 268 - 273, 1998
- [17] D. Lowe, “Fitting parameterized three-dimensional models to images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 441 - 450, 1991

REFERENCES

- [18] Z. Ren, “Shape based hand gesture recognition,” Ph.D thesis, School of Electrical & Electronic Engineering, Nanyang Technological University, 2012.
- [19] J. Suarez, R. Murphy, “Hand gesture recognition with depth images: A review,” in *IEEE International Symposium on Robot and Human Interactive Communication*, pp. 411 - 417, 2012
- [20] Y. Li, “Hand gesture recognition using Kinect,” in *IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 196 - 199, 2012
- [21] H. Lee and J. Kim, “An hmm-based threshold model approach for gesture recognition,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 961 - 973, 1999
- [22] A. Wilson and A. Bobick, “Parametric hidden markov models for gesture recognition,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 884 - 900, 1999
- [23] H. Zhou, D. Lin and T. Huang, “Static hand gesture recognition based on local orientation histogram feature distribution model,” in *Proceeding of Workshop on CVPR*, pp. 161–161, 2004.
- [24] A. Just, Y. Rodriguez and S. Marcel, “Hand Posture Classification and Recognition using the Modified Census Transform,” in *Proceeding of International Conference on Automatic Face and Gesture Recognition (FGR)*, pp. 351-356, 2006.
- [25] S. Priyal and P.K. Bora, “A robust static hand gesture recognition system using geometry based normalizations and krawtchouk moments,” *Pattern Recognition*, vol. 8, pp. 2202-2219, 2013.
- [26] P. Barros, N. Junior, J. Bisneto, B. Fernandes, B. Bezerra and S. Fernandes, “Convexity local contour sequences for gesture recognition,” in *Proc. 28th Annual ACM Symposium on Applied Computing*, 2013.
- [27] Y. Song, D. Demirdjian and R. Davis, “Tracking body and hands for gesture recognition: Natops aircraft handling signals database,” in *Proceeding of International Conference on Automatic Face and Gesture Recognition (FGR)*, pp. 500–506, 2011.

REFERENCES

- [28] <http://www.chilton-computing.org.uk/gallery/ral90/orig/r90r7049.jpg>
- [29] <http://www.creativeapplications.net/news/kinect-opensource-news/>
- [30] <https://www.leapmotion.com/>
- [31] K. Hu and L. Yin, “Multi-scale Topological Features for Hand Posture Representation and Analysis,” in *Proceeding of International Conference on Computer Vision*, pp. 1928–1935, 2013.
- [32] D. Sturman and D. Zeltzer, “A survey of glove-based input,” *Computer Graphics and Applications*, pp. 30–39, 1994.
- [33] J. Triesch and C.v.d. Malsburg, “A system for person-independent hand posture recognition against complex backgrounds,” *IEEE. Trans. Pattern Anal. Mach. Intell.*, pp.1449–1453, 2001.
- [34] H. Reitboeck and T. Brody, “A transformation with invariance under cyclic permutation for applications in pattern recognition,” *Inf. Control*, vol. 15, pp. 130–154, 1969.
- [35] P. Wang and R.C. Shiao, “Machine recognition of printed chinese characters via transformation algorithms,” *Pattern Recognition*, vol. 5, pp. 303–321, 1973.
- [36] K.-K. Ma, P. Paul, and J. Wang, “Automatic recognition of low resolution tactile sensing data using rapid transformation,” *Robotics and Artificial Intelligence*, vol. 11, pp. 159–170, 1984.
- [37] J. Ma, C. Wu, and X. Lu, “A fast shape descriptor,” *Computer Vision, Graphics, and Image Processing*, vol. 34, pp. 282–291, 1986.
- [38] F. Müller and A. Mertins, “Contextual invariant-integration features for improved speaker-independent speech recognition,” *Speech Commun*, vol. 53, pp. 830–841, 2011.
- [39] C. Li, J. Li, D. Gao, and B. Fu, “Rapid-transform based rotation invariant descriptor for texture classification under non-ideal conditions,” *Pattern Recognition*, vol. 47, pp. 313–325, 2014.

REFERENCES

- [40] V. Spruyt, A. Ledda, and S. Geerts, “Real-time multi-colourspace hand segmentation,” in *Proceeding of International Conference on Image Processing (ICIP)*, pp. 3117–3120, 2010.
- [41] M. Kolsch and M. Turk, “Robust hand detection,” in *Proceeding of International Conference on Automatic Face and Gesture Recognition (FGR)*, pp. 614-619, 2004.
- [42] A. Mittal, A. Zisserman, and P.H.S. Torr, “Hand detection using multiple proposals,” in *Proceeding of British Machine Vision Conference (BMVC)*, 2011.
- [43] X. Teng, B. Wu, W. Yu, and C. Liu, “A hand gesture recognition system based on local linear embedding,” *Journal of Visual Languages and Computing*, vol. 16, pp. 442–454, 2005.
- [44] N. Otsu, “A threshold selection method from gray-level histograms,” *Automatica*, vol. 11, pp. 23–27, 1975.
- [45] S. Belongie, J. Malik and J. Puzicha, “Shape matching and object recognition using shape contexts,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 509-522, 2002.
- [46] D. Lowe, “Object recognition from local scale-invariant features,” in *Proceeding of IEEE International Conference on Computer Vision* , vol. 2, pp. 1150-1157, 1999.