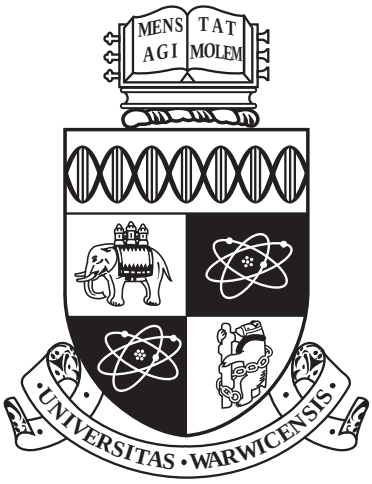

Hand Gesture Recognition in Uncontrolled Environments

by

Yi Yao

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Department of Computer Science

December 2014

THE UNIVERSITY OF
WARWICK

Contents

List of Tables

List of Figures

Acknowledgments

The journey of completing my PhD research has significant influences on my life. It taught me the meaning of responsibility and perseverance. This journey was made possible because of the support, guidance and encouragement I got from many individuals. I would like to take this opportunity to express my gratitude to all of them.

First of all, I would like to express my gratitude to my supervisor, Prof. Chang-Tsun Li. For the past few years, despite my ignorance and sloth, Prof. Li introduced me to the world of science, and taught me everything I know about how to conduct scientific research. He provided me many opportunities that I could only dream about. I hope I can inherit some of his work ethics and many valuable qualities as a person. He is not only my supervisor, he also sincerely cares about my life. My wife and I have received countless helps from Prof. Li and his lovely family over the years. We will never forget that.

This journey would be much harder without my lab mates. Xingjie Wei, Yu Guan, Xin Lu, Xufeng Lin, Ruizhe Li, Ning Jia, Qiang Zhang, Xin Guan, Roberto Leyva, Alaa Khadidos, Faisal Azhar, Chao Chen, Bo Gao, Huanzhou Zhu, you are all exceptional scientists. Many ideas of mine came from our heated discussions. We worked hard together and had fun together, thank you all for being my friends and accompanying me along this journey. Our friendships won't stop here.

I owe my deepest gratitude to my parents, who always give me unconditional love and support. Without your encouragement, I would never finish this PhD. Thanks to my father who always is my role model, for all the helps on my career.

Thanks to my mother who has been supporting me and my marriage thousands miles away from China since day one of my PhD. I hope I can make you both proud. Also, thanks to my parents in-law, who gave me permission to take their little girl half the world away for years.

Finally, special thanks to the love of my life, my best friend and wonderful wife Jing Ma, who literally did this PhD with me. Thank you for staying with me in the lab every day, thank you for tolerating my long hours, thank you for standing right by my side and holding my hand during all the ups and downs. You saved me and changed me from a boy to a man.

Declarations

I hereby declare that, except where acknowledged, the work presented in this thesis is my own work. No part of the work contained in this thesis has previously been accepted in substance for any degree nor submitted elsewhere for the purpose of obtaining an academic degree.

Abstract

Human Computer Interaction has been relying on mechanical devices to feed information into computers with low efficiency for a long time. With the recent developments in image processing and machine learning methods, the computer vision community is ready to develop the next generation of Human Computer Interaction methods, including Hand Gesture Recognition methods. A comprehensive Hand Gesture Recognition based semantic level Human Computer Interaction framework for uncontrolled environments is proposed in this thesis. The framework contains novel methods for Hand Posture Recognition, Hand Gesture Recognition and Hand Gesture Spotting.

The Hand Posture Recognition method in the proposed framework is capable of recognising predefined still hand postures from cluttered backgrounds. Texture features are used in conjunction with Adaptive Boosting to form a novel feature selection scheme, which can effectively detect and select discriminative texture features from the training samples of the posture classes.

A novel Hand Tracking method called Adaptive SURF Tracking is proposed in this thesis. Texture key points are used to track multiple hand candidates in the scene. This tracking method matches texture key points of hand candidates within adjacent frames to calculate the movement directions of hand candidates.

With the gesture trajectories provided by the Adaptive SURF Tracking method, a novel classifier called Partition Matrix is introduced to perform gesture classification for uncontrolled environments with multiple hand candidates. The trajectories of all hand candidates extracted from the original video under different

frame rates are used to analyse the movements of hand candidates. An alternative gesture classifier based on Convolutional Neural Network is also proposed. The input images of the Neural Network are approximate trajectory images reconstructed from the tracking results of the Adaptive SURF Tracking method.

For Hand Gesture Spotting, a forward spotting scheme is introduced to detect the starting and ending points of the predefined gestures in the continuously signed gesture videos. A Non-Sign Model is also proposed to simulate meaningless hand movements between the meaningful gestures.

The proposed framework can perform well with unconstrained scene settings, including frontal occlusions, background distractions and changing lighting conditions. Moreover, it is invariant to changing scales, speed and locations of the gesture trajectories.

Chapter 1

Introduction

1.1 Human Computer Interaction

The methods of inputting information into computers have significant influences on system efficiency, system usability and user experience. People have been using mechanical devices as input methods since the advent of electronic computers [11]. The history of Human Computer Interaction (HCI) began in 1959 [12] with Shakel's article on "*The ergonomics of a computer*" [13]. People have been developing novel HCI methods since then, from the Atari 2600 Joystick in 1977 [11], the first commercially successful integrated keyboard in personal computer IBM PC 84 key keyboard in 1981 [14] and the first widely used mouse on Apple Lisa PC in 1983 [15], to the huge success of Microsoft Kinect in 2009 [16]. With the rapid growth in the hardware computational power and the accuracy of machine learning methods, the computer science research community has changed people's fundamental perspectives. However, we are still pushing buttons to feed information into machineries byte by byte with low efficiency. The needs for more intuitive and efficient ways to interact with computers are acute. People are seeking methods to "walk" in virtual reality, "talk" to computers and "wave" to control machineries. The novelty of the armband MYO [17] and LeapMotion [18], are two of the many proofs of the

fact that HCI just about to enter the new era of semantic level communications.

Computer vision based methods are becoming one of the most promising directions for the next generation HCI. Hand Gesture Recognition (HGR) as one of the earliest ideas of the semantic HCI, has already started the process of large scale commercialisation since the Microsoft Kinect 2009 [16]. More and more novel applications for HGR are emerging. This thesis is focusing on solving the bottleneck that the HGR research community is facing today: pure appearance based robust HGR method for uncontrolled environments.

1.2 Hand Gesture Recognition

According to Psycholinguistcs gesture is the "critical link between our conceptualizing capacities and our linguistic abilities" [19]. In Biology, gesture is defined as "all kinds of instances where an individual engages in movements such that their communicative intent is manifested and openly recognized" [20]. Gestures are utilised in communications as part of the expression. Sometimes gestures alone can be used as a communicative language [21] [22] [23]. Due to the important role of gestures in human communication, gesture recognition has been explored in many multimedia applications [24, 25, 26, 27, 28, 29, 30].

The tasks for Hand Gesture Recognition vary under different contexts. Generally, there are two main tasks: Hand Posture Recognition (HPR) and Hand Gesture Recognition. For HPR, the task is to recognise different hand postures based on the spatial features extracted from the static images [31, 2, 32, 33, 34, 35, 36, 37, 38]. The task of HGR could be combination of analysing the latent patterns based on both the temporal and spacial features extracted from the trajectories and shapes of hands in video streams [39, 40, 41, 42, 43, 44, 45, 46]. For some specific applications, the functionalities of HPR and HGR are usually combined, which means detecting the presence of certain hand posture, then analysing its trajectory. In the research

field of HGR, some works on HPR still claim that the methods are under the scope of HGR. Hence there is ambiguity in the definition of HGR in the research field. This thesis separates the definition of hand gestures and hand postures, and novel solutions for both HPR and HGR in uncontrolled environments are proposed. Due to the similarity between HPR and HGR, the principles of HGR methods can also be applied to HPR methods. Hence, HPR is treated as a sub-task of HGR in this thesis. The content about HPR methods can be found within the content of HGR, without dedicated titles. Additionally, another sub-task of HGR to recognise continuous hand gestures is called Hand Gesture Spotting (HGS), which can be seen as a branch of HGR research. Contents about HGS can be found in Chapter 7.

There are three key components of the HGR process: 1) Hand Segmentation and Tracking, 2) Trajectory Feature Extraction and Selection, 3) Gesture Classification. Hand Segmentation and Tracking can be seen as the pre-processing of HGR, which is about detecting possible hand candidates in the scene, and record the positions and trajectories of the hand candidates through video streams. For various background scene settings, the task of hand segmentation and tracking can be very different. Hand Tracking in the uncontrolled environments with intensely distracted backgrounds has always been a difficult problem for the HGR research community. But at the same time, it is one of the crucial challenges that must been tackled before HGR can be widely used in real-world applications. In this thesis, a novel hand tracking scheme is presented specially for uncontrolled environments. Similar with other pattern recognition problems, classification process can always benefit from discriminative features. Different combinations of trajectory features can produce diverse invariance properties against changes of size, speed and orientation of the gesture trajectories. The more types of trajectory features adopted, the more discriminative the feature combination will be. But the less invariant the method would be against alignment and scale change (of course, some of the features are correlated and therefore redundant). Hence the trade-off between invariant

properties and feature discriminability can be decisive for HGR methods. Attempts on pairing various classifiers and features have been conducted in many works (see Chapter 2).

HPR needs pre-processing to detect the hand region as well. If the method is for real-time HPR, the method would also require a tracking scheme to keep record on the current position of the target hand region. Then instead of trajectory features, spatial features are extracted as the main feature for posture classification, including contour descriptors, texture features and colour cues.

The concept of HPR is perfectly clear, but when it comes to real-world applications, the definitions of HGR and HPR are often ambiguous, due to the reason that gestures in the vocabulary are normally combinations of dynamic hand gestures and still hand poses. Hence, the task becomes combination of Hand Gesture Recognition and Hand Posture Recognition. Hand poses can be used as an individual feature for HGR. Hence, for the majority of the commercialised methods, HPR is used as a complimentary method for HGR.

1.3 Problem Statement

The research of Hand Gesture Recognition started in the late 80s to the early 90s [47, 48, 49, 50, 51, 52, 53] to deal with problems such as Sign Language Recognition and Virtual Reality. The fundamental idea of the early stage HGR research is using pattern recognition and image processing techniques to interpret hand movement, without too much considerations for the further usability issues. The researchers were rather focusing on the basic HGR tasks in well-controlled lab environments. To this very day, the majority of the HGR community are focusing on Sign Language Recognition, where the main challenges are massive vocabulary, massive amount of classes and extremely low amount of training samples for each class. There were few attempts for HGR in uncontrolled environments in the academic community

Figure 1.1: Sample from Warwick Hand Gesture Database, with uncontrolled environments [4].

[54, 8, 2], however the robustness of these methods were far from satisfactory.

On the other hand, the commercialisation wave of HGR technologies began with the launch of Wii from Nintendo [55] and Kinect from Microsoft [16]. HGR as an innovation HCI method started to attract attention. People started to focus on usability and robustness against real-world scene settings, instead of complex pattern recognition problems with relatively low potential commercial values, such as Sign Language Recognition. Due to the lack of robust appearance based methods, industry oriented researchers came up with alternative approaches, that involve additional sensors such as stereo cameras [54], laser scanner [56], infrared cameras [16] and other mechanical sensors [17], to overcome challenges from the uncontrolled real life scene settings (see Fig 1.1). But the cost of integrating these relatively expensive sensors has been proven to be an issue for embedded systems and portable devices.

Developing a robust appearance based method for HGR in uncontrolled en-

vironments is the main contribution of this thesis. Due to the variety of HGR applications, the key parameters of HGR problems can largely vary. There are a large number of challenges that can significantly affect the performance of the HGR methods. In this thesis, the uncontrolled environments are define as: ***The environments with no constrains on the scene settings and the manner of gesture performance.*** The challenges that HGR community are currently facing, from controlled or uncontrolled environments, will be systematically defined in this section. The key challenges can be grouped into 4 categories. They are discussed below:

1.3.1 Vocabulary Structure

Vocabulary structure basically defines the task for HGR and HPR: What kind of gestures or postures the method wants to recognise.

- Large Vocabulary Size. In applications such as Sign Language Recognition, normally the task is recognising a large number of gestures and postures. Large amount of words means large amount of classes in classification process which is always a challenge for pattern recognition methods. It is a common knowledge that for pattern recognition tasks, the more training samples available, the better results the method will produce [57], of course if overfitting stays in reasonable level. For tasks like Sign Language Recognition, the amount of classes are massive. It is fairly difficult to collect a database with descent amount of samples for every word. The most popular dataset in the field is Boston ASL dataset [58]. It comprises hand signed short stories, dialogues, instead of individual words. In this way, more words would be covered in the dataset. But there are still no more than few dozens of samples for each words.

- High Gesture Similarity. Since the articulation limitation of the human upper body, including the 27 Degree of Freedom (DoF) in human hands [59, 60, 61],

the possible movements of joints in arms and hands are confined. Hence the amount of different hand gestures and postures (up to certain complexity) are limited. In order to present different words with limited articulation, introducing certain extent of similarity among words is inevitable. That means the inter-class variance is small. From pattern recognition point of view, small distinction among samples from different classes is always a problem. Especially for a massive vocabulary, gesture similarity is the main challenge.

- High Gesture Complexity. For some application, the gestures need to be defined with certain complex hand movements, such as Human Robot Interaction or Sign Language Recognition. Overly complex gestures can lead to confusion in classification, which also known as the sub-gesture problem [8]. Taking the task of HGR in this thesis as example, namely recognising 10 hand signed digits. The gesture "5" can be seen as first half of the gesture "8" (Fig. 1.2).

- Double Handed Gestures. In sign languages, two handed gestures are common. These gestures require tracking methods that can track multiple objects with similar texture, colour and contour features, without getting confused. Two handed gestures require more sophisticated tracking methods, hence normally less used in commercialised systems. For most of the applications, single handed gestures are more than enough to form the vocabulary. For interactive hand gestures utilised in HCI applications, the less occupied the users are by performing the gestures, the better user experience the system will produce. From usability point of view, simpler gestures would make the system easier to use. Therefore, in commercialised methods on the market, the vocabularies are filled with simple single-handed gestures, such as "swipes". There is no reason for requiring both hands of the gesture performer to use the system, while the single handed gestures are more than adequate. But two handed gesture still is one of the main challenges for some HGR problems.

Figure 1.2: Sub-gesture problem. On the left: a sample of gesture "5"; On the right: gesture "5" can be seen as part of gesture "8". The red and blue dots indicate start and end point of both gesture respectively.

1.3.2 Scene Settings

The scene settings are the contents of the scene environment other than the gesture performer.

- Changing Lighting Conditions. The content of the scene directly affects the tracking methods. Therefore the features of trajectory classification will also be affected. Lighting changes as one of the scene setting challenges, can affect the tracking method by giving object surfaces different colours and textures.

- Background Distractions. The main challenge in uncontrolled environments is background distractions. For further explanation, this challenge can be sub-categorised into 4 types: Static Background Non-Skin-Coloured Regions; Moving Background Non-Skin-Coloured Regions; Static Background Skin-Coloured Regions; Moving Background Skin-Coloured Regions. The 4 types of objects in the background can potentially present similar colour and texture features as the target hand region and confuse the tracking method. In Chapter 4, the influence of these challenges and corresponding solutions will

be discussed.

- Frontal Occlusion. Frontal occlusions means objects (moving or static) appearing in between the camera and the gesture performer, during certain time period of the gesture. Intuitively the solution for this challenge would involves using the last known feature of the target region before the occlusion occurred, or use partial features from the regions that have not been covered by the occlusion. In Chapter 4, this issue will be discussed in details.

1.3.3 Performance Constrains

Performance Constrains: The definition of performance constrains is the challenges caused by the manner of gesture performing.

- Continuous Gestures. For some specific application scenarios, the HGR methods are facing the task of recognising continuous gestures, with interconnection hand movements. Namely, detecting the starting and ending points of each gestures in a continuous gesture stream, similar with phrasing all words in a sentence. This particular task is called Hand Gesture Spotting (HGS). For every frame, the method should be able to determine whether it is part of predefined gestures, or meaningless hand movements.

- Face/Hand Overlapping. If the hand is overlapping with face regions during gestures, that could cause a tracking error for methods based on colour features. Facial region of the gesture performer can also be seen as a background distraction. In early studies of HGR without the consideration of background distractions, this challenge was one of the main obstacles.

- Hand Out of Scene. During the gestures, if the target hand is stepping out of the scope of the camera, only leaving the wrist or arm region in the scene, there would be no colour, texture or motion features can be extracted on the

target hand region. Hence the tracking method can easily lost track on the hand.

- Pause During Gesture. Pauses during the gestures can be catastrophic for HGR methods that extract speed or position as trajectory features. In Chapter 4, solution for this challenge will be introduced.

1.3.4 Intra-class variance

Intra-class variance includes variance on the gesture size, speed, location and orientation. HGR methods are suffering from large intra-class variance as other pattern recognition problems. Performing hand postures or gestures in the air with no reference objects can cause largely varying posture or gesture patterns. People have different level of brain-hand coordination and spatial imagination abilities, which can affect how people draw signs in the air. Even with specific instructions of standard postures or gestures, same hand posture or gesture samples performed by different individuals could have a distribution with large variance in the feature space. Even the same performer signing the same posture or gesture, the results could still vary in sizes, speed, locations and orientations. That presents a challenge for classification.

1.4 Map of this thesis

In the following chapters, a framework for Hand Posture Recognition, Hand Gesture Recognition and Hand Gesture Spotting in uncontrolled environments is introduced (Fig 1.3). Chapter 2 contains a detailed review on existing methods of the commercial and academic communities of HPR and HGR. Chapter 3 introduces a solution for Hand Posture Recognition in uncontrolled environments. In Chapter 4, a novel tracking scheme called Adaptive SURF Tracking is proposed for hand tracking in uncontrolled environments. Following that, Chapter 5 introduces a novel gesture

classifier based on Hidden Conditional Random Fields, for gesture classification in uncontrolled environments with multiple hand candidates in the scene. In Chapter 6, a forward hand gesture spotting scheme is proposed for Hand Gesture Spotting in uncontrolled environments. The final chapter presents conclusions and possible future directions of the proposed framework.

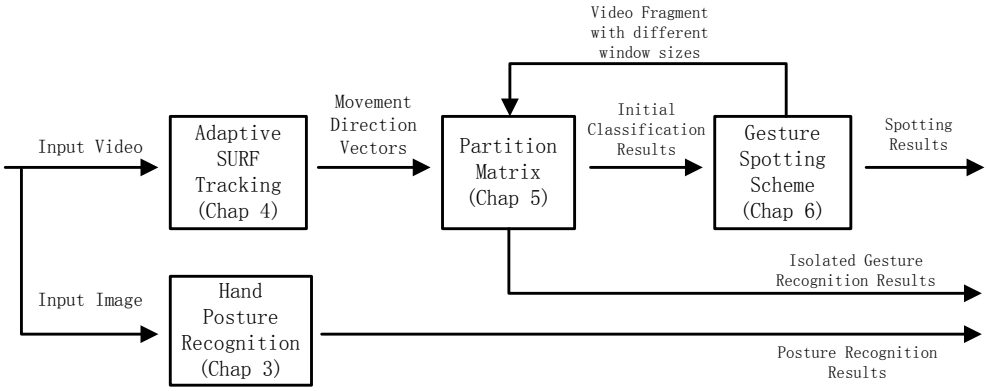

Figure 1.3: The structure of the proposed Hand Gesture Recognition framework in this thesis.

Chapter 2

Literature Review

As listed in Section 1.3, the key aspects that define the complexity of the HGR task in different contexts include: 1) vocabulary structure (vocabulary size, gesture similarity and complexity, and double handed gestures), 2) scene settings (lighting conditions, background content, and occlusions), 3) performance constrains (continuous gestures, face/hand overlapping, hand out of scene, and pause during gesture) and 4) intra-class variance (gesture size, speed, location and orientation). Under different circumstances, the task of HGR can be fairly diversified. In the first half of this chapter (Section 2.1), different definitions of HPR and HGR under different contexts will be categorised. Since this thesis mainly focuses on appearance based HGR methods, the second half of this chapter (Section 2.2) is dedicated for review on the milestones and other related works in the category of 2D computer vision based HGR methods.

2.1 Categorisation of HGR Methods

To present a systematic review on the works produced by both academic and industrial communities, the existing HGR methods can be categorised according to their purposes and approaches. In Section 2.1.1, HGR methods are introduced under

different contexts. A categorisation of HGR methods based on the input sensors can be found in Section 2.1.2.

2.1.1 Categorisation by Context

People use hand gestures to convey various messages in communications. For different gesture vocabularies, HGR can be classified into two categories: *Communicative Hand Gesture Recognition* and *Manipulative Hand Gesture Recognition*. Communicative gestures mainly comprise sign languages. As standalone languages, sign languages are normally formed with 2500 - 3500 words [62]. From pattern recognition point of view, the massive size of vocabularies leads to a massive number of classes in the feature space, which is always a tough task for classification. Due to the presentation limitations of using only two hands, the similarities among different classes are relatively high. Moreover, to simulate specific semantic meanings of certain words, out-of-plane rotations are often involved in the gestures, (such as word "bicycle" in American Sign Language [21], shown in Fig 2.1). The circles are hard to be distinguished from vertical strokes with normal 2D cameras. There are three types of signs in sign languages [63]: 1) Word Signs: the signs are designed based on the semantic meaning of the words [64, 63, 65, 47], 2) Non-Manual Signs: they are signs with additional features other than hand movements and poses, such as head poses, tongue poses and other body postures [66, 67, 68], 3) Finger Spelling Signs: these signs require the sign performer to spell the words by drawing the numbers or characters from written languages in the air [69, 70, 71, 72]. Researchers are focusing on tackling the massive size of vocabularies for the communicative HGR methods, instead of the usability issues from the unconstrained environments.

For Manipulative Hand Gesture Recognition, the task is more about sending commands to the computers, rather than serving communicative purposes. Hence, the gestures are intuitive and simple, such as swipes for turning pages. From usability and user experience's point of view, gestures as commands need to be as simple

Figure 2.1: Word Bicycle in American Sign Language [5].

as possible. In that way, performing the gestures can be as less distracting for the user as possible. Also, the size of the vocabularies are relatively small. The similarities among gestures can be low. For real world applications, Manipulative HGR methods are used as interactive user interfaces. The potential market is beginning to emerge for the Manipulative HGR methods [16, 17, 18]. This thesis is focusing on Manipulative HGR, more specifically, recognising 10 hand-signed digits.

Hand Posture Recognition methods are also widely used in both Communicative Hand Gesture Recognition and Manipulative Hand Gesture Recognition. In Sign Language Recognition, many words are designed to have distinct hand shapes. Therefore, the hand posture is naturally one of the trajectory features. Sometimes the hand trajectories in different signs are the same, and only the hand poses are different (Fig 2.2). In these situations, detecting different hand poses becomes part of the recognition process. Also, for the finger spelling alphabet in sign languages [73, 74], the only task is Hand Posture Recognition. As for Manipulative HGR, most of the vocabularies in real-life applications are normally formed by static hand poses [75, 2, 76]. For HPR applications, the camera scope is focused on the hand region with relatively small background areas. That leaves the distractions in the background smaller proportion of the scene to present complex and moving textures. The challenges from uncontrolled environments faced by HPR methods are far less severe than HGR, which makes real-time response an easier task for HPR methods. HPR methods are also widely used for solving the Hand Gesture Spotting prob-

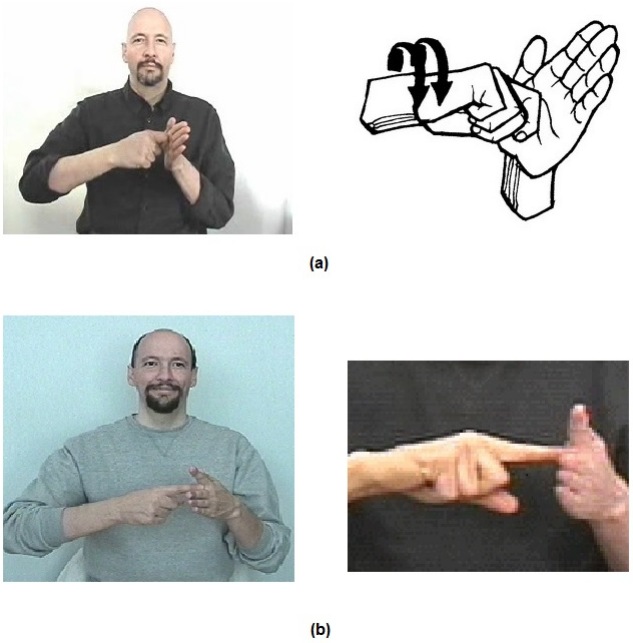

Figure 2.2: Example of signs that are only different on the hand poses. (a): word Key in the American Sign Language. (b): word start in the American Sign Language [5].

lem. The simplest solution for HGS is defining certain hand postures as the starting signal for the predefined gestures. This idea has been adopted in both academic researches [54, 77] and commercial systems [75, 16, 76, 78].

2.1.2 Categorisation by Sensor

As a typical pattern recognition problem, the performance of HPR and HGR methods highly depends on the quality of features. Due to the various applications of HPR and HGR methods, different input sensors can provide various imaging methods, such as depth information and infrared data. Hence the classifiers in different applications are facing different types of features. To review HPR and HGR methods, categorisation of methods based on input sensors is an vital perspective. In this section, methods with non-vision based sensors, 3D and 2D vision based methods will be introduced.

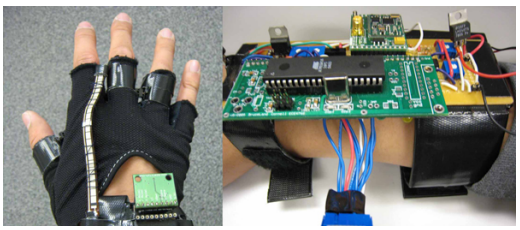

Figure 2.3: Mister Gloves, Cornell University [6].

2.1.2.1 Non-vision-based HGR

For commercialised HGR technologies, stable and robust sensors can provide solutions for challenges from the uncontrolled environments (see Section 1.3). Various types of sensors can produce additional information such as depths of pixels or hand articulation models. The methods using these sensors can identify target hand without being distracted by objects in the background, even frontal occlusions. Despite the advantages, comparing with normal 3D or 2D cameras, the cost of deployment or integration of complex sensors can be relatively high. But that did not stop the non-vision-based methods being commercialised in the recent years. This section provides a brief review on these methods.

Intuitively, glove is a natural form to capture hand gestures and postures. Glove based method is one of the earliest HGR ideas. Mister Gloves, developed by Cornell University [6], uses wireless USB transmission integrated on a pair of circuit gloves as the input method. By monitoring hand joints movements to detect different hand postures and transmitting wireless USB signal to report the hand position, Mister Gloves system can accurately recognise dynamic hand gestures and static hand postures. However the sizeable circuit gloves and hardware costs prevented this method from vast production.

Another leading technology among the glove based methods is CyberGlove, developed by CyberGlove Systems [79]. The sensor glove is attached with joint-angle measuring sensors and an array of pulse-vibration stimulators. It can not only capture the hand movements, but also simulate simple sensations onto the user's

hand. Namely, the user can "feel" the virtual objects through the glove.

Touch screen technologies has been widely used today for capturing dynamic hand trajectories. Due to the commercial value, the production process of the touch screens has been perfected in the past decade. The costs of the touch screens are low enough to make them standard embedded hardware for portable devices. Touch screen technology is considered the most reliable HGR method commercially available on the market.

MYO [17], is an armband sensor that monitors the user's electrical activity from the arm muscles. The armband can predict the hand gestures and postures through the patterns in the electrical signals. This product is extending the usability boundary of HGR technologies. It is portable, wearable, and it does not require any cameras.

2.1.2.2 3D Vision based HGR

Computer vision based methods that utilise RGB Depth (RGB-D) cameras as input sensors are becoming the main stream for tasks like 3D hand pose estimation and human pose estimation. With input feed of static images or video streams from the RGB-D sensors, 3D computer vision methods analyse the content of the images or video frames using image processing, pattern recognition, machine learning techniques, and recognise predefined hand trajectories or hand poses.

3D depth information is widely used for hand pose estimation [80, 81, 82]. With the depth information, these methods are able to distinguish the foreground hand from the background textures as they are in different depth scales. The challenge of cluttered background is overcame from the sensor level, which gives the methods more focus on challenges such as high intra-class variance. For applications where the only available sensor is one 2D camera and the environment is containing cluttered background, a method is proposed in Chap 3 in this thesis for this specific task. Despite the different sensors, methodologies of the state-of-the-art

methods using RGB-D sensors need to be reviewed here. Keskin et al. [80] proposed a method for 3D hand pose estimation which firstly clustering the training set, then training multiple experts using the method introduced in [83] for every cluster. However, with given depth information, the spatial features used in [80] for the shape classification trees are neither rotation or scale invariant. Similar spatial features are used in other RGB-D methods [83, 84]. That motivated the idea of using rotation and scale invariant texture key points to describe the hand poses, which will be introduced in Chap 3. Melax et al. [81] introduced a tracking method for human hand poses in markerless depth sensor data based on physical simulation. They used a strong prior called Rigid Body Dynamics which is a set of assumptions and constraints that simulates the articulation and interaction of human body parts. This prior largely lowers the computational cost of the tracking method. But for human computer interaction in uncontrolled environments where no assumptions should be made on the articulation or manner of interaction of the human body parts, the method of [81] would not be able to track human body parts with different articulations (such as face tracking). That motivates the method proposed in Chap 3 which does not make any assumptions on articulation of the body parts. It only extracts and learns the effective texture key points of the body parts, in the experiments of this thesis, human hands. Ballan et al. [82] used discriminatively learned salient points on the fingers for two hands interaction motion capture. They also proposed a differentiable objective function that can take optical flow and edge information into account. But the method of [82] requires additional computation for collision and self-occlusion detection. For the method proposed in Chap 3, the possibility of discard the additional compution on collision/occlusion detection is explored. By learning the most effective texture key points in the collision/occlusion free training set, the occluded texture key points in the testing samples will simply produce low matching scores, no different than texture key points of the cluttered background. Also, the method of [82] assumes the input frames are spatially calibrated, while

the method of Chap 3 does not require any pre-calibration.

As for depth information based hand gesture recognition, Peng et al. [85] proposed a method to train linear Support Vector Machine for action and hand gesture recognition in depth video based on a super vector representation of dense features. This method uses a sliding window scheme with trained isolated gesture models for gesture spotting. However, the method is vulnerable to multiple distractions including people other than the gesture performer. That inspires the spotting method introduced in Chap 6 uses similar sliding window scheme with additional analysis on different hand candidates to deal with the challenge of other people in the scene. Pei et al. [86] introduced a method for hand gesture recognition in depth video based on a set of heterogeneous attributes learned from the depth data. They use SVM-HOG detector and tracklet generation to detect and track multiple gesture performers in the scene. Similar to [85], method of [86] is vulnerable to situation of additional people in the scene. Bagdanov et al. [87] proposed a method for predict the status of the gesture performer's hand in real-time by using temporal filtering. This method detects the hand region with pre-defined hand poses as the target hand. For presence of multiple hand candidates with similar hand poses, the method would not be able to do target hand segmentation. That motivates the method introduced in Chap 4 and 5 to explore the possibility of tracking and analysing all hand candidates in the scene. Other methods use silhouette spatial features [88, 89, 90] with depth information to segment and track the target hand. However for uncontrolled environments where occlusion is allowed, these methods would not be able to extract accuate silhouette features, which motivates the tracking method introduced in Chap 4 to tolerate temporal occlusions.

2.1.2.3 2D Vision based HGR

This thesis focuses on 2D vision based HGR, namely appearance based HGR methods. The only sensor used in the methods is a normal 2D camera, without the ability

to capture depth information. The reason for this choice is based on the current usability issues with the RGB-D sensors. Currently, RGB-D sensors are not as widely deployed in portable personal electronic devices (laptop, mobile and tablet) as 2D cameras due to the size and extra hardware costs of the RGB-D sensors, despite the fact that the sizes of the RGB-D sensors are getting smaller and the manufacturing cost difference between the RGB-D sensors and the normal 2D cameras is also getting smaller. In other words, from the point of view of real-world application usability, the general population user base of RGB-D sensors is still in a relatively smaller scale than normal 2D cameras. Moreover, no matter how small, the size and price differences of RGB-D sensors and normal 2D cameras remain the major obstacles of the adoption of RGB-D sensors in main stream consumer electronics. Although The ability of utilising depth information of the RGB-D sensors is a major advantage over the 2D cameras, the primary motivation of this thesis is to explore the possibility of performing robust HGR with the most widely deployed camera type on the market which currently is the 2D cameras. The appearance based HGR methods normally have 3 main steps, hand segmentation/tracking, feature extraction and gesture classification. The following section provides a detailed review on various techniques used in appearance based HGR.

2.2 Methodologies of Appearance based HGR

In this section, a review on appearance based HGR methods is presented. The review follows the three basic steps of the HGR process. For each basic step, widely used classic techniques and the state-off-the-art methods will be introduced. Brief analysis on the merits and drawbacks of the techniques can also be found in this section.

2.2.1 Hand Segmentation and Tracking

To record the hand trajectory for HGR or extract the hand area for HPR, the target hand area needs to be segmented from the background and tracked throughout the video stream. In this thesis, to distinguish the features for Hand Segmentation and Tracking from the features for Gesture Classification, the features for the former are called *spacial tracking features*, while the features for the latter are called *temporal trajectory features*. The most distinctive characteristic of the target hand region against the background is the skin colour. Hence, skin colour detection is widely used in HGR community, which is also one of the active research fields in the computer vision community [91, 92, 93, 94, 95, 96] . Human skin colour has a relatively distinct distribution in the colour spaces [94]. The skin colour tones from different races share similar hue value, but the saturations are different [97], especially in the HSV colour space. That makes the HSV colour space the most widely used colour space to perform skin detection [98, 99]. It is obvious that a model of skin colour can be trained to perform skin detection. Jones et al. [91] trained a Mixture of Gaussian Model through a large database of skin and non-skin images. The detection time is unacceptable for time-sensitive applications. There are also some methods focus on other colour spaces. Li et al. [100], proposed a Gaussian model for skin detection in the YCbCr colour space. Unsupervised learning methods are also used in skin detection. Kovac et al. [93] proposed a clustering method for skin detection to overcome the changing illumination. For real-time applications, the common choice of skin detection method is simple thresholding with carefully chosen thresholds.

For HGR methods that only consider the controlled background without any moving distractions, the target hand can be easily segmented by using scene depth information. Some HPR methods adopt the depth information to facilitate the hand segmentation through stereo cameras. Kim et al. [101] proposed a Latent Tree Model that is capable of presenting the hierarchical topology of the hand postures, with depth information. Khamis et al. [102] proposed a method for learning a

compact and efficient model of the surface deformation of human hands from depth information.

For HPR methods, there is no need to perform hand tracking, given that the samples are normally still images. For HGR methods, after detecting the target hand, the hand region needs to be tracked. Hand tracking is no different than other standard tracking tasks such as object tracking. Hence, various traditional tracking methods have been utilised for hand tracking. Mean Shift as one of the traditional methods, is firstly proposed by Fukunaga et al. [103] for feature space analysis. Mean Shift has been used for tracking in many works [104, 105]. The basic idea is firstly to train a model of the target represented by texture, contour or colour features. Then similar to the optimisation methods, Mean Shift takes a density function of the target region model and searches for the optimised matching area in the frames. Continuously Adaptive Mean Shift (Camshift) [106] is a variation of Mean Shift which changes the window size in Mean Shift when the search is near the convergence point. It also has been used for hand tracking [107, 108, 109].

Particle filter methods are essentially grid-based iterative search methods. With the extracted features from the given target region, the methods search for similar regions in the whole frame iteratively. In each iteration, the search biases to the matching result from the last iteration, until the search converges at a region. Particle filter methods are widely used for tracking [110, 111, 112]. Shan et al. [113] proposed a hand tracker that combining the Particle filter with Mean Shift. Another well-known hand tracker proposed by Stenger et al. [114], trains a dynamic model to guide the particle filter search. Stenger et al. [115] proposed a method using hierarchical bayesian filter for model-based hand tracking.

Optical flow [116, 117], is a velocity field that shows the movement of pixels. Optical flow can be used to extract trajectories of pixels in video streams. However, it works under two assumptions. The first one is that the target feature point has constant texture and brightness during the motion. The second one is that all pixels

within the neighbouring region of the target feature point are moving towards the same direction as the target point. Rehg et al. [118], proposed a optical flow hand tracking method that can recover the full hand articulation model with 27 degree of freedom from the gray level images. Lu et al. [119] proposed a fusion model that combines optical flow with other features to perform hand tracking.

2.2.2 Feature Extraction

For Hand Posture Recognition, the feature for posture classification is the data representation of the hand shapes. There are two main types of features for HPR, contour and texture features. Contour features are descriptors that represent the exterior contour of the target hand region. The Fourier descriptor is one of the most widely used contour features. The basic idea is making the contour an one dimensional vector, similar to the chain code. Then Fourier Transform is applied to the vector. The Fourier coefficients can then be treated as the feature of the contour [120, 121]. Moments are descriptors that represent various contour properties, such as the sum of horizontal and vertical directed variance. Other contour features include Wavelet descriptors [122], shape signatures such as average distance between pixels on the contour, gradient shape features [123] and the centroid of the contour, etc.

Texture features are essentially various representations on the gradient patterns. Histogram of Gradient (HoG) [124] as one of the popular texture features, is a histogram presentation of the local gradients. Scale-Invariant Feature Transform (SIFT) [125] is another popular texture feature. SIFT contains selected key points with coordinates and gradient descriptors. The key points are local extreme values in the Laplace of Gaussian (LoG) pyramid. The descriptors summarise the orientation and intensity of local gradients. The two important advantages of SIFT are its invariance against rotation and scale. Hence, the SIFT key points represent the edges and ridge-like textures regardless of the orientation and scale of the target

object. Moreover, it can also tolerant certain level of view point changing. Speeded-Up Robust Features (SURF) was originally presented by Herbert Bay et al [126] based on the idea of SIFT. It also inherited in-plane rotation and scale invariance, which makes it desirable for Hand Posture Recognition. However, with high precision of texture matching, the 64 dimensional descriptor of SURF requires intense computations for both key point extraction and matching. Calonder et al. [127] proved that this problem can be fixed by directly building a short binary descriptor with independent bits. That is called the Binary Robust Independent Elementary Feature (BRIEF) [127]. This binary descriptor uses Hamming distance as matching criteria, instead of Euclidean distance between the descriptors. That can largely fasten the texture matching process. However, the descriptors are not rotation and scale invariant. Rublee et al. [128] proposed an improved binary descriptor which is rotation invariant, called Oriented Fast and Rotated BRIEF (ORB). There is an additional advantage of ORB which is its robustness to noises. Leutenegger et al. [129] also introduced a binary descriptor that is both rotation and scale invariant. It is called Binary Robust Invariant Scalable Keypoints (BRISK). The main characteristic of BRISK is that, in each scale pyramid octave, a corner detection method called Features from Accelerated Segment Test (FAST) [130], is used to detect the key points, instead of simply locating local extreme values as in SURF and SIFT. That makes the key point selection more efficient in BRISK, and ensures that the amount of key points in BRISK descriptor is lower than SIFT and SURF. However, the robustness of BRISK descriptor can be affected by out-of-plane rotation or rapid texture changes. That is a vital drawback for applications such as HPR. Fast Retina Keypoint (FREAK) proposed by Alahi et al. [131] is the latest development on gradient based key point texture features. It simulates the principle of human retina visualisation. A cascade of binary descriptors is used instead of a single descriptor. The cascade structure is constructed by comparing image intensities over a sampling pattern based on the human retinal ganglion cells distribution, in

which the method picks more key points on the central area. Although Alahi et al. [131] reported better performance over SURF, its robustness against illumination and view point changes in HPR applications still remains untested. For the tracking method in Chap 4, the only texture feature used is SURF. The reason is two-fold. Firstly, using short binary descriptors including BRISK and ORB sacrifices certain invariance properties and accuracies [127, 128]. Secondly, SURF has certain level of tolerance for view-point variance, which BRISK and ORB do not have.

For HGR applications, the aforementioned gradient based texture key point descriptors can also be used for texture matching based tracking. In Chapter 4, a novel texture matching tracker will be introduced. Instead of shape descriptors, descriptors of dynamic hand trajectories are the features for gesture classification in HGR problems. Since for HGR in uncontrolled environments, the method should not require certain hand shape to be presented by the user, the spatial features of hand shapes are not considered in the HGR methods introduced in Chap 4-6. After the hand tracking method has located the position of the hand candidate in the frames, temporal features are extracted to represent the trajectories. Unlike contour and texture spatial features, there are only a few commonly used trajectory features. They can be categorised into two types, local and global features. Local trajectory features include hand speed, location and movement direction [8, 3]. The hand velocity, orientation of movement and coordinates displacements of the hand between adjacent frames can be used to describe the elementary trajectory segments. Global features are shape descriptors that are extracted from the complete hand trajectories. All contour and texture features mentioned before can potentially be used as global trajectory features. Because the same as hand postures, trajectories can be seen as stationary images.

2.2.3 Gesture Classification

All classifiers are trying to summarised feature patterns from the training set, and then applying these patterns to classify the testing samples. For HGR, the hand trajectories are sequence data represented by both spatial and temporal features. The classifiers must be capable of processing sequential features in order to classify the hand trajectories. But for HPR, the classifiers does not have to process sequence data since the samples are still images without any temporal information. In this section, a review on some of the popular classifiers for HGR and HPR is presented.

Template matching is a strategy for directly measuring the distance between the testing sample and the predefined gesture class models. The advantages of this strategy are twofold. Firstly, minimum training is required. Since the methods directly calculate the distance of two feature vectors in the feature space, instead of extracting latent patterns or measuring elementary local patterns, the templates of gesture classes are usually pre-processed feature vectors of training samples. Hence, there is no need to build statistical models for the gesture classes through the training process. Secondly, the inference time cost is usually low. Namely, the calculation of the feature vector distances does not require complex computation. Although the computational complexity depends on the dimensionality of the feature vectors, template matching methods are still considered less computational intensive than statistical models. Continuous Dynamic Programming (CDP) as one of the popular template matching methods, is proposed by Nishimura et al. [132] for segmenting and recognising continuous hand gestures. A set of sequence patterns are used to represent trajectories in the spatio-temporal space. A dynamic programming based method is used to match the sequence patterns, which accumulatively adds the distances between corresponding elements in the sequence patterns. Decent results are reported on a 8 hand gesture database [132]. Alon et al. proposed a hand gesture segmentation scheme based on CDP [133]. A pruning method is introduced in this scheme to discard hand trajectories with relatively short length. An improved

version with template matching method based on Dynamic Time Warping (DTW) is proposed later [8]. This work introduces the concept of sub-gesture reasoning, which learns the relationships among the gesture classes. For Hand Gesture Spotting, sub-gesture reasoning can improve the ability of segmenting similar gestures. However, these two methods require extra computation on estimating the location and scale of the gestures. In other words, the methods of [133, 8] do not have gesture scale and location invariance property. The DTW classifier itself still requires estimated scale and location of the gesture trajectories. A pruning technique is also used in the method to reduce the amount of hypotheses. DTW based methods are normally used for tackling temporal element displacements in the templates. If the sequential order of the elements has a certain level of variance in the testing set, DTW based methods can overcome the variance by matching the elements within a corresponding time window without considering the order of the elements. However, using the length of the trajectories as a criteria for pruning means the method is not gesture speed invariant. For the testing samples where the gesture performer signs the gestures relatively faster than the performers in the training samples, there is a high probability that the method of [8] would prune off these testing samples regardless of the actual gestures labels of the samples. Hense for the uncontrolled environments, the methods of [133, 8] are sensitive to the various scale, speed and location of the gestures. This motivates the methods in Chapter 4,5 and 6 to propose gesture recognition and spotting methods that are invariant to gesture scale, speed and location.

Statistical models are also widely used for HGR. Hidden Markov Models (HMM) [134] is one of the early probabilistic models proposed for pattern recognition applications. It has been successfully applied to HGR problems [135, 136, 137, 138]. HMM based methods have to obey the independence assumption, which is assuming there is no dependencies among the observation states within the input observation sequence. In other words, this assumption is essentially discarding the long range

temporal features in the hand trajectories. The purpose of this assumption is to keep the calculations of training and inference tractable. As a complementary measure, HMM based methods use a set of transitional probabilities to simulate the local dependencies between the adjacent observation states. The model needs to estimate the state and transitional probability matrices among the observations and hidden states in the training set. Then the inference task is performed through forward-backward propagation based on the trained probability matrices. Starner et al. [135, 138] introduced two HMM based models to perform HGR on a 40 words sign language vocabulary, and reported decent accuracies. A few HMM variations are proposed for different specific applications. Elmezain et al. [139] trained a dedicated HMM model for each gesture class, with various number of hidden states. Brand et al. [140] introduced a coupled HMM method for classifying two-handed signs. This method is proven to be robust against initial observation probability changes. Wilson et al. [141] proposed a parametric HMM, which extends the original HMM by including a global parametric variation on the output probabilities of the hidden states. But HMM is only able to take the last observation state into account for infering the current hidden state. That means HMM is incapable of monitoring long-range dependencies within the observation sequences. In the context of HGR, the transition probabilities in HMM can only represent trajectory temporal features within adjacent frames. The transition probabilities are not considered under the context of the entire trajectory. That inspires the method in Chap 5 to model the long-range dependencies in the observation sequences.

Another popular concept in pattern recognition is Deep Learning. One of the main concepts of Deep Learning is to train the features instead of using man-made features. Dan et al. [142] showed the potential of Deep Learning methods in solving various computer vision problems. Convolutional Neural Networks (CNN) proposed by Lecun et al. [143], is one of the best examples of Deep Learning methods. The key innovation of CNN is choosing trainable features over heuristic

features. It breaks the conventional concept of man made spatial features. In the CNN model, a set of fix-sized trainable kernels are defined to extract local texture features. The training process is based on optimisation methods with an error function on the whole training set as the objective function. Hence, the training process is essentially searching for optimised kernels that can minimise the error function of the training set. The kernels act like texture feature extractors. In each convolution layer of the neural network, the input image will be convolved with the trained kernels for feature detection. Every convolution layer is followed by a pooling layer which downsamples the input image to one-fourth of the size. As the input image goes deeper into the network, the kernels with fixed size are monitoring texture features on larger scales. The final output layer of the network produces the final scores based on the input signals from all previous layers. Simard et al. [144] simplified the CNN model. The simplified version does not require weight decay and averaging layers. Multi-column Deep Neural Networks (MCDNN) is introduced by Ciresan et al. [145], as a CNN structure with a large number of feature maps in each convolution layer, and large number of convolution-pooling layer pairs. In other words, this is a wider and deeper neural network. It is proven in this work that with more feature maps to monitor a large number of local texture features, the MCDNN is capable of producing state-of-the-art performance on various computer vision applications. The methods in Chap 5 and 6 are monitoring the temporal features on different scales similar with deep learning methods. But the proposed methods in this thesis are not capable of generating features on different scales in the training stage.

Conditional Random Fields (CRF) [146] is invented for segmenting and labelling sequence data [147]. It is proposed to tackle the drawbacks of independence assumption and Label Bias Problem in the generative models such as HMM. CRF models utilise the concept of factorisation to relax the independence assumption. A series of elementary feature functions, namely the "factors", are defined to detect

specific local observation patterns regardless of their positions within the observation sequence. In other words, these feature functions are temporal independent. For each feature function, the CRF model assigns a trained weight to balance the voting power on all feature functions. The training process of the weights can be understood as a search for the optimised weight distribution for all feature functions. The search is carried out by gradient based optimisation methods, with the likelihood function of the training set as the objective function. Also a penalty term is included in the likelihood function to reduce the influence of overfitting. The likelihood function in the original CRF model is guaranteed to be convex, since it is the summation of a linear function and a known convex function. Hence, there is a global optimisation point for all pattern recognition problems that use CRF models. Although CRF is able to monitor the long-range dependencies in the observation sequences, it is unable to learn the underlining structures of the vocabulary with trained latent variables as HMM does. Hence, Hidden Conditional Random Fields (HCRF) [148], is proposed to extend the original CRF with a set of latent hidden states, similar to the hidden states in HMM. The hidden states are used to simulate the inner-structure of the observation sequences. Each hidden state can be seen as a "component" of the predefined classes. HCRF model has been used to model the underlying structure of similarities among strokes of hand gestures for gesture recognition in controlled environments [148]. The main drawback of the HCRF model is that the likelihood function is no longer convex. With the latent hidden states, the training process is not guaranteed to reach the global maxima. But in different applications, it has been proven that the loss of convexity does not affect the over-all model performance definitively [149]. In the training process of the HCRF model, the learning process of the hidden states is semi-supervised. Although the class label of the training samples are provided, but there are no label of hidden state on all observation states. That leads to the invention of the Latent Dynamic Conditional Random Fields (LDCRF) [150], in which the training of the hidden states are fully

supervised. In this way, the model can learn the underlying structure in the hand gesture observation sequence in supervised manner. The CRF and its variants have a close relationship with Convolutional Neural Network and its variants. Both series of classifiers are discriminative models (more details about discriminative and generative models can be found in Section 5.1.2). The CRF models can be treated as a single layer Neural Network. In other words, Convolutional Neural Networks are also capable of monitoring the local feature patterns hierarchically. The classifier of the proposed HGR framework in this thesis is built upon the HCRF structure. More detailed discussions on CRF principles and how CRF models tackling the Label Bias Problem under the context of HGR can be found in Chapter 5. For gesture recognition and spotting in uncontrolled environments, the class of CRF related models are not considered before. That motivates the methods in Chapter 5 and 6 to build a novel weighting scheme with HCRF as the initial classifier.

Some recent publications proposed promising methods for activity recognition and prediction in uncontrolled environments. Yu et al. [151] proposed a novel hough-transform based voting method which uses random projection trees to perform feature voting. This method reported taking about 10 seconds to perform the activity classification for a video with four seconds length. It is far from real time, but the method is robust against crowded scenes. Ryoo et al. [152] proposed a forward human action prediction method which represents an activity as an integral histogram of spatio-temporal features. A novel recognition method called dynamic bag-of-words is also proposed in this work, which is capable of taking the sequential nature of human activities into account while maintaining the merit of noise tolerance of the bag-of-words method. This work can perform prediction in real-time given that the features are fed to the method in real-time. Gall et al. [153] proposed Hough Forests for human activon recognition which are random forests variations utilised to perform a generalized Hough transform. They reported 10 seconds time cost for classifying pre-existing actions in 100 frames. The method in Chapter 6

introduced a forward spotting scheme which also uses a sliding window mechanism as [152].

Chapter 3

Hand Posture Recognition

Hand Posture Recognition is the task of recognising predefined stationary hand poses from still images. That means no sequential temporal information is involved in recognising the hand poses. Despite the simpler task compared with HGR, HPR in complex backgrounds still remains a challenging problem. Existing RGB-D HPR methods are using depth information to segment the target hand in the scene, while the appearance-based methods are struggling with cluttered backgrounds. In this chapter, a novel method is proposed for performing HPR in real-world applications where depth information is not available and the background is cluttered. This HPR method introduces the idea of using boosting-based method to select a optimised set of rotation and scale invariant texture features in the training stage, which makes the method capable of recognising the pre-defined postures with more distinctive featuers against the cluttered background.

The proposed method is a combination of Speeded-Up Robust Features (SURF) [126] and Adaptive Boosting (AdaBoost) [154]. Firstly, SURF key points are extracted to describe the blob or ridge-like structures from grey level hand posture images. These SURF key points are potential points of interest that can be used to match with other images with similar textures. Then the tendency of gradient changes within small patches surrounding the points of interest are calculated

as feature vectors. With all the points of interest, a boosting based method is used to train a strong classifier for each posture by selecting and combining the most efficient features. This boosting process can significantly reduce the computational cost of inference. The proposed method was tested on the Hand Posture Recognition Benchmark, the Triesch Hand Posture Database. Experimental results showed that our method outperforms existing methods in terms of better recognition accuracy.

3.1 Background Knowledge

Brief discussions on the advantages of SURF and AdaBoost and the reason for utilising them are presented in this section.

3.1.1 Texture Features

Various features have been tested in previous HPR works, including colour, contour and texture. In this method, texture features are used to achieve skin colour invariance and performer independence. The varying skin colours and hand appearances of different individuals are two crucial factors that cause large intra-class variance in HPR. Different individuals tend to have unique hand joint articulation patterns. That means for the same hand pose, the actual contour and texture on different hands may vary. Using colour or contour features can blur the decision boundary among classes in the feature space. Hence, only texture features are used in this method, which means only distinct patterns of gradient changes in hand regions are used to recognise hand postures. Intuitively, different people would have unique ways of performing the same hand pose. But, the basic articulation of joints are the same. If we take the "fist" hand pose for example, regardless of the various appearances of thumbs and divergent positions of the thumbs in the hand regions, people are required to cross the thumb horizontally with other four fingers in the front. Hence, for all "fist" images, there are certain amount of common edges and

corners located in the intersection of fingers or on the external contours. Hence, the proposed method extracts texture key points with extreme values of gradient within local areas, and selects texture features that are "common" among the samples of the same hand pose.

Speeded-Up Robust Features [126] is used in our method as the texture feature. It has in-plane rotation and scale invariance properties and a certain level of tolerance for view point and illumination changes, which makes it desirable for HPR. SURF was originally presented by Bay et al [126] and built upon the idea of Scale-Invariant Feature Transform (SIFT) [125]. The interest points are detected at first from the scale space using Hessian matrix approximations with the approximate second order Gaussian derivatives.

SURF uses filters of various sizes to build a scale space, instead of using the same filter iteratively as in SIFT. A method called 'integral image' was used for fast convolution. In an integral image, the value of pixel x is the sum of pixels in the original image within a rectangular region from the top-left corner to x. With the approximate second order derivatives of the Gaussian filter and the calculated integral image, convolution with Gaussian filters of various sizes can be done at the same speed. With the integral image, the summation of intensities inside a rectangular area of any sizes only requires three addition operations and four accesses to the memory.

The candidate points are then picked out. These points have the extreme determinant value of the Hessian Matrix within a 27 pixels neighbourhood. This neighbourhood comprises 3×3 pixels in each of the upper, lower and current scale levels. The interest points are located using scale space interpolation. For every interest points, the Haar wavelet response in both x and y direction within a circular neighbourhood with $6s$ radius of interest points are calculated, where s stands for the scale of the level where this key point is found. Then within a sliding orientation window covering an angle range of $\pi/3$, the orientation with the largest sum of

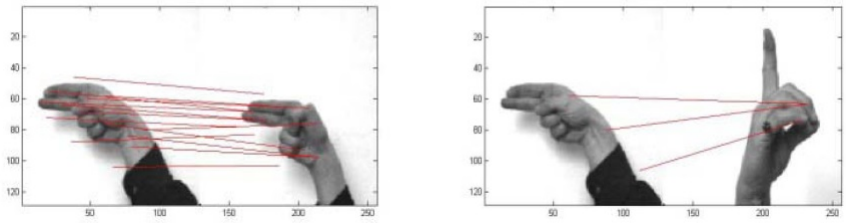

Figure 3.1: Matched SURF pairs in different postures.

wavelet response is selected as the dominant orientation. At last for each interest point, Haar wavelet responses within a $20s \times 20s$ surrounding region are used to form the final descriptor. The final feature vector of SURF is 64 dimensional, which is half the size of the SIFT descriptor. Fig 3.3 shows that with similar blob or ridge structures, despite the presence of the sleeve or ring on the finger as noise, the same posture from different performers (left-hand side of Fig 3.1) still have more matched interest points than different postures from the same performer (right-hand side of Fig 3.1).

There are good reasons why we chose SURF as feature over SIFT and other texture features. Firstly, the calculation of SIFT is rather time consuming. Since SURF uses integral images and LoG approximations, the process of building the scale space is significantly accelerated. This process can even be paralleled. This makes the computational cost of SURF relatively less dependent on the image size than SIFT. Secondly, most of other features used in HPR are based on binary images with enhanced hand regions. However, they require hand segmentation, while SURF does not require hand segmentation since it is based on gray scale images. Thirdly, SURF depends on gradient information on sub-patches, instead of individual gradients, which makes SURF less sensitive to noises. SURF not only uses the sum of wavelet responses, but also the sum of absolute values of them. As such the descriptor can also indicate the number of changes across the patch. For the region on the left, without dramatic intensity changes, the sum of wavelet response and the absolute values of them do not have much differences. For the

region on the right, which has more intensity changes in the x direction, the sum of the wavelet response will not change much, which fails to show the intensity changing information within the region. On the other hand, the sum of absolute values of wavelet response is high, which can preserve important information for patch-matching process. Hence for every sub-patches, a 4 dimensional descriptor $v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$ is built as part of the final feature vector. The final SURF descriptor comprises the 4 dimensional descriptors from all 16 sub-patches, which makes the final descriptor 64 dimensional. Certain parts of hand postures, for example thumbs in the fist posture images, can have rather large orientation variance among samples in the same class. SURF can provide robust orientation and scale invariance of the same texture pattern. The most important merit of the proposed method is that it handles the complex background noises without any additional training images with random background. For those methods that use colour information, skin coloured regions in the background can largely affect the performance. But for the proposed method, the gradient of intensity changes is used which has much more tolerance on distractions in the background.

In [2], they tackled the background noise problem by using images with single-coloured background as positive samples. Our method solves this in an even simpler and faster manner. With the boosting based feature selector, the training set is filled with images of complex backgrounds. Since all positive and negative training samples have complex backgrounds, the selected interest points are the ones that can best distinguish the target posture from other postures and background textures. Hence the selected patches are the ones appear in positive samples with high frequency and almost never appear in other postures and random backgrounds. Experimental results showed that this method can achieve high accuracy.

3.1.2 AdaBoost

Feature selection methods that are capable of screening out correlated redundant features are crucial for pattern recognition applications. If the classifier receives a large number of features, the dimension of the feature space would be high. That would make the computation more inefficient. For example, classifiers that perform optimisation on an objective function normally use gradient based search. The extent of mathematical tractability of the gradient based search largely depends on the dimensionality of the feature space. For a feature space with thousands of dimensions, even reaching a local optimum could take hours.

For applications with noisy training and testing samples, it is not fully justifiable to make a set of heuristics to extract and select features. The prior knowledge about which features are the most efficient for a particular dataset is not available. Especially for texture features such as HoG, SIFT and Gabor features, it is difficult to tell what gradient patterns are the distinctive characteristics for every class against other classes, and most importantly, against the unfamiliar random noise patterns. Intuitively, it makes sense to pick a set of features through a trial and error iteration process on a validation set in the training process. For images with rich textures, the amount of SURF key points would be high, since SURF key points are local extreme values in a three-layer subset in the LoG pyramid. Moreover the level of noise in testing samples are fairly high (Fig 3.5). Hence, a small number of distinctive features would lower the chance of mis-matching with the noisy backgrounds. Due to all aforementioned reasons, Adaptive Boosting is used in the proposed method as the feature selection method.

Boosting methods essentially test the effectiveness of various combinations of features on the validation set, rather than improving the distinctiveness of the features themselves. It is worth mentioning that feature selection does not necessarily mean dimensionality reduction (such as LDA, PCA, etc). The point of feature selection is not using eigenvectors to present higher dimensional feature vectors. In-

Figure 3.2: Testing samples in the Triesch Hand Posture Database. The level of noise in the background is high [7].

stead, we can simply build a set of evaluation rules, to test the effectiveness of all features, and prune off the features that are correlated or do not have high degree of distinctiveness among classes. Obviously, the key question is how to choose the set of evaluation rules. For boosting methods normally the evaluation process is minimising the error function through iterations of classification on the validation set. For AdaBoost, the concept is to assume that certain weak classifiers can produce accurate classification results among a certain set of samples. Then we can combine weak classifiers with complementary "specialities" into a strong classifier.

3.2 Methodology

Regardless of the relatively small 64 dimensional feature vector of SURF, even images of size as small as 128 by 128 pixels, dozens of interest points can still be detected. Boosting based methods can be used to find a subset of points of interests. The basic idea is to individually evaluate all SURF key points through an iterative process. In each iteration, a SURF key point with the lowest error rate would be picked as one of the weak classifiers. The process goes on until at least one of the stop criteria is met.

In this method, we treat the 64 dimensional descriptor of a SURF key point as a weak classifier. The feature selection mechanism of the proposed method is described in details in Algorithm 1. Assuming there are X images and Y postures in the training set, and let T be the maximum number of weak classifiers for one posture. The rationales behind the manual setting of the maximum number of the weak hypotheses are threefold. Firstly, since the number of weak classifiers in the strong classifier represents the precision of the classifier. Restricting the convergence criteria to a smaller value would requires larger amount of weak classifiers to produce higher precision. If we do not restrict the amount of weak classifiers, and the convergence criteria is too strict, or the quality of weak classifiers are too poor

to converge, it could take all weak classifiers to form the strong classifier. Then there is no point calling this a "feature selection scheme" if we select all features. Hence we need to stop the iteration at a reasonable point. Secondly, the training computational cost is proportional to the number of weak classifiers. We need to balance the trade-off between precision and tractability. Finally, if the objective function is converged to some extent, the selected weak classifiers are no longer generally effective. In other words, they over-fit to the training set. Even the testing samples can be highly distracted by the background textures, the diversity of the noise textures are still confined by the fact that we can only produce limited sample diversity in the database. The trained sets of weak classifiers could be only fitting the noise distribution in the database. Hence, we need to avoid overfitting by constraining the convergence of the objective function, therefore the number of weak classifiers. There is an optimal number of weak classifiers, with the combination of all features that are complementary in terms of distinctiveness among the classes. Adding more features would be either redundant (adding correlated features) or decreasing the precision of the strong classifier (adding noisy SURF key points). The actual number of selected weak classifiers for every target posture could be different (Fig 3.7).

For a certain class label, the SURF features are extracted from all positive samples in the training set. Different weights $w_{t,x}$ will be assigned to all samples in the training set, where t is the number of iteration in boosting process and x is the sample index. All positive samples share the same initial weight w_p:

$$w_p = \frac{1}{2N_p} \tag{3.1}$$

and all the negative samples share the same initial weight w_n:

$$w_n = \frac{1}{2N_n} \tag{3.2}$$

where N_p is the total number of positive samples of this class label and N_n is the number of negative samples. For every target posture, the SURF vectors of all positive training samples will be put into the weak classifier pool. All SURF vectors in the pool will be tested to label all samples in the training set. To evaluate the performance of each SURF vector, an error rate will be calculated for every vector using the weight of all training samples.

$$e_t = \sum_{x=1}^{X} w_{t,x} \left| h_t \left(x, f, \theta \right) - P_x \right| \tag{3.3}$$

where the $h_t\,()$ represents the weak classifiers, x is the training sample index, f is one SURF key point and θ is the corresponding threshold of f. P_x is the ground truth class label of the sample. One SURF vector with the lowest error rate will be chosen as one of the weak classifiers that form the strong classifier of this posture. For each chosen weak classifier, a final weight α_t will be calculated based on its error rate:

$$\alpha_t = \log \frac{1 - e_t}{e_t} \tag{3.4}$$

Then the weights of all training samples will be updated based on the error rates of the chosen SURF vectors. The weights of the correctly classified samples will be reduced by a factor:

$$w_{t+1,x} = w_{t,x} \cdot \left(\frac{e_t}{1 - e_t} \right) \tag{3.5}$$

For the misclassified samples, the weights will stay the same. The process iterates until the error rate of the latest chosen vector is smaller than a threshold or the number of selected vectors reaches the predefined limit T. In every iteration, since the misclassified samples have larger weights, the process will look for the next weak classifier which can specifically classify these samples. This process has a fairly

intuitive pattern. Every chosen SURF vector is specifically suitable for classifying samples with certain characteristics, which cannot be efficiently classified by other chosen vectors. Every weak classifier consists of a weight, a selected SURF feature and its threshold.

Algorithm 1 Training process for all predefined posture classes.

Input:

The training set consists of X samples of Y classes: $(I_1, P_1) \ldots (I_X, P_X)$, where I_x is the x^{th} training sample, P_x is the posture class label of sample I_x.

Output:

Strong classifiers for all posture classes, $H = \{H_1, H_2, \ldots, H_Y\}$.

Assign weights $w_p = 1/2N_p$ to all positive samples, $w_n = 1/2N_n$ to all negative samples, where N_p and N_n are total number of positive and negative samples respectively;
for $t = 1, \ldots, T$ **do**
  Initialise the error rate $e_t = 1$;
  while $e_t \leq ERT$, (ERT is the Error Rate Threshold) **do**
    Normalise weight of all training samples, so that
    $\sum_{x=1}^{X} w_{t,x} = 1$
    Select one feature f_t with its threshold θ_t, from the SURF key points of all positive samples, which minimise the error rate: $e_t = \sum_{x=1}^{X} w_{t,x} \left| h_t\left(I_x, f_t, \theta_t\right) - P_x \right|$
    Assign weight α_t to the selected weak classifier $h_t\left(I_x, f_t, \theta_t\right)$,$\alpha_t = \log \frac{1-e_t}{e_t}$
    Put $h_t\left(I_x, f_t, \theta_t, \alpha_t\right)$ into H_y.
    Update the weights of all training samples:$w_{t+1,x} = w_{t,x}\left(\frac{e_t}{1-e_t}\right)^{1-\lambda}$, where $\lambda = 0$ if sample is correctly classified, $\lambda = 1$ otherwise.
  end while
end for
return H;

In the process of selecting weak classifiers, for every chosen vector, an optimised matching threshold will be evaluated. The matching threshold is the Euclidean distance between the selected vector and the first matched vector on the matching score list, divided by the distance between the selected vector and the second best matched vector on the list. This technique will be explained with details in Chapter 4, please refer to Eq 4.12 and 4.13 for more details. This threshold represents the winning margin between the best matched interest point and the second best. It indicates the uniqueness of this selected vector and the efficiency

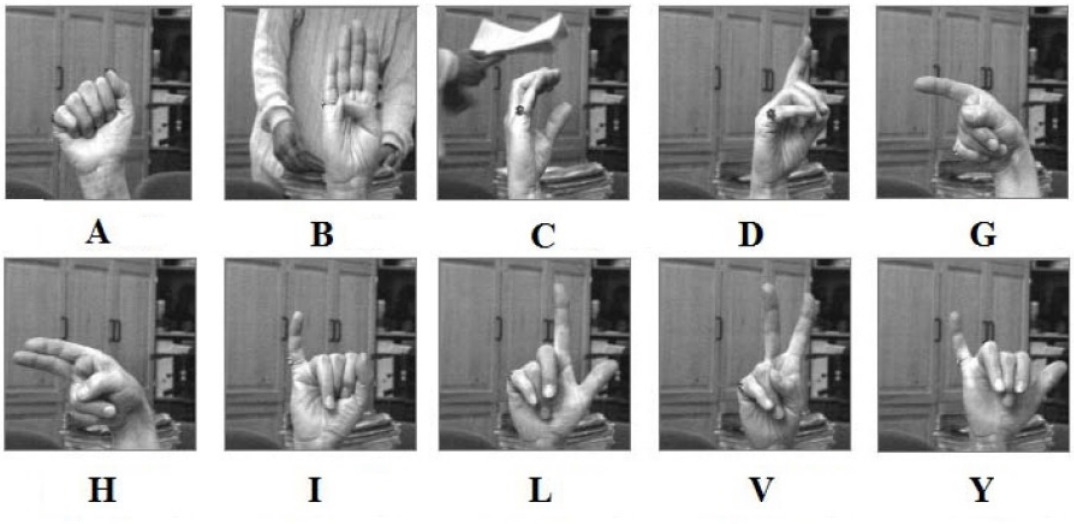

Figure 3.3: All ten pre-defined hand postures in the Triesch Hand Posture Database [7].

of classification using this vector. The value of the threshold between 0.20 to 0.95 are tested for every selected SURF vector, and the optimised value with the best performance is chosen.

The proposed method incurs even less computation in the classification stage than existing boosting based HPR methods such as [2] due to the various number of weak classifiers h_t involved in every strong classifier H_y of each posture class. That means in the classification stage the computation will be focused on those postures with a relatively large number of weak classifiers. The method of [2] uses boosting method to select texture feature SIFT for posture recognition. This is very important for pattern recognition tasks that require real time response like hand posture and gesture recognition. For postures with a high degree of similarity which are relatively harder to classify, like posture 'I' and 'Y' in the Triesch database [7] as shown in Fig 3.6, there will be more features picked out for classification. As shown in Fig 3.7, posture 'I' has 19 selected features. For posture 'L', only 4 features are selected. Because the SURF features of posture 'L' are so discriminative that only 4 of features are enough to distinguish this posture from the others.

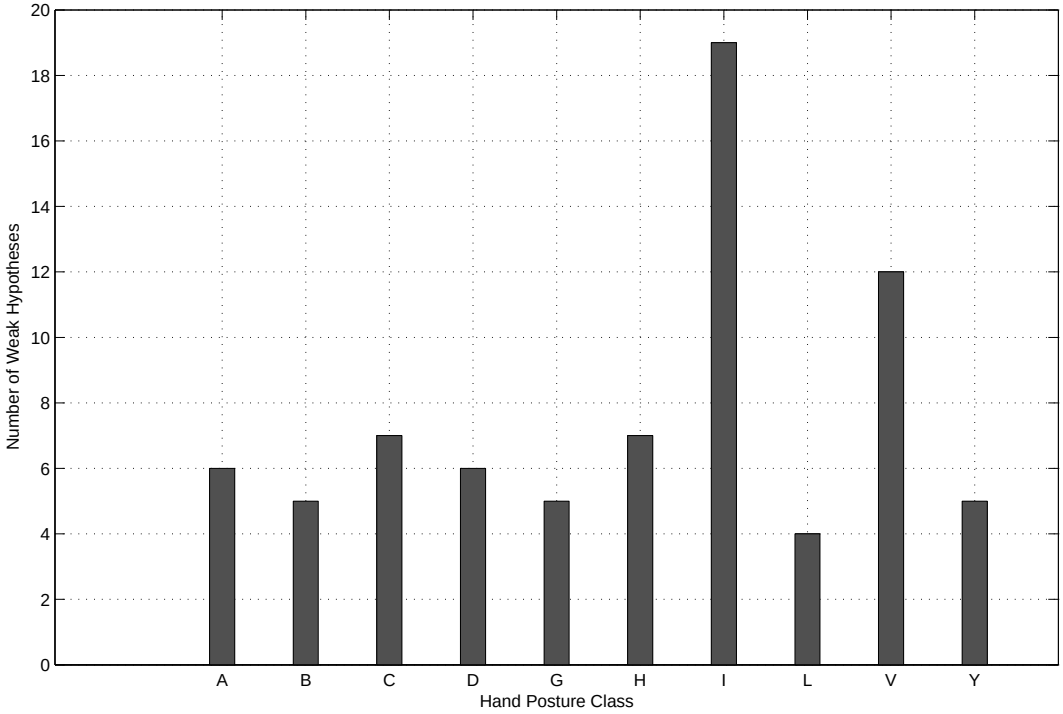

Figure 3.4: Number of weak hypotheses within trained strong classifiers for all 10 postures.

Algorithm 2 Classification process.

Input:

A testing image I, trained strong classifiers $H = \{H_y \,|y = 1, 2, \ldots, Y\}$, where H_y, $y = 1, 2, \ldots, Y$, each strong classifier H_y consists of T_y weak classifiers. Each weak classifier $h_t\,(I, f_t, \theta_t, \alpha_t)$ has weight α_t, SURF descriptor f_t and threshold θ_t.

Extract the SURF key points $S = \{s_1, \ldots, s_m\}$, from I;
Initialise WeightSum = 0;
for $t = 1, \ldots, T_y$ **do**
  Find the best match s_m of h_t in S;
  if Euclidean distance $d\,(h_t, s_m) < \theta_t$, **then**
    WeightSum = WeightSum + α_t;
  end if
end for
if WeightSum $< \mu \sum_{t=1}^{T} \alpha_t$ **then**
  return 1;
else
  return 0;
end if

The classification process is shown with details in Algorithm 2. Given a testing sample, firstly the SURF feature vectors will be extracted. Every weak classifier within all trained strong classifiers will produce a binary matching result based on the thresholds of all weak classifiers. These scores are assigned with the weight of the corresponding weak classifiers. The rationale behind this is that by multiplying the score with the weight of the weak classifier that produces it, the score is weighted by the confidence level of the weak classifier. Hence the higher the weight is, the more contribution the score should be accounted for in the final score of the strong classifier. Then the weighted sum of the results of all weak classifiers in this strong classifier $H_y(I)$ will be produced as the matching result of this strong classifier.

If $H_y(I)$ is larger than a certain percentage μ of the sum of all weights α_t, then the output of the strong classifier is 1. The parameter μ needs to be estimated through experiments. If there are more than one classifiers producing 1 as results, the actual weighted sum in Eq(3.6) from these classifiers will be compared and the one with the highest sum wins.

3.3 Experiments

As the benchmark of HPR, the Triesch Hand Posture Database consists of 10 hand postures performed by 24 persons. There are three kinds of background settings: uniform light, uniform dark and complex. Uniform light and dark backgrounds represent the single-coloured white and black backgrounds, respectively. The complex background means unconstrained background contents. In total, there are 720 greyscale images with size of 128×128 pixels, 72 images for each posture. Some samples of the database are shown in Fig 3.6. The proposed method is tested in two different experiments. The maximum number of features selected for one posture class T is set to 20 based on the performance on the whole training set.

The first experiment is conducted for comparison with state-of-the-art methods, including Triesch et al. (FG 1996) [7], Fang et al. (ICPR 2008) [155] and Kumar et al. (ICARCV 2010) [156]. Triesch et al. [7] proposed a method for hand posture recognition by using elastic graph matching. Fang et al. (ICPR 2008) [155] proposed a co-training method for hand posture recognition using semi-supervised learning that treats each new posture as unlabeled data and updates the classifiers in a co-training framework. Kumar et al. (ICARCV 2010) [156] also proposed a elastic model matching algorithm for hand posture recognition. The same experiment settings in Fang et al.(ICPR 2008) [155] is adopted. For each posture, the training set consists all 30 images from 10 performers and the remaining images of the other 14 performers constitute the testing set.

Another state-of-the-art method Just et al. (FG 2006) [1] reported results on the Triesch Hand Posture Database with different amount of training and testing samples. The method uses the novel features that are based on modified census transform. For comparison with Just et al. [1] the second experiment is conducted with the same amount of training and testing samples as [1]. For each posture class, the training set has images of 8 performers and the testing set has images from the remaining 16 performers. The results are shown in Table 3.1 and 3.2, respectively.

In Table 3.1, the first two columns represent results on light and dark background images, respectively. The third column is the average accuracy on light and dark background images. Results of complex background images and over-all accuracy are in the fourth and fifth column, respectively. Fang et al. [155] did not provide any detailed experimental results, besides the over-all accuracy. The reason for Kumar et al. [156] only providing accuracies on light and dark background images is that, their method can only perform well with unified backgrounds. In other words, the method is sensitive to the complex background textures. Both [155] and [156] are based on elastic models. If there are complex texture in the background, the methods are not able to extract the elastic model accurately since the location

Table 3.1: Results of experiment 1 and comparisons with state-of-the-art methods on the Triesch Hand Posture Database.

	Light	Dark	Average on Unified Back-ground	Complex	Over-all
Triesch et al. (FG 1996)	94.3%	93.3%	93.8%	86.2%	91.0%
Fang et al. (ICPR 2008)	N/A	N/A	N/A	N/A	90.1%
Kumar et al. (ICARCV 2010)	96.8%	95.9%	96.4%	N/A	N/A
Proposed method	93.9%	94.4%	94.2%	90.2%	92.8%

Table 3.2: Results of experiment 2 and comparison with A. Just et al [1].

	Light	Dark	Complex	Over-all
Just et al. (FG 2006)	92.8%	92.8%	81.3%	89.0%
Proposed method	93.6%	94.3%	90.0%	92.6%

of the hand is unknown. The proposed method delivered slightly lower accuracies on the unified backgrounds than Kumar et al. [156], but our method is capable of perform robust HPR against uncontrolled complex background scene settings. Also, the proposed method is the first one to out-perform the original Triesch et al. (FG 1996) [7] method on all experimental categories.

As shown in Table 3.2, the proposed method out-performed Just et al. [1] on all experimental categories. In [1], the classifier is a simple set of liner feature lookup-tables. The lookup-table is not capable of determine which features are the optimised ones that can represent the posture against the complex background. On the other hand, our method is capable of selecting a optimised set of texture features through the training process. With smaller amount of training samples than the original experimental protocol of the Triesch Hand Posture Database, the proposed method still produced satisfactory results.

The challenges from uncontrolled environments in the experiments of this chapter include: 1) complex background texture; 2) high gesture similarity; 3) size,location and orientation variance. That means other challenges listed in Section 1.3 are not considered in this chapter. The reason for that is the close range between the target hand and the camera, which makes some challenges not applicable to HPR, including double handed gestures, continuous gestures, face/hand overlapping, pause during the gesture and speed variance. The remaining challenges on the list in Section 1.3 have not been considered by the HPR community, including large vocabulary size, high gesture complexity, lighting change hands out of the scene and occlusion.

3.4 Conclusions

In this chapter, a novel HPR method is introduced to tackle the challenge of pure appearance-based HPR in uncontrolled environments with cluttered background.

By learning the most effective texture features of the posture classes in the training set with single-coloured background, the proposed method is able to recognise the pre-defined postures with learnt texture features in the testing samples with cluttered background. The main contribution of this chapter is proposing a boosting-based method to select optimised set of texture features to represent the pre-defined postures against each other and the cluttered background. This method could be further improved by applying deep learning methods. Adaptive Boosting is capable of selecting better features from a given feature set, while deep learning methods are capable of generating optimised features.

Chapter 4

Hand Tracking in Uncontrolled Environments

Within the context of HGR, the challenges from the uncontrolled environments including the presence of cluttered backgrounds, moving objects in the background, gesturing hand out of the scene during gestures, pauses during gestures and the presence of other people are the main difficulties that keep this intuitive way of Human Computer Interaction from widely utilised in real-world scenarios. Moreover, the position, scale and length variance of the hand gestures can be large even for the same gesture from the same performer under the same environment. In this chapter, a novel hand tracking method for uncontrolled environments will be introduced. Analysis on the robustness of the proposed tracking scheme against challenges from uncontrolled environments is also included in Section 4.2.

With all the challenges, detecting pre-defined hand gestures from the scene becomes a tough task. Segmenting the target hand from the complex background is the first step to analyse the trajectories. For traditional tracking methods, the task is normally locating the exact coordinates of the target hand candidate. In the unconstrained scene settings that the proposed framework aims to tackle, the background usually contains moving objects, including skin-coloured and hand-like

regions. Namely, there are multiple moving hand regions in the scene. With no prior knowledge, there is no way to distinguish the target signing hand from the other hand regions. For commercialised HGR methods, constraints can be made to distinguish the system user from other distractions in the background, e.g. requiring the user to appear within a certain area of the scene at a certain scale. Also, unlike methods that use depth information, the proposed method has to perform tracking without any knowledge on the Depth of Field (DOF). Hence the method would not be able to discard other interferences based on depth information.

Different from the traditional tracking schemes, instead of segmenting one certain target region and tracking the exact position of the region throughout the video, the proposed tracking method in this chapter detects and tracks all eligible hand candidates in the scene. The rationale behind this is that, similar to the idea of Maximum Entropy Model, the only fair assumption to make under the circumstances when there is no prior knowledge about the feature distribution of the predefined patterns, is no assumption should be made at all. In other words, with no knowledge of the scene content whatsoever, the only fair heuristic rule to make is enabling the selection criteria of the hand candidates as soft as possible.

The proposed tracking scheme uses a set of heuristic rules to match the texture features (SURF) of hand candidates in adjacent frames. The main contributions of the proposed tracking scheme are: 1. This method is capable of adapting to uncontrolled scene contents including lighting variance, gesture scale, speed and location variance; 2. This method does not need a hand segmentation process. 3. It is capable of dealing with multiple hand candidates in the scene.

4.1 Adaptive SURF Tracking

The novel tracking scheme proposed in this thesis is called *Adaptive SURF Tracking*. The key differentiating feature of the Adaptive SURF Tracking is that it can adapt

to arbitrary scene contents, and it does not need a hand segmentation process. The framework can locate and track all eligible hand candidates, namely Regions of Interests (ROIs) from the first frame. The tracking process has three key steps: first frame processing, texture matching and trajectory feature extraction. This section presents detailed explanation of each step.

The objective of the tracking scheme is to perform trajectory feature extraction. The essential method of the trajectory feature extraction is texture matching, namely matching similar texture patterns in the adjacent frames. Limited by the real-time response criteria, the tracking method can only select characteristic texture patterns of the hand candidates for matching, namely the SURF key points. In this way, the exact location of the hand candidates can hardly be determined. The reason is that, the matching error of SURF descriptors causes small displacements on coordinates of all matched SURF key points. Therefore, matching key points introduces certain level of displacement on the centroid of the hand candidates. In our experiments, when the hand candidates are overlapping with fast moving distractions in the background in a video stream with 320×240 pixels resolution and at a rate of 30 frames per second, the matching error can be larger than ten pixels. This disadvantage of key point matching does not affect the trajectory analysis in the proposed HGR framework. That is because the exact coordinates of the hand candidates are not needed in this HGR framework.

4.1.1 First Frame Processing

The reason for introducing *first frame processing* in a dedicated section is that, the proposed tracking scheme only track the hand candidates that appeared in the first frame. In this way, other interferences that enter the scene after the first frame will not cause any texture pattern mismatches. Also, to achieve better response time, the computation for detecting eligible hand candidates in every frame is spared. This strategy is viable under one assumption, which is the target signing hand must

Figure 4.1: Processing of the first frame, (a) The skin color binary image, (b) Results of the denoising process, (c) The initial ROIs, (d) SURF key points within the initial ROIs.

be in the scene from the first frame. Also, the detection of hand candidates plays a vital factor on the response speed of the tracking scheme, since the number of hand candidates affects the complexity of texture matching.

To locate all hand candidates, skin colour cues are used. Processing of the first frame is illustrated in Fig 4.1. Skin colour tone can vary under different illumination conditions. In order to estimate the skin colour under the current lightings in the scene, skin-colour tone has to be estimated using features other than colour cues. Hence, the proposed tracking scheme detects eligible human faces in the first frame using the Viola-Jones face detector [157]. Because the Viola-Jones method can detect facial regions based on texture features rather than colour cues.

Then the thresholds in the HSV colour space for producing the skin-colour binary image (Fig 4.1a) are estimated using the pixels within the detected facial regions. If no faces are detected in the first frame, a Gaussian Mixture Model (GMM) in the RGB colour space which trained out of a large skin-colour database [158] will be used to produce the skin-colour binary image, until eligible facial regions are detected in a later frame. A simple thresholding strategy is used for binary skin pixel classification with the lower bound threshold $T_{skin,Min}=(t_{Min,H}, t_{Min,S}, t_{Min,V})$, and the upper bound threshold $T_{skin,Max}=(t_{Max,H}, t_{Max,S}, t_{Max,V})$, for all three channels of the HSV colour space. The two thresholds are calculated as:

$$T_{skin,Min}=\mu\left(\mu_H, \mu_S, \mu_V\right) - \sigma\left(\sigma_H, \sigma_S, \sigma_V\right) \tag{4.1}$$

$$T_{skin,Max}=\mu\left(\mu_H, \mu_S, \mu_V\right) + \sigma\left(\sigma_H, \sigma_S, \sigma_V\right) \tag{4.2}$$

where,

$$\mu\left(\mu_H, \mu_S, \mu_V\right) = \frac{\sum_{f\in F}\sum_{(x,y)\in f} v_{x,y}\left(h, s, v\right)}{\sum_{f\in F} A_f} \tag{4.3}$$

$$\sigma\left(\sigma_H, \sigma_S, \sigma_V\right) = \sqrt{E\left[v_{x,y}(h,s,v)^2\right] - \mu(\mu_H, \mu_S, \mu_V)^2} \tag{4.4}$$

are mean and standard deviation vectors for all three HSV channels of all facial pixels that lie in each face region f, within the set of all detected human facial regions F. $v_{x,y}$ is the pixel value on coordinate (x, y), and A_f, $f \in F$, indicates the area of the detected facial regions. The thresholding process is shown below:

$$C_{x,y} = \begin{cases} 255, & T_{skin,Min} \leq v_{x,y}\left(h,s,v\right) \leq T_{skin,Max} \\ 0, & otherwise \end{cases} \tag{4.5}$$

where $C_{x,y}$ is the pixel value at coordinate (x, y) in the skin-colour binary image. In other words, only pixels lie in the range between $T_{skin,Max}$ and $T_{skin,Min}$ are determined as the skin-coloured pixels. By using simple band-pass filter on pixels, the processing speed of skin-colour detection is ensured. Also, the simplicity of skin-colour detection makes it possible to calculate additional colour cues for ruling out non-skin-coloured background distractions after the first frame.

At this stage, the error rate of skin-colour detection is relatively high, due to the following reasons:

- Error of face detection: The Viola-Jones detector is one of the most popular face detectors. But with the presence of printed artificial human faces, it is possible that the background distractions could be wrongfully determined as human facial regions.

- Poor lighting conditions: The proposed HGR framework aims at tolerating environments with unsatisfying lighting conditions. Within the detected facial regions, shadow areas usually cause distorted thresholds. Hence for extreme lighting conditions, the accuracy of this skin detection thresholding process could be low.

- Background skin-coloured distractions: With multiple skin-coloured regions in the background, overlapping of skin-coloured regions is expected. However the skin detection process is not able to segment different skin colour objects in a connected skin region. Thereby overlapping skin regions are likely to be treated as one Region of Interest.

- Simple thresholding: Since only band pass thresholding is used, the quality of skin detection is limited. Using simple thresholding strategy increases the speed of the framework.

Also, the low precision of skin region detection has only limited influence on the over-all performance of the framework. The reason is that the whole point of skin detection is to rule out the regions that are unlikely to be hands and narrow down the choices of hand candidates. The significance of the first frame processing is to include the target hand region in one of the ROIs, instead of locating all possible hands with high level of precision. As long as the target hand is covered in one of the ROIs, its trajectory will be tracked and analysed in the framework.

Then in the binary skin colour image, all closed exterior contours are located. That means the contours which are included within other contours are discarded. The priority is to lower the number of hand candidates in the first frame, rather than segmenting boundaries of all overlapping skin regions. A denoising process is performed on all the closed contours in the skin-colour binary image. All the interior contours and contours with areas smaller than a threshold T_{dsr} are deleted (Fig 4.1b).

$$T_{dsr} = \bar{A}_f \times 0.25 \tag{4.6}$$

where $\bar{A}_f$ is the average area of all the detected facial regions in the first frame. Intuitively the denoising threshold should be defined based on the average hand regions, but no assumptions on the areas of hands should be made in completely

uncontrolled environments. Also, the only concrete data the tracking scheme calculated so far is the area of the facial regions, and the hand and facial regions have similar areas. Hence the denoising threshold is calculated based on the facial regions. Then the ROIs are defined as the minimum bounding rectangles of the remaining contours (Fig 4.1c). Then the SURF key points are extracted from the ROIs in the first frame.

4.1.2 Texture Matching

After the first frame, all the eligible hand candidates are located. In the rest of the video, the trajectories of all hand candidates are tracked by matching texture features. Since the SURF key points are used to represent the texture patterns, the basic idea of tracking is to match SURF key points within all ROIs in the adjacent frames. From the second frame, SURF key points are extracted from the whole frame at current time t, and matched with SURF key points from ROIs in the frame at time $t-1$. The rationale behind this is that, for the same gesture sample, the displacement of ROI between adjacent frames depends on the frame rate of the video stream. For the same gesture sample, the lower the frame rate is, the larger the displacement would be. This framework is specially designed for uncontrolled environments, that includes the unconstrained video specifications. Hence, to adapt to different frame rates, there is no assumption on the exact current locations of the ROIs. Namely, the only assumption is that the displacement would falls within a reasonably range which will be explained later in this section. No prior knowledge on the movement direction and speed in the current frame is used. Based on this, the SURF key points in the whole frame is extracted to match with the textures of ROIs in the previous frame, rather than making a heuristic rule of possible ROI locations in the current frame.

Assuming the set of SURF key points

$$S_t = \{S_t^p \,|p = 1, 2, \ldots, P_t\} \tag{4.7}$$

of the $r^{th}ROI$ in the frame at time t contains P_t key points, and every SURF key point

$$S_t^p = (X_t^p, Y_t^p, E_t^p) \tag{4.8}$$

contains coordinates (X_t^p, Y_t^p) and the corresponding 64-dimensional SURF descriptor E_t^p. We define a matched SURF key point S_t^p for a given key point S_{t-1}^p as:

$$S_t^p = \underset{S_t^p \in S_t}{\arg\min} \left\| E_t^p - E_{t-1}^p \right\| \tag{4.9}$$

and the key point S_t^p is subject to the condition:

$$\left\| E_t^p - E_{t-1}^p \right\| \Big/ \left\| E_t^{p'} - E_{t-1}^p \right\| \geq T_{match} \tag{4.10}$$

where $E_t^{p'}$ is the SURF descriptor of the second best matching point $S_t^{p'}$ for S_{t-1}^p:

$$S_t^{p'} = \underset{S_t^{p'} \in S_t - S_t^p}{\arg\min} \left\| E_t^{p'} - E_{t-1}^p \right\| \tag{4.11}$$

The Euclidean distance between the two 64 dimensional SURF descriptors is used as the matching score. Only when the ratio of the best match and the second best is larger than a threshold T_{match}, the best match can be used as the valid matching SURF key point (Fig 4.4). In this way, only when the best match is winning by a large margin can it be counted. If there is no such matching SURF point S_t^p found, the given SURF key point S_{t-1}^p is discarded. Namely, the uniqueness of the matching is the only criteria. When there are multiple regions with the similar texture of the given SURF key point, this matching is considered with low level of

distinctiveness. Thereby, only the most distinctive SURF key points are used to represent the texture patterns of the ROI. Also, this strategy can keep the number of matching SURF key points relatively low, which is a positive factor for fastening the processing speed. There is another merit of this matching strategy. For video streams with relatively large resolutions, the texture in the ROIs would be naturally richer. That will rise the numbers of extracted SURF key points within the ROIs (Fig 4.5). By setting the matching threshold T_{match}, the number of matched SURF key points in the same ROI with different video resolutions will be levelled down to the same scale. Thereby texture matching in various image resolutions would have similar computational complexity.

As Fig 4.5 shows, the number of SURF key points rises with the increasing video resolutions. Even for controlled environments with relatively simple texture in both the foreground and background, with 640 * 480 pixels resolution, there are still over 100 key points extracted from the frame. Learned from the experiments on Hand Posture Recognition in uncontrolled environments, to recognise a hand region in complex background, only 10-20 key points are needed to represent the texture patterns. Hence, it is essential to down-sampling the frame when it is in a relatively high resolution. In the experiments of Chapter 5, 6 and 7, the frames resolution is set to 320 * 240 pixels.

Fig 4.6 shows the decreasing tendency of the amount of matching SURF key points with respect to the decreasing value of T_{match}. If the threshold for matching is set too small, for complex background, the chance of mismatching would be high. In the experiments of this thesis, the threshold is set to 0.8. If the frame resolution has a certain level of influence on the SURF extraction process, naturally it can also affect the texture matching. As shown in Fig 4.7, the changing resolutions can affect the number of matching key points. The higher the resolution, the more matching points would be found.

Once the matched pairs $M_t = \left\{ \left\langle S_{t-1}^1, S_t^1 \right\rangle, \left\langle S_{t-1}^2, S_t^2 \right\rangle, ..., \left\langle S_{t-1}^{P_{t-1}}, S_t^{P_{t-1}} \right\rangle \right\}$

Figure 4.2: Texture matching for the ROI of the target signing hand.

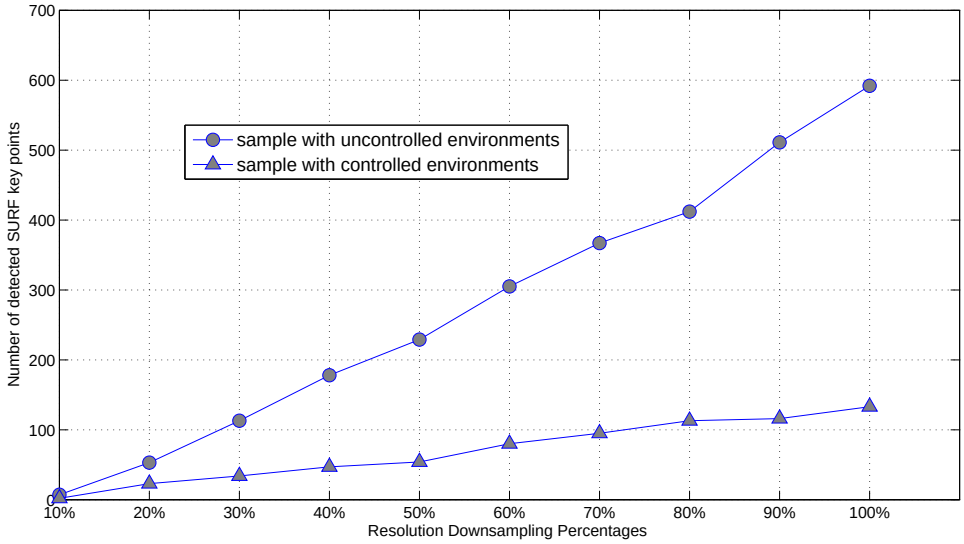

Figure 4.3: The graph shows the number of SURF key points extracted from different downsampled resolutions of the same image. The lines in the graph represent the number of detected SURF key points from two images with uncontrolled and controlled scene settings, respectively. The full image size is 640 * 480 pixels.

are found, where M_t is the set of the matching pairs. A pruning process is performed on all matched pairs in the ROIs. Only the matching texture key points with displacement in a certain range are preserved. All the matched pairs with displacements smaller than the lower bound $T_{\min,t}$ of the ROI's displacement range are

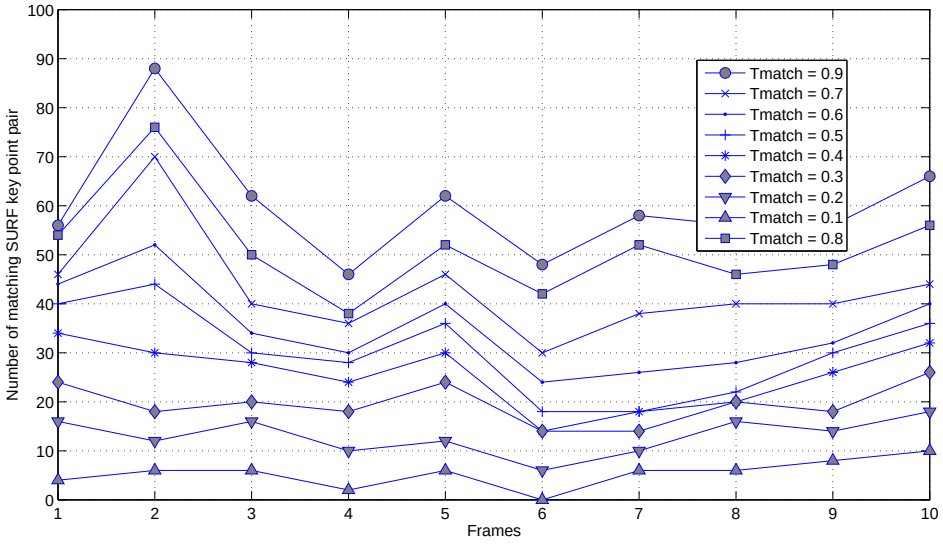

Figure 4.4: For the first 10 frames in the sample displayed in Fig 4.5, with a fixed resolution of 340*240 pixels, this graph shows how the number of matching SURF key point pairs changing with different values of T_{match} from 0.1 to 0.9.

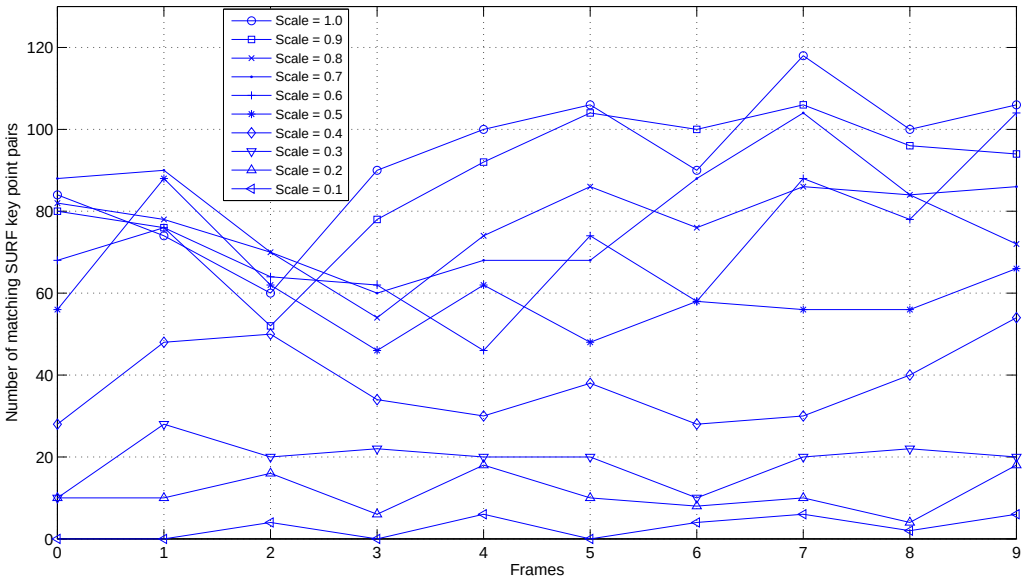

Figure 4.5: For the first 10 frames in the sample displayed in Fig 4.5, with fixed T_{match} value 0.9, this graph shows how the number of matching SURF key point pairs changing with different value of scale from 10% to 100%.

dropped. On the other hand, if a matched key point has displaced more than the upper bound $T_{\max,t}$ of the ROI's displacement range in the next frame, it is most likely a mismatch. The fast moving objects would cause motion blur and it is hard to extract texture features in motion blur regions. Thereby no matching can be found in the blurred areas. Then that is a reasonable assumption that matching key points with large displacements are mismatches, because no object is capable of moving that far within such a short period of time without causing massive motion blurring. The lower and upper displacement bounds of the ROIs in frame t are calculated based on the average displacement of all matched key point pairs in the ROIs between frame $t-2$ and $t-1$, namely $M'_{t-1} = \left\{ \left\langle S^1_{t-2}, S^1_{t-1} \right\rangle, \left\langle S^2_{t-2}, S^2_{t-1} \right\rangle, ..., \left\langle S^{P'_{t-2}}_{t-2}, S^{P'_{t-2}}_{t-1} \right\rangle \right\}$, where the prime symbols indicate "after pruning". The displacement bounds constitute the example of adaptive tracking in uncontrolled environments. Due to the small time window, the motion of objects has a certain level of consistency on speed in the adjacent frames. Hence, to prune off mismatches of SURF key points, the bound for displacements should be calculated based on the current speed of the ROIs. Namely, if the ROIs are accelerating, the upper bound should be raised to include matches with larger displacements. The definition of the lower and upper bounds are:

$$T_{\min,t} = \frac{\sum_{p=1}^{P'_{t-2}} \left\| \left(X^p_{t-1} - X^p_{t-2}, Y^p_{t-1} - Y^p_{t-2} \right) \right\|}{P'_{t-2}} \times F_{mov,\min} \tag{4.12}$$

$$T_{\max,t} = \frac{\sum_{p=1}^{P'_{t-2}} \left\| \left(X^p_{t-1} - X^p_{t-2}, Y^p_{t-1} - Y^p_{t-2} \right) \right\|}{P'_{t-2}} \times F_{mov,\max} \tag{4.13}$$

$F_{mov,\min} = 0.25$ and $F_{mov,\max} = 3$ are adjusting factors of minimum and maximum displacement, the values are chosen through experiments. For the first frame, various default displacement ranges have been tested and we found that the default values of the lower and upper bounds $T_{\min,0} = 3$ and $T_{\max,0} = 40$ pixels were empirically feasible. Also if $T_{\min,t}$ is less than the default value, it would be set to

the default value. Hence for the stationary regions (e.g. facial regions), where no large movements can be found, the majority of the matched key point pairs would be dropped, so the challenge of face/hand overlapping is naturally resolved. The displacement range of ROIs is recalculated in every frame adaptively based on the movement distances of the main objects in the ROIs. Hence the proposed tracking scheme can adapt to speed changes of the target. With pre-defined trajectories in the vocabulary of this thesis, namely the hand signed digits from 0 - 9, the strokes could have 1 to 2 corners. That means the signing hand will stop and accelerate again at every corner. If the displacement thresholds for pruning is fixed, then to cover object moving in different velocities, the acceptable displacement range must be set wide, which could cause high probability for including mismatches. Thereby, to keep the ROIs as fit to the hand candidate as possible, the threshold must be adaptively calculated based on the current velocities of the ROIs. If the target, namely the main objects in the ROIs are accelerating, the displacement range will move up according to the actual acceleration, which can be represented by the average displacement of all the matched key point pairs. An example of pruning is shown in Fig 4.9b.

After the pruning process, the new ROIs in the current frame are drawn. The new ROIs in the current frame are defined as the minimum bounding rectangle of the remaining matched key points after pruning. Instead of only keeping the matched key points in the new ROIs of the current frame, all key SURF points within the new ROIs are preserved for matching with the SURF key points in the next frame. The pruning process discards a large percentage of key points in the ROIs. The bounding rectangles of the remaining matched key points do not necessarily fit the contours of the hand candidates in the ROIs. Normally the matched key points concentrate in a subregion of the hand candidates, the new ROIs are only covering the subregion, rather than the minimum bounding rectangle of the exterior contour of the hand candidates. The solution is to increase the sizes of the newly drew ROIs,

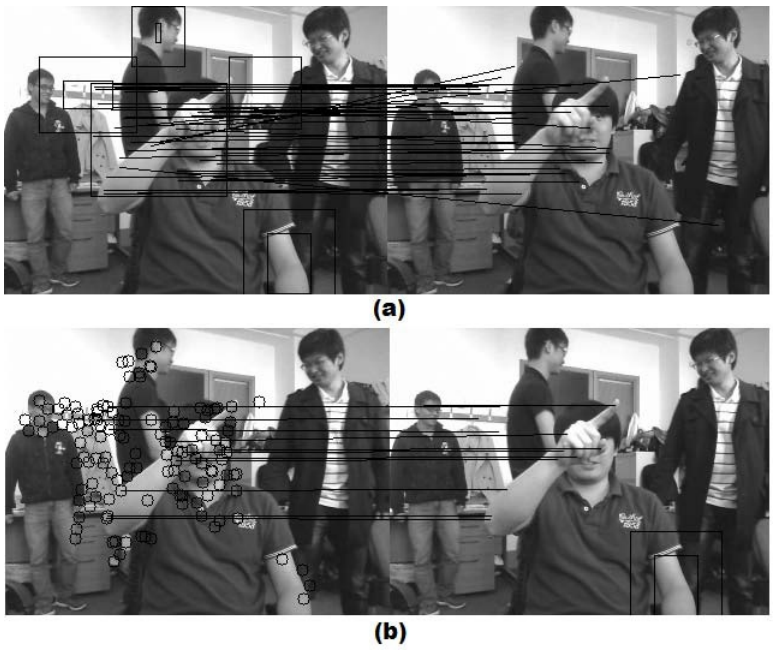

Figure 4.6: Pruning process. (a) matched key point pairs from one of the ROIs, between the previous frame (left) and the current frame (right), (b) the remaining matched key point pairs after pruning.

to cover the entire hand candidate regions. Hence, the ROIs need to be enlarged to make sure that the new ROIs can cover the corresponding hand candidates.

Assuming the number of the remaining matched key points after pruning is P_t' and the area of the new r^{th} ROI is A_t, the boundaries of the new r^{th} ROI are then extended by e_t pixels. The value of ROI extension is determined by the amount of remaining SURF key points after pruning and the area of the new ROIs before extension. The extension rules can be summarised as: a) Only the ROIs with areas that are smaller than a pre-defined threshold A_{ROI} will be extended; b) The ROIs with number of remaining SURF key points less then a threshold T_e which after pruning will be assigned with larger extension values. The extension value e_t is defined as:

$$e_t = \begin{cases} 0, A_{r,t} > A_{ROI} \\ \exp\left(-\frac{A_{r,t}}{A_H}\right) \times E_r, A_H < A_{r,t} < A_{ROI} \\ \left[\exp\left(-\frac{P'_{t-1}}{P_{\min}}\right) + E_{boost}\right] \times E_r, A_{r,t} < A_H \wedge P'_{t-1} \leq T_e \\ \exp\left(-\frac{P'_{t-1}}{P_{\min}}\right) \times E_r, A_{r,t} < A_H \wedge P'_{t-1} > T_e \end{cases} \tag{4.14}$$

where T_e is set to 3 in this thesis, and

$$A_{ROI} = (h_s \cdot w_s)/20 \tag{4.15}$$

$$A_H = (h_s \cdot w_s)/60 \tag{4.16}$$

A_{ROI} is the estimated maximum area of ROIs that are qualified for extension. Definition in Eq. 4.15 is based on accuracy tested on the training set of Warwick Hand Gesture Recognition for isolated gestures (will be introduced in Chap 5). The reason for setting this threshold is that, if the ROI already covers a large area due to mismatch of texture or high velocity of the hand candidate, further extension would cause redundant background textures to be included in the ROI. h_s and w_s are the height and width of the frame respectively.

A_H is the estimated plausible area of the hand regions. Definition in Eq. 4.16 is based on accuracy tested on the training set of Warwick Hand Gesture Recognition). This parameter does not have to be precise to various gesture scales, since it merely affects the extension of ROIs. However, by using this parameter the precision of ROIs' fitness to the hand candidates is enhanced. $P_{\min}$ is the empirically estimated suitable amount of the remaining SURF key points after pruning, the value of 10 is based on accuracy tested on the training set of Warwick Hand Gesture Recognition for isolated gestures. E_{boost} is a coefficient to ensure that the lower the

value of P'_{t-1} is, the higher the extension is given to the ROI. The value of E_{boost} is set to 0.3 empirically based on accuracy tested on the training set of Warwick Hand Gesture Recognition for isolated gestures. E_r is the enlargement scale for the r^{th} ROI, which is defined as:

$$E_r = \begin{cases} \left[\left(h_{r,0} + w_{r,0}\right)/2\right] \cdot F_s, & h_{r,0} \cdot w_{r,0} < \bar{A}_f \cdot 2.5 \\ \sqrt{\bar{A}_f} \cdot F_s, & otherwise \end{cases} \tag{4.17}$$

where $h_{r,0}$ and $w_{r,0}$ are the initial height and width of this r^{th} ROI in the first frame. $F_s = h_s \cdot w_s/30$ is the enlargement factor corresponding to the frame size. Hence e_t also depends on the original size of this ROI in the first frame. For hand candidates in a large scale during the gesture, namely the hand candidates that are appearing in a close range to the camera, the ROI extensions should be increased accordingly. After the extension of the ROIs, all eligible SURF key points within the ROIs are extracted as the texture feature for locating the ROIs in the next frame, regardless of whether they have matching key points in the previous frame.

4.1.3 Trajectory Feature Extraction

From the pattern recognition point of view, the more discriminative and uncorrelated the features are, the better results the classifier should produces. Instead of using the combination of hand candidate's position, speed and orientation as trajectory feature, the only trajectory feature used in this framework is movement orientation. The reasons of this are twofold.

Firstly, to tackle the position variance of the gesture performer, the location of hand candidates should not be used. Some methods [8, 54, 77] normalises the trajectories of hand candidates into the same coordinate system to deal with the location variance. The drawback of this approach is that the processing must begin after the gesture performer finishes the whole gesture. Coordinate system transfor-

mation requires the centroid of the trajectory, which can only be calculated with the full trajectory. Therefore, it makes the real time response a harder task for the trajectory classification (definition of real-time response will be explained in Section 5.2.3).

Secondly, using speed of hand candidates as feature makes the method sensitive to velocity variance, which could be at a relatively large scale due to various signing habit of different individuals. However, the acceleration of the hand candidates is a viable trajectory feature, since it can be seen as normalised speed. The texture matching method used in the proposed framework is not designed for extracting the exact positions of the hand candidates. Thereby the precision of the acceleration calculated based on texture matching is not satisfactory enough to improve the overall accuracy. Hence, speed is not used as the trajectory feature in the proposed framework.

For every frame, after the texture matching, the dominant movement directions of all ROIs are extracted as the trajectory feature. Since there are P'_{t-1} matched SURF pairs between frames t and $t-1$ after pruning in the r^{th} ROI. The corresponding dominant movement direction of this ROI in the frame t is defined as:

$$\mathrm{drt}\,(t,r) = \arg\max_{d} \{q_d\}_{d=1}^{D} \tag{4.18}$$

where $\{q_d\}_{d=1}^{D}$ is the histogram of the movement direction, and d indicates the index of directions. q_d is the d^{th} bin of the histogram. The width of each bin is α, $D = 360^\circ/\alpha$ is the total number of bins.

Various values for α have been tested, 20 degrees is employed which can produce the best results. Fig 4.10 - 4.13 illustrate the movement orientation vectors of all training samples of gesture "6" from the Palm Graffiti Digits database [8], with α set to 10, 20, 30, 40 degree, respectively. q_d is defined as

$$q_d = C \sum_{p=1}^{P'_{t-1}} k\left(\|S_t^p\|^2\right) \delta\left(S_t^p, d\right) \tag{4.19}$$

where $k(\cdot)$ is a monotonic kernel function which makes the key points that are located far away from the centre of the ROI having smaller weights. $\delta\left(S_t^p, d\right)$ is a simple Kronecker delta function used to see whether the direction of $\left\langle S_{t-1}^p, S_t^p \right\rangle$ falls in the d^{th} bin. The constant C is a normalisation coefficient defined as

$$C = 1 \Big/ \sum_{p=1}^{P'_{t-1}} k\left(\|S_t^p\|^2\right) \tag{4.20}$$

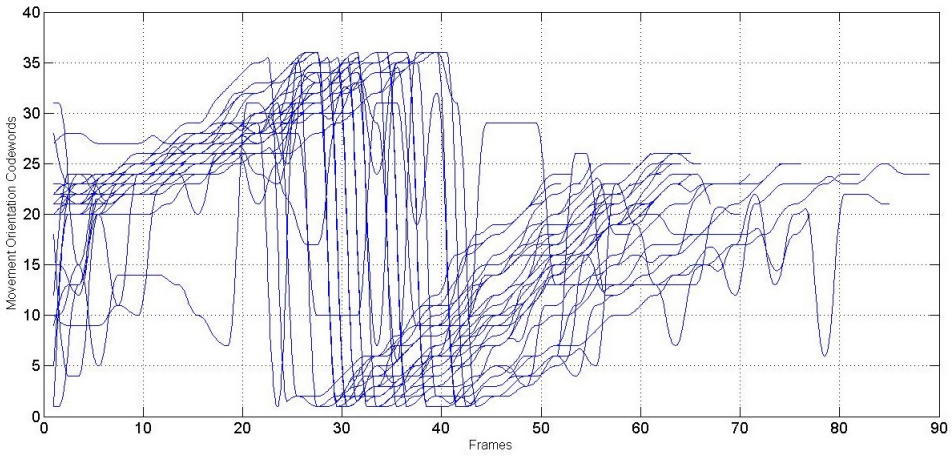

Figure 4.7: The movement orientation vectors of all training samples of the gesture "6" from the Palm Graffiti Digits database [8], with orientation bin size 10 degree.

4.2 Robustness

With the problem statement in Section 1.3 and the introduction of Adaptive SURF Tracking in Section 4.1, further analysis on how the tracking scheme of the proposed framework tackles some of the challenges from the uncontrolled environments is given in this section. More analysis on how the gesture classifier overcomes the remaining challenges identified in Section 1.3 can be found in Section 5.3.

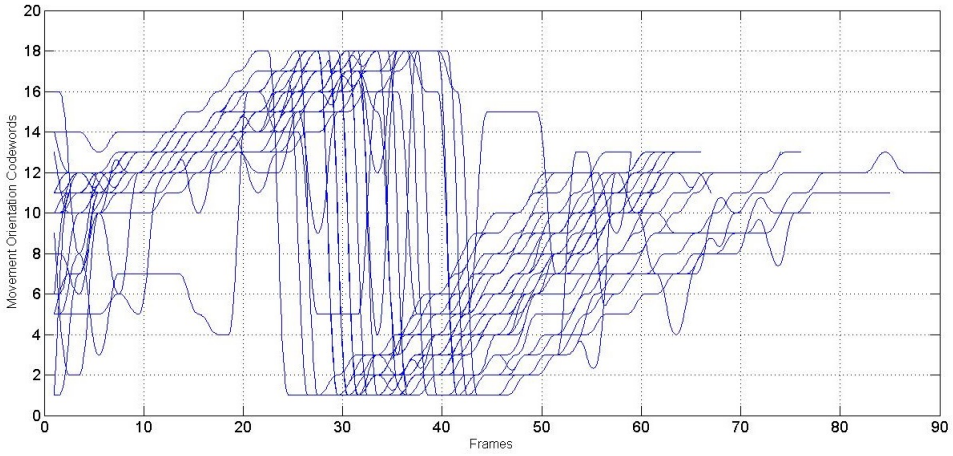

Figure 4.8: The movement orientation vectors of all training samples of the gesture "6" from the Palm Graffiti Digits database [8], with orientation bin size 20 degree.

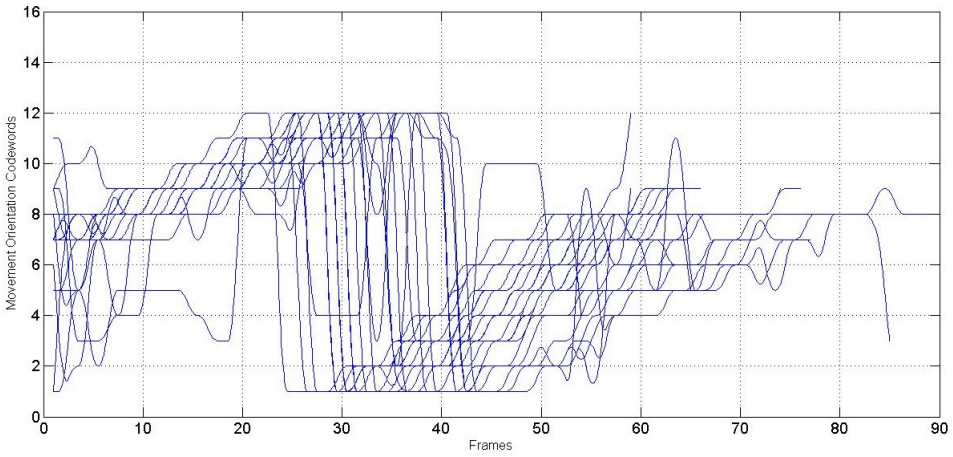

Figure 4.9: The movement orientation vectors of all training samples of the gesture "6" from the Palm Graffiti Digits database [8], with orientation bin size 30 degree.

4.2.1 Changing Lighting Conditions

To calculate skin colour tone under different lighting conditions, the proposed tracking scheme needs to locate some skin-coloured regions using features other than colour cues. Hence a texture feature based face detector is used to locate eligible human facial regions before the calculation of skin colour. The proposed tracking scheme estimates the skin colour tone with the pixels on the detected human fa-

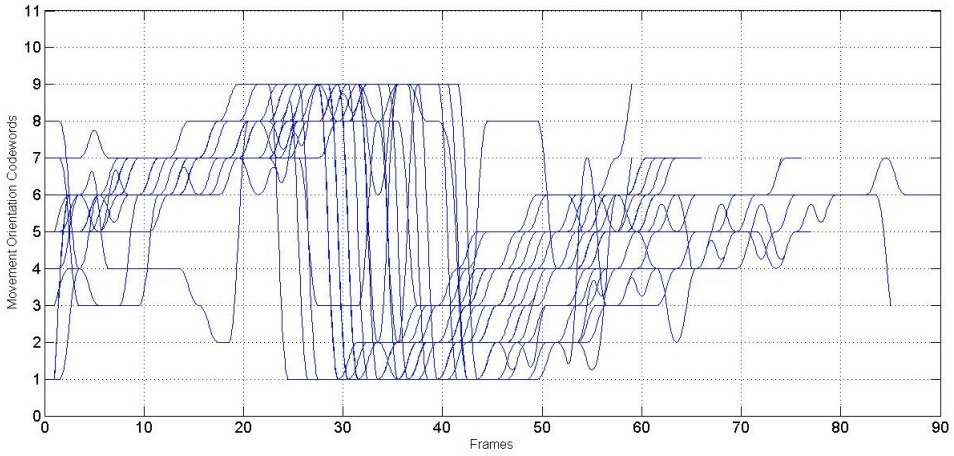

Figure 4.10: The movement orientation vectors of all training samples of the gesture "6" from the Palm Graffiti Digits database [8], with orientation bin size 40 degree.

cial regions in the first frame. In this way, the proposed framework is capable of detecting skin-coloured regions under changing lighting conditions.

4.2.2 Background Distractions

Additional distractions in the background is the main challenge for most of the existing works in the research field of HGR. The categorisation of different distractions is listed below, with explanation of the countermeasures in the proposed framework.

- Distractions appearing in the first frame:
  - Skin-coloured regions (stationary or moving): For small skin-coloured regions, they would be discarded by the pruning processing during the selection of hand candidates (see Section 4.1.1). For the eligible hand candidates, their trajectories are recorded by the Adaptive SURF Tracking method. The distractions with meaningless trajectories would not affect the proposed method much (please see more details in Section 5.2.2). Since the proposed framework does not distinguish the target hand from the background distractions, the trajectories of the distractions that have

certain level of similarity with the pre-defined gestures classes would cause certain level of confusion for the classifier. The proposed method tackles these situations with a novel classifier weighting algorithm called Partition Matrix which will be introduced in Section 5.2.2. For background regions that are overlapping with the target hand (including hand/face overlapping), the textures of these regions are hidden behind the hand candidate. The texture features are extracted again in every frame and the ROI is kept as fitting to the hand candidate as possible. Hence the background area within the ROI is relatively small. Namely the majority of the SURF key points would be extracted from the hand candidate. Therefore the dominant movement orientation must be the moving direction of the hand candidate.

  - Non-skin-coloured regions (stationary or moving): These regions could not pass the pruning process of hand candidate selection (see Section 4.1.1).

- Distractions entering the scene after the first frame:

  - Skin-coloured regions (stationary or moving): Since the skin-coloured distractions that enter the scene after the first frame are not treated as hand candidates, there are only two scenarios that these distractions could cause negative influences. One is when the distractions are overlapping with one of the ROIs. If the distractions are in the foreground, this situation falls into the category of frontal occlusion (will be explained later). If they are in the background of the ROI, the overlapping would not cause any problem to the tracking scheme. The distraction's texture that occluded by the ROI does not affects the tracking scheme. The other scenario is the distractions present similar texture feature as the ROIs. Namely, other people are trying to sign a gesture alongside the

gesture performer with the same hand posture. There is the possibility that during the texture matching, the texture features located in the distraction areas could have a certain amount of matches. If the distractions are at distance from the ROIs, these matches would be discarded during the pruning for matched SURF key points, due to the large displacements. If the displacements of the texture matches with the distractions are within the pruning thresholds (Eq 4.15 and 4.16), namely the distractions are right next to the ROIs without any overlapping, the mismatches could make the new ROI boundary in the current frame cover both the hand candidate and the distractions. This is the only scenario that skin-coloured distractions appeared after the first frame could lead to mis-tracking on the hand candidates. Other than this, the proposed framework is not sensitive to the background distractions.

– Non-skin-coloured regions (stationary or moving): For the distractions with non-skin-colours, they will not cause any trouble since they are discarded by the skin-colour detection step in every frame.

4.2.3 Frontal Occlusion and Hand Out of the Scene

If the object that is occluding with one of the ROIs is non-skin-coloured, then this ROI would be discarded by the skin-colour detection process, and there would be no matching texture features for this ROI. The tracking scheme keeps the current position of the ROI for the next frame. Hence the ROI remains on the same location until the occluding object passes and viable texture matches present themselves. If the occluding object is skin-coloured, the tracking results depends on the level of texture similarity between the distraction and the hand candidate. If there are no viable texture matches, the tracking method keeps the current position of the ROI for the next frame. If there are some texture matches, namely the occluding object is presenting similar hand posture as the hand candidate, the tracking scheme could

fail.

The challenge of hand candidates going out of the camera scope could be treated as frontal occlusion. The only difference between the countermeasures for this situation and the frontal occlusions is that the tracking method no longer performs skin-colour detection for this ROI if the hand candidate is out of the scene. The tracking method keeps tracking the texture within the ROI, which is likely to be the arm region, until the hand returns to the scene.

4.2.4 Pause During Gestures

As for the challenge of hand candidates pausing during the gesture, methods that use moments, normalised hand positions or hand velocity as trajectory features could suffer from this challenge. In the proposed framework, if the hand candidate is remaining in the same position, namely there are no displacements of the matching texture features, the current frame will be discarded in the trajectory, until a new movement direction presents itself.

4.2.5 Speed Variance

As discussed in Section 4.1, due to the fact that the thresholds for pruning the matched SURF key points are calculated based on the movement speed of the ROI in the previous frame (Eq 4.15 and 4.16), the proposed framework is capable of tracking hand candidates with changing speed.

4.2.6 Location Variance

For hand candidates located in different areas of the scene, the proposed method can extract the trajectories regardless of the location variance. In the first frame, the locations of the hand candidates are irrelevant to the tracking scheme, since the hand candidate detection is based on skin-colour detection, not prior knowledge on the possible hand locations.

4.3 Conclusions

A novel hand tracking method is proposed in this chapter. The main advantages of this method over the existing methods are: 1. It does not need any segmentation process. 2. It is capable of dealing with multiple hand candidates in the scene. 3. It can adapt to various lighting conditions, gesture scales, speed and locations. The proposed tracking method provide hand candidate trajectories to the gesture classification method in Chapter 5 and the gesture spotting method in Chapter 6. This method uses SURF as the texture feature for its rotation and scale invariance properties. For real-world applications that demand low response delay, alternative texture features could be used such as BRISK and ORB. Utilising these short binary descriptors can lower the computational complexity, but at cost of invariance properties and accuracy. Also, SURF has certain level of tolerance for view-point variance, which BRISK and ORB do not have.

Chapter 5

Probabilistic Model based Hand Gesture Recognition for Uncontrolled Environments

In this chapter, a novel weighting scheme is proposed to perform HGR in uncontrolled environments. The weighting scheme uses the initial classification results from the trained HCRF model to monitor temporal features on different scales and analysis trajectories of all hand candidates. The main contributions of this method are: 1. It can monitor temporal features on different scales. 2. It is capable of dealing with multiple hand candidates at the same time. For classification of the dynamic hand gestures, the task is essentially predicting an output class label y for the input sequence data $X = \{x_0, x_1, x_2, \ldots x_n\}$ with dependencies, which is also called the observation sequence. Namely, there are dependencies among the variables in the input vector. This task is at the root of many applications in different research fields include locating a specific gene segment in a strand of DNA [159], human activity prediction [160], parsing natural scenes [161] and natural language processing [162]. Many methods have been tested for solving this widely

shared problem, include sequence matching methods such as Dynamic Time Warping (DTW) [163], and probabilistic models, such as Bayesian Network, Maximum Entropy Model (MEM) and Hidden Markov Model (HMM). Probabilistic model is one of the most intensively studied categories in machine learning and pattern recognition because of the effectiveness of the natural principle of summarising patterns from training data through statistics. Among the probabilistic models, there are two major types of models with different basic strategies, *generative models* and *discriminative models*. Generative models typically estimate a joint probability distribution of the input observations and the class labels, through maximizing the joint likelihood function of all training samples. This type of probabilistic model suffers from a major drawback, which is making the assumption of independence among variables in the input sequence. Hence, for segmenting and labelling sequence data with long range dependencies, the discriminative models which directly simulate conditional probability distribution between the sequence data and the class labels, can potentially produce more satisfactory results than the generative models. With this strategy, the model is capable of summarising dependencies and correlations among the input observations.

In the context of HGR, the trajectories of hand candidates are sequences of trajectory features extracted from the video frames. The main characteristic of various trajectories is the combinations of the trajectory features, instead of the values of single trajectory features. In this thesis, Conditional Random Fields (CRF) and its variations are used for recognising dynamic hand gestures from uncontrolled environments. This chapter is dedicated to introduce the proposed solution for hand trajectories classification based on CRF. In Section 5.1, key advantages and issues of applying CRF on gesture classification are discussed. In Section 5.2, the proposed gesture classifier is introduced with analysis on the main advantages and disadvantages.

5.1 Advantages and Issues of Applying CRF on Gesture Classification

Probabilistic models (or statistical models) have been intensively studied for decades in the Machine Learning community. Among them, Conditional Random Fields [146] was introduced for segmenting and labeling sequence data. The potential of CRF was rapidly recognised by the computer vision community [164, 165, 166, 147]. CRF was introduced with two main advantages. One is to relax strong independence assumptions in the generative models. Another is avoiding the Label Bias Problem. Discussions on the difference between the discriminative models including CRF and the generative models is given in Section 5.1.1. The Label Bias Problem within the context of HGR is discussed in Section 5.1.2.

5.1.1 Generative Models versus Discriminative Models

Probabilistic models that focus on modelling the joint probability of the input observation sequence X and the class label y, $p(y, X)$ are called generative models. The essential idea of generative models is that, calculating the statistics of $p(X|y)$, $p(y)$ and $p(X)$ from the training set, then using the statistics to simulate the probability density function of the class label. On the other hand, probabilistic models that directly calculate the conditional probability $p(y|X)$ are called discriminative models. Discriminative models generate a set of feature functions to simulate the local structure of the pre-defined patterns in the training set. The contribution of each feature function is bounded by a weight, which is optimised through the training process to simulate the conditional probability distribution. During inference, the similarities between the testing sample and the local feature functions are accumulated into a final score. The difference of the two kinds of models can be illustrated with the following two aspects.

Firstly, for sequence data with long-range dependencies, only the discrimina-

tive models can produce decent recognition rate. Take Part-of-speech Tagging (POS tagging) for example, which is the task of labelling words with different properties, the meaning of the word "record" in the phrase "break the world record" is completely different from its meaning in the phrase "record the speech". In the context of HGR, for hand-signed digits, the gesture "7" and "4" both contain a vertical stroke. If HMM is used for gesture classification, the feature function of the vertical strokes in the gesture class "7" and "4" will have similar weights. That means the HMM model ignores the vertical strokes as trajectory features when classifying samples from the gesture class "7" and "4". But the vertical strokes as a part of the trajectory sequence in the two gesture classes are discriminative. In other words, if taking longer trajectory sequences that contain two or more strokes including the vertical strokes into consideration as the CRF models do, the feature functions of the vertical strokes in both classes will have different weights.

Secondly, the discriminative models do not have to model $p(X)$. When X contains highly interdependent variables, $p(X)$ becomes intractable to calculate. For pattern recognition applications, calculation of $p(X)$ can be avoided in the Naive Bayes model. But in HMM, calculation of the initial state probability is inevitable. The only way of doing it is by counting appearances of different initial states through the training set. In other words, the precision of $p(X)$ is depend on the diversity of the training set. If the training set is too small or not diversified enough, the performance of HMM could be poor.

5.1.2 Lebal Bias Problem in HGR

Discriminative models that monitor the dependencies in the input sequence data, include Maximum Entropy Markov Models (MEMM) [167] suffer from the Lebal Bias Problem which is firstly defined in the original CRF paper [146]. The transitional features leaving a given state only compete against each other, rather than against all other transitional features in the model [146]. Maximum Entropy Taggers [168],

Maximum Entropy Model (MEM) [167] and other non-generative finite state models are all typical victims of this problem [146]. In backward recursion of the MEM classifier, given a current position t, the previous position $t+1$ ($t+1$ is the previous position of t since it is the backward recursion), the observation state x_{t+1} at position $t+1$, the class label y_t and y_{t+1} at position t and $t+1$ respectively, the summation of all possible outputs from position $t+1$ is always 1:

$$\beta_t(i) = \sum_j p(y_{t+1} = j \,|y_t = i, x_{t+1}) \cdot \beta_{t+1}(j) \tag{5.1}$$

where $\beta_t(i)$ indicates the summation of all conditional probabilities of the state value at $t+1$ given the state value at state t. i and j are possible class label values. $\beta_t(i)$ is always 1, no matter what i realy is.

Some of the Generative models also suffer from this problem. Take HMM as an example, given a state t, there are only finite number of choices for the value of the next state, since the state value can only be one of the class labels. Summation of all possible transitional probabilities that leaving from this state must be 1. That indicates the transitional features could not consider the context of the input sequence as a whole.

In the context of HGR, local dependencies are crucial for gesture classification. As shown in Fig 5.2, the two same local features covered by the solid circles in gesture "4" and "2", are represented by feature f_i. The two local features in the doted circles are represented by feature f_j and f_k. In HMM, the feature f_i is a product of multiple transitional probabilities. The actual values of the transitional probabilities are determined by counting the number of training samples that have the features. The calculations for the probabilities of feature f_j and f_k are in the same fashion. Hence, the probability values of the three local structures have no influence on each other whatsoever. The values only depend on the features' frequency of appearance in the training set. On the other hand, for CRF,

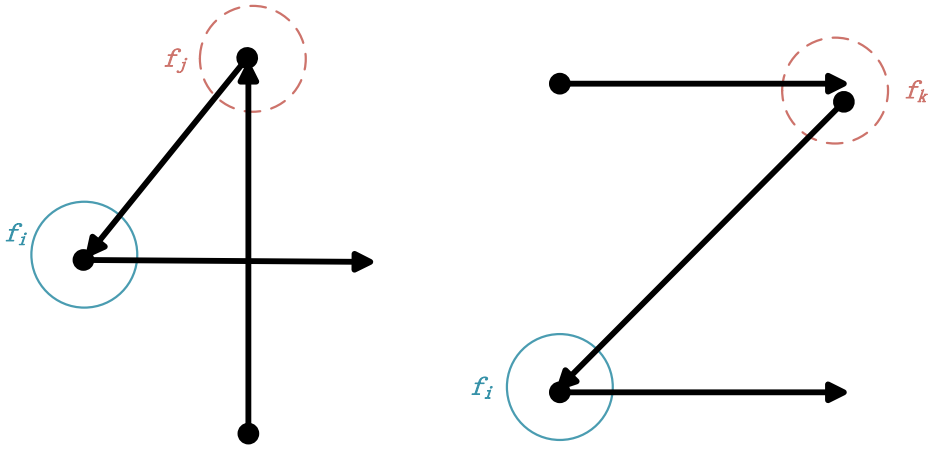

Figure 5.1: The transitional features of gesture "4" and "2". The solid circles indicate one of the common features of the two gesture class, while the doted circles represent two distinctive features that can separate the two classes.

in the training process, the appearance of feature f_i can be found in samples of both gesture "4" and "2". Hence, the feature f_i has relatively small contribution on classifying the two gestures. Thereby the optimisation search would assign a small weight on this feature. For feature f_j and f_k, they are distinctive between the two gestures. Then the two features will be assigned with large weights. In other words, compared with f_i and other less discriminative features, f_j and f_k have certain level of uniqueness for representing gesture "4" and "2" respectively. That makes the two distinctive features have larger weights than the shared features. The CRF is able to evaluate the distinctiveness of the local features, and entitle the effective features with stronger "voting" power by assigning large weights. In this way, the summation of all possible values leaving a specific state is no longer 1 because the summation of the weights is no longer awayls 1. Thereby CRF is capable of solving the Label Bias Problem.

5.2 Gesture Classification for Uncontrolled Environments

The Adaptive SURF Tracking (Section 4.1) is designed to produce multiple trajectories for all hand candidates within one video sample. The next step to complete the task of HGR is gesture classification. In this section, a classifier based on Hidden Conditional Random Fields (HCRF) [148] which is a variation of CRF is proposed for gesture classification in uncontrolled environments. The main task of gesture classification is to detect and classify the trajectory of the target signing hand from the set trajectories of all hand candidates including distractions in the background.

To increase the inter-class distance between the pre-defined gesture trajectories and the noise trajectories, a novel classifier with a weighting scheme is proposed, called Partition Matrix. Essentially this weighting scheme takes tracking results from different ROIs and frame selection patterns into account. The trajectory of the target signing hand has relatively low level variance under different frame rates,

while the background distractions tend to produce various tracking results under different frame rates, due to high probability of out-of-plane rotations, regions overlapping and texture changes.

5.2.1 Gradient based Parameter Estimation

Since Adaptive SURF Tracking uses texture features to track the hand candidates, matching the SURF key points from one frame with different frames may produce different tracking results. In order to take as many tracking results as possible into consideration, all tracking results from different ROIs and frame selection patterns are evaluated in the classifier. The frame selection patterns means down-sampling the original video into different frame rates. Assuming that $V = \{c_i | i = 1, 2, \ldots, N\}$ is a video with N frames, c_n is the n^{th} frame, video fragment V_p with frame rate F_p is defined as,

$$V_p = \{c_i | i = p, 2p, 3p...\} \tag{5.2}$$

The downsampled video fragments are subsets of V. Frame rates F_1 to F_4 are used to create different tracking results in our experiments. After the tracking stage, the set of movement direction vectors $X = \{x_{u,r} | u = 0, \ldots U-1, r = 0, \ldots R-1\}$ are fed into a multi-class HCRF model as the observation sequences (Fig 5.2). Each movement direction vector $x_{u,r} = \{o_0, o_1, \ \ldots \ , o_{l-1}\}$ contains l observation states as l is the number of frames in the video fragment corresponding to $x_{u,r}$ with the r^{th} frame rate. The total amount of the observation sequences equals to the number of frame rates U, times R which is the number of ROIs. In our experiments, $U = 4$ and one single frame is treated as a single node in the HCRF model. $Y = \{y_0, y_1, \ldots, y_{m-1}\}$ is the class label set and m is the total number of the class labels. In this chapter, since the task is recognising hand-signed digits (Fig 5.3), we define the hidden states $H = \{H_0, H_1, \ldots, H_{n-1}\}$ as the strokes of gestures and n is the total number of the

hidden states. Since the stroke structures are natural segmentations of the trajectories, and the hidden states are used to simulate the structure of the trajectory features with high level of similarity, using strokes as hidden states is a reasonable choice. There are certain level of similarities among the gestures (for example gesture 0, 6 and 9), namely some digits share the similar strokes. The number of hidden states are lower than the total number of strokes of the whole vocabulary. For each observation sequence $x_{u,r}$, a vector of hidden states $\vec{h} = \{h_0, h_1, \ \ldots \ , h_{l-1}\}$ is assgined to it. Each element of the hidden state vector $\vec{h}$ is one of the hidden states in H. Also, each element of the hidden state vector $\vec{h}$ is corresponding to an observation state in the observation sequence $x_{u,r}$. Hence the length of the hidden state vector $\vec{h}$ is also l, the same as the observation sequence $x_{u,r}$. The experiments of this chapter contain tests on two datasets, the Warwick Hand Gesture Dataset and Palm Graffiti Digits database [8]. The hidden state definitions are, 11 states in the HCRF model for the Warwick Hand Gesture Dataset (Fig 5.2 shows 4 of the 11 states, which form the gesture of digit 4), and 15 states in the Palm Graffiti Digits database. The optimisation scheme used in our model is Limited Memory Broyden-Fletcher-Goldfarb-Shanno method [169]. In our experiments, the weight vector $\vec{\theta}$ is initialised with the mean value, and the regularisation factors are set to zero.

The HCRF model with hidden latent variables are proven to be effective for gesture recognition [148, 149], the basic structure from the original HCRF model [148] is designed for dealing with one input observation sequence $x_{u,r}$ given a class label y_g, a hidden state vector $\vec{h}$ and the weight vector $\vec{\theta}$:

$$P\left(y_g|x_{u,r},\vec{\theta}\right) = \sum_{\vec{h}} P\left(y_g,\vec{h}|x_{u,r},\vec{\theta}\right) = \frac{\sum\limits_{\vec{h}} \exp\left\{\Psi\left(y_g,\vec{h},x_{u,r}|\vec{\theta}\right)\right\}}{\sum\limits_{y}\sum\limits_{\vec{h}} \exp\left\{\Psi\left(y,\vec{h},x_{u,r}|\vec{\theta}\right)\right\}} \tag{5.3}$$

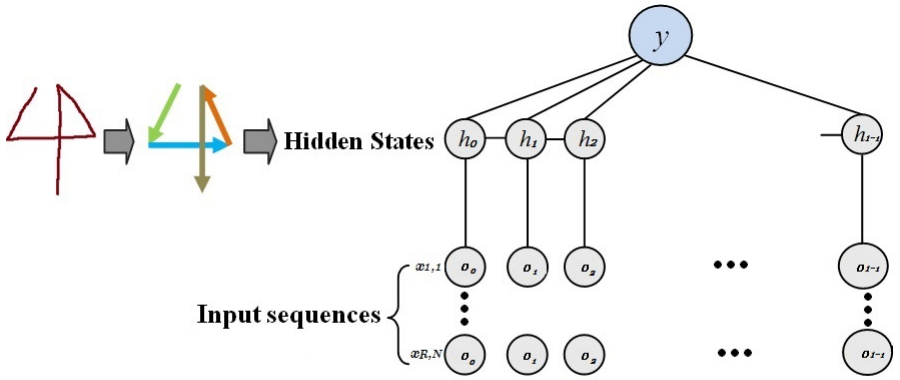

Figure 5.2: HCRF model, the hidden states are defined as strokes of gestures, input sequence x is the movement direction vector of one hand candidate under one frame selection pattern. $x_{u,r}$ means vector with u^{th} frame selection pattern and r^{th} ROI.

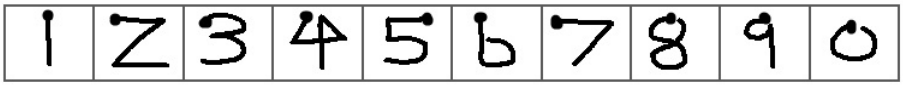

Figure 5.3: Vocabulary of ten hand signed digits.

where the sum over the hidden state vector $\vec{h}$ means the summation of all possible combinations of l hidden states in the hidden state vector. The partition function is defined as:

$$Z\left(y_g|x_{u,r},\vec{\theta}\right)=\sum_{\vec{h}}\exp\left\{\Psi\left(y_g,\vec{h},x_{u,r}|\vec{\theta}\right)\right\} \tag{5.4}$$

The partition function is the sum of all potential functions $\Psi\left(y_g,\vec{h},x_{u,r}|\vec{\theta}\right)$ over all possible combinations of hidden states in the hidden state vector. $\vec{\theta}$ is the set of weights of the potential functions. The potential function with a given class label y_g, a hidden state vector $\vec{h}$, a input observation sequence $x_{u,r}$ and the set of weights $\vec{\theta}$ is defined as:

$$\begin{aligned}\Psi\left(y_g,\vec{h},x_{u,r}|\vec{\theta}\right)=&\sum_{j=0}^{l-1}\sum_{i=0}^{D\times n}\theta_{1,i}\cdot f_{1,i}\left(x_{u,r},h_j\right)+\sum_{j=0}^{l-1}\sum_{i=0}^{m\times n}\theta_{2,i}\cdot f_{2,i}\left(y_g,h_j\right)+\\&\sum_{(j,k)\in E}\sum_{i=0}^{m\times n^2}\theta_{3,i}\cdot f_{3,i}\left(y_g,h_j,h_k\right)\end{aligned} \tag{5.5}$$

where j and k are hidden state index, E is the set of adjacent hidden states in the hidden state vector $\vec{h}$. $f_{1,i}$ and $\theta_{1,i}$ are the i^{th} feature function and the corresponding weight of the first feature function type. The same rule of nomenclature applies to $f_{2,i}$,$f_{3,i}$, $\theta_{2,i}$ and $\theta_{3,i}$. For each input sequence, the value of the potential function is the weighted sum of all feature functions. Three types of feature functions and their corresponding weights are defined in the proposed model of this chapter. The first feature function type is the compatibility between an observation sequence $x_{u,r}$ and a hidden state h_j, $f_{1,i}\left(x_{u,r},h_j\right)$,

$$\begin{aligned}f_{1,i}&=\left\{f_{1,i}\,|i\in[1,L_1]\right\}\\\theta_{1,i}&=\left\{\theta_{1,i}\,|i\in[1,L_1]\right\}\end{aligned} \tag{5.6}$$

where $L_1 = D \times n$ is the total number of the first type of feature function. D is the number of possible movement orientations (see Eq 4.18) and n is the total number of hidden states as defined earlier in this chapter. This kind of feature function can be understood as the probability of the observation being part of a certain stroke represented by a hidden state (Fig 5.4). To monitor the long-range dependencies, the neighbouring observations of the current observation state are also taken into calculation of the feature functions. The window size w indicating how many neighbouring observations are included in the feature function. Assuming the set of neighbouring observations of the current observation state o_c is W,

$$W = \{o_{c-w}, \ldots, o_{c-1}, o_c, o_{c+1}, \ldots, o_{c+w}\} \tag{5.7}$$

If $w = 3$, that means three previous and three future neighbour observations are in the feature window, so there are 7 observations in W.

The second type feature function $f_{2,i}(y_g, h_j)$ is the compatibility between a hidden state h_j and a class label y_g. There are in total L_2 feature functions of the second type,

$$\begin{aligned} f_{2,i} &= \{f_{2,i} \,|i \in [1, L_2]\} \\ \theta_{2,i} &= \{\theta_{2,i} \,|i \in [1, L_2]\} \end{aligned} \tag{5.8}$$

where $L_2 = m \times n$ is the total number of the second type of feature function. m is the total number of classes in the vocabulary as defined earlier in this chapter. This kind of feature function can be seen as the probability of the local structure of a certain gesture class containing a specific stroke (Fig 5.5).

The third kind is the compatibility of a pair of adjacent hidden states h_j and h_k with a class label y_g: $f_{3,i}(y_g, h_j, h_k)$. There are in total L_3 different feature functions of the third kind,

$$f_{3,i} = \{f_{3,i} \,|i \in [1, L_3]\}$$
$$\theta_{3,i} = \{\theta_{3,i} \,|i \in [1, L_3]\} \tag{5.9}$$

where $L_3 = m \times n^2$ is the total number of the third type of feature function. This feature type is similar to the transitional probability in the HMM classifier. The features describe the probability of certain gesture class containing certain local trajectory patterns represented by local trajectory feature sub-sequences (Fig 5.6).

Since every single feature function takes all frames in the input sequence into calculation, to make each feature function extracts a certain local pattern from the input sequence, the feature functions are designed to produce non-zero values for only this pattern in the model. Take one of the first type feature functions as an example:

$$f_{1,i}\left(x_{u,r}, h_k\right) = \begin{cases} 1, & \exists o_j : o_j = q_{12} \; and \; o_j \in x_{u,r} \;,\; k = 3 \\ 0, & otherwise \end{cases} \tag{5.10}$$

where i is the index of this specific feature function. This feature function only produce value 1 when the 12^{th} movement orientation q_{12} (Eq 4.18) appears in the observation sequence $x_{u,r}$ as an observation state, and the corresponding hidden state of q_{12} is h_3. In this way, the feature functions are not depending on specific positions in the observation sequence. In other words, even if the target local pattern appears at arbitrary stage of the gesture, the feature function can still detect this local pattern. This makes the model more tolerant to distortions in the hand trajectory. As shown in Fig 5.7, intra-class variance could be relatively large in HGR problems, that means gesture distortions appear frequently in both the training set and the testing set of HGR databases. This feature extraction strategy could be sensitive for trajectories of background distractions that share high level similarity

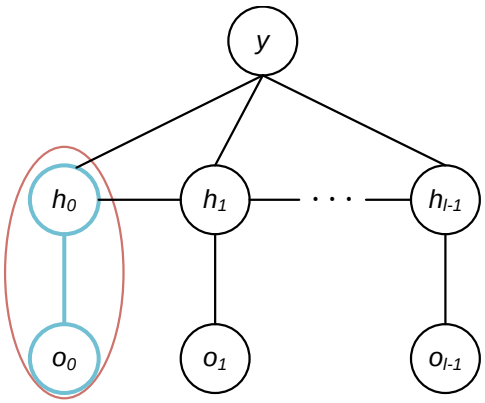

Figure 5.4: The feature function contains the observation state and corresponding hidden state.

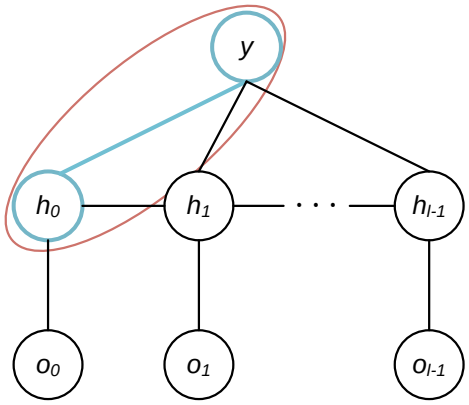

Figure 5.5: The feature function contains the hidden state and the class label.

with pre-defined gestures. For HGR in uncontrolled environments, the inter-class distance between the noise patterns and pre-defined gestures are still averagely larger than intra-class variance within the gesture classes. Hence the proposed framework benefits from the position independent feature functions.

The learning process of the proposed framework is essentially estimation of the parameter set θ, which represents the "voting power" for each and every feature function. Therefore, the training process is the search for the optimised voting power distribution for all feature functions. As all other CRF-like probabilistic models, the parameter estimation of the proposed framework is done through a gradient based

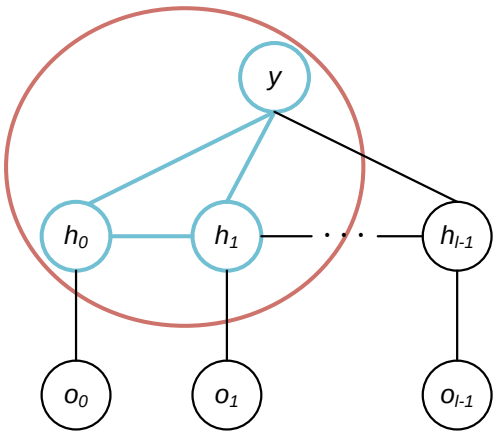

Figure 5.6: The feature function of the transitional hidden states and the class label. In this case, the window size is 0.

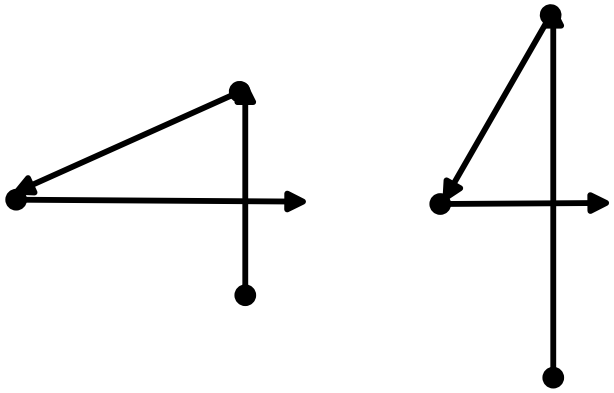

Figure 5.7: Samples of distorted hand trajectories.

search, to be more specific, through maximising the likelihood function $L(\theta)$ of the training set T_r:

$$L\left(\vec{\theta}\right) = \sum_{(X,y)\in T_r} \log P\left(y \left| X, \vec{\theta}\right.\right) - \frac{1}{2\sigma^2}\left\|\vec{\theta}\right\|^2 \tag{5.11}$$

where (X, y) is a observation-label pair with the set of input observation sequences extracted from a training sample and the ground truth class label for this training sample. The second term in the likelihood function is a penalty term. The reason to include this penalty term is to avoid overfitting. The idea is introducing a regularisation term to prevent a few feature functions to possess dominant weights. The common choice of the penalty is based on the Euclidean norm of the weight vector $\vec{\theta}$ [149, 148, 170, 171]. The actual strength of the penalty is determined by a regularisation parameter $\frac{1}{2\sigma^2}$. σ is a Gaussian prior variance of the assumed Gaussian distribution of $\vec{\theta}$, namely

$$P\left(\vec{\theta}\right) \sim \exp\left(-\frac{1}{2\sigma^2}\left\|\vec{\theta}\right\|^2\right) \tag{5.12}$$

The search for the optimised value of σ could be computation intensive. Theory of Charles Sutton [170] indicates that the accuracy of the CRF model is not sensitive to the precision of σ. For databases with medium-sized training sets, $\sigma^2 = 10$ is typical. The proposed framework inherited this value of σ. That is reasonable strategy for HGR problems, since there are normally only few hundreds of training samples in the databases. Intuitively, the scale of the bias towards the noises in the training set produced by the optimisation search depends on the amount of distorted training samples. Value of the variance σ represents the extent of domination of biased feature functions. The larger the variance σ is, the more significant the bias is, thereby the larger the penalty should be.

Given the likelihood function, the training process becomes a parameter optimisation problem with the likelihood function $L\left(\vec{\theta}\right)$ as the objective function. For

functions with one dimensional variable, the optimisation is naturally the process of calculating the first order derivative of the variable, then equating the function to zero and solving the equation. But for pattern recognition applications including HGR, the variable is usually with high dimensions. Calculating the derivatives becomes intractable. For the weight vector $\vec{\theta}$ in this thesis which is about 1800 dimensional vector depend on different window sizes of the feature functions in the HCRF model, the optimisation turns from a simple line search to a search in a high dimensional space.

Many gradient based methods could potentially be viable solutions, including Line Search, Simulated Annealing, Gradient Ascent, Conjugate Gradient, Levenberg Marquart and Newton's Method. In this thesis, Limited Memory BFGS algorithm is used as the optimisation method [172]. The fundamental idea of all optimisation methods is simple. In order to find a maximum or minimum value of a certain function, which could be bound by additional constrains, a iterative search is conducted. There are few strategies to initialise the start point of the search, such as starting at mean point or random point. In every iteration, the algorithm calculates the next search direction. The common strategy is to follow the gradient. Then the algorithm finds the local extreme value along the calculated search direction and evaluates this extreme value to see whether it fits the converge criteria. If not, the search continues to the next iteration, until it locates a value that fits the converge criteria. By simply following the gradient, the search usually takes too many iterations to converge. Hence, other strategies of calculating more efficient search directions were considered. The well-known Newton's method converge much faster than the original gradient descent methods. Since they use not only the gradient which is the first order derivative, the Hessian Matrix which essentially is a matrix of second order derivatives is also used to calculate the search directions. For objective functions with high dimensional variables, the calculation of the Hessian Matrix could be computation intensive, since the second order derivative of every

dimension must be calculated. The Quasi-Newton methods are optimisation methods that do not calculate the Hessian Matrix in every search step directly. Instead, an approximation of the Hessian Matrix is built to speed up the process. The Limited Memory BFGS method is essentially one of the Quasi-Newton methods. The uniqueness of the Limited Memory BFGS method is that it only uses the first order derivative to estimate the Hessian Matrix of the objective function. For HGR in the uncontrolled environments, the number of feature functions is huge. That makes the fast converge speed an important feature to consider for choosing the optimisation method. Hence, the Limited Memory BFGS is the best choice.

To use the optimisation methods, the obvious issue is that, is the objective function namely the likelihood function of the training set $L\left(\vec{\theta}\right)$ even convex? For normal CRF model without the hidden latent states, each observation state has a corresponding class label. Given the training set T_r, the likelihood function can be written as,

$$
\begin{aligned}
L\left(\vec{\theta}\right) &= \sum_{(X,y)\in T_r} \log P\left(y \left| X, \vec{\theta}\right.\right) - \frac{1}{2\sigma^2}\left\|\vec{\theta}\right\|^2 \\
&= \sum_{(X,y)\in T_r} \sum_{j=1}^{3} \sum_{i=1}^{L_j} \theta_{j,i} \cdot f_{j,i} - \sum_{(X,y)\in T_r} \log \sum_{\vec{y}} \exp\left[\sum_{j=1}^{3} \sum_{i=1}^{L_j} \theta_{j,i} \cdot f_{j,i}\right] - \\
&\quad \frac{1}{2\sigma^2}\left\|\vec{\theta}\right\|^2
\end{aligned}
\tag{5.13}
$$

where j is the feature function type index and i is the feature function index as defined before. The sum over $\vec{y}$ is the sum of all possible combinations of the class label vector for the observation sequence. The first term in the equation on the second line is three fold summation of the weighted feature functions. The summation over weight $\vec{\theta}$ is linear function. The second term is the normalisation term. It is a Log-Sum-Exp (LSE) Function, which is a known convex function

[173]. The last term, namely the penalty function, is a quadratic function of the weight $\vec{\theta}$. The whole equation is a linear combination of a linear function, a convex function and a quadratic function. Therefore, the likelihood function of normal chain-structured CRF model is guaranteed to be convex.

For CRF models with hidden latent states, the likelihood function can be written as,

$$L\left(\vec{\theta}\right) = \sum_{(X,y)\in T_r} \log P\left(y \left| X, \vec{\theta}\right.\right) - \frac{1}{2\sigma^2}\left\|\vec{\theta}\right\|^2$$

$$= \sum_{(X,y)\in T_r} \log \sum_{\vec{h}} \exp\left(\sum_{j=1}^{3}\sum_{i=1}^{L_j} \theta_{j,i}\cdot f_{j,i}\right) - \sum_{(X,y)\in T_r} \log \sum_{y'\in Y}\sum_{\vec{h}} \exp\left(\sum_{j=1}^{3}\sum_{i=1}^{L_j} \theta_{j,i}\cdot f_{j,i}\right) - \frac{1}{2\sigma^2}\left\|\vec{\theta}\right\|^2 \tag{5.14}$$

where $\sum_{(y'\in Y)}$ and $\sum_{\vec{h}}$ means sum over all class labels and all combination of hidden states in the hidden state vector in the feature functions, as defined in Eq 5.5. The right hand side of the second line equation is essentially a difference of two LSE functions, which is not always convex. The reason for the loss of convexity is the variation of partition function (Eq 5.4) in HCRF models. Since the partition function is the summation over the hidden states, so the first term in the likelihood function is no longer linear. In the context of the proposed framework, the loss of convexity of the likelihood function is one of the main drawbacks of hidden state CRF models. This means the optimisation search is only capable of locating the closest local optimisation, not the global optimisation. Therefore, the starting point of the optimisation search is vital to the training process. In the proposed framework, the initialisation value of the weight vector is the mean point.

Another important concept of the proposed framework is the definition of

the partition function (Eq 5.4). Using the partition function can largely speed up the training and inference processes. To evaluate the status of convergence of the optimisation search during the training process, in every iteration the Limited Memory BFGS algorithm calculates the difference between the current value of the likelihood function, and the value of the likelihood function with the previous weight vector from the last iteration. The value of the likelihood function can be calculated as,

$$L\left(\vec{\theta}\right) = \sum_{(X,y)\in T_r} \log P\left(y\left|X,\vec{\theta}\right.\right) - \frac{1}{2\sigma^2}\left\|\vec{\theta}\right\|^2 \tag{5.15}$$

$$= \sum_{(X,y)\in T_r} \log\left(\frac{Z\left(y|X,\vec{\theta}\right)}{\sum\limits_{y'\in Y} Z\left(y'|X,\vec{\theta}\right)}\right) - \frac{1}{2\sigma^2}\left\|\vec{\theta}\right\|^2$$

The inference process can also be simplified by only calculating the partition function. Since the final score of input sequence set X against class label y_g can be written as,

$$P\left(y_g|X,\vec{\theta}\right) = \sum_{\vec{h}} P\left(y_g,\vec{h}|X,\vec{\theta}\right) = \frac{\sum\limits_{\vec{h}} \exp\left\{\Psi\left(y_g,\vec{h},X;\vec{\theta}\right)\right\}}{\sum\limits_{y}\sum\limits_{\vec{h}} \exp\left\{\Psi\left(y,\vec{h},X;\vec{\theta}\right)\right\}} = \frac{Z\left(y_g\left|X,\vec{\theta}\right.\right)}{\sum\limits_{y} Z\left(y\left|X,\vec{\theta}\right.\right)} \tag{5.16}$$

The denominator $\sum\limits_{y} Z\left(y\left|X,\vec{\theta}\right.\right)$ is summation of the partition functions over all class labels, which can be treated as a normalisation term. Therefore, for classification which is finding the largest probability $P\left(y_g|X,\vec{\theta}\right)$ where $y_g \in Y$ among all class labels, if the normalisation term is ignored for partition functions of all class label, the value of the partition function is equivalent to the final score y^*:

$$y^* = \arg\max_y P\left(y|X,\vec{\theta}\right) \quad = \quad \arg\max_y \frac{Z\left(y|X,\vec{\theta}\right)}{\sum_{y'\in Y} Z\left(y'|X,\vec{\theta}\right)} \quad \approx \quad \arg\max_y Z\left(y\,\middle|X,\vec{\theta}\right) \tag{5.17}$$

This largely simplified the inference process has significant contribution to the overall real-time performance of the entire proposed framework. Also, the idea of the proposed weighting scheme Partition Matrix is built based on this equivalence (see section 5.2.3).

The training process indicates that the CRF models are supervised classifiers. But for the hidden latent states, training process can be seen as semi-supervised learning. For every training sample, although the ground truth class label is provided, the hidden states that corresponding to all single observation states are unlabelled. In other words, we do not need to provide the model with any information on the hidden state sequences of every training sample, expect the total number of the hidden states. Similar to the parameter estimation of HMM, the model will calculate the optimised configuration of the weight vector $\vec{\theta}$ automatically. Intuitively, this semi-supervised training strategy for the hidden states is essentially a feature selection process. The model is capable of prioritising the "voting power" among the feature functions with hidden states. Latent Dynamic CRF [150], is another version of CRF. In the LDCRF, the hidden state sequences of every training sample are labelled. That makes the training for hidden state completely supervised. This strategy is also proven to be effective for gesture recognition [150].

5.2.2 Inference with Partition Matrix

The proposed framework takes the idea of the Partition Function from the original CRF model one step forward to solve the gesture classification problem in uncontrolled environments with multiple hand candidates in the scene. More specifically, a weighting matrix of the partition functions from different frame rate and ROIs is

proposed.

When a new video clip comes in for classification, all vectors in X will be evaluated against every gesture classes. A normalisation process is applied on all partition functions of X. The reason for the normalisation is that, observation sequences with different frame rates have various sizes. The observation sequence with frame rate F_3 has only one third amount of frames as the sequence with frame rate F_1. Since the value of partition function is proportional to the amount of feature functions, and the amount of feature functions depends on the amount of frames in the observation sequence. Hence, the trajectory of background distractions with large amount of frames could potentially produce higher partition function values than the trajectories of the target signing hand, solely due to the large number of feature functions, instead of high level of similarity with pre-defined trajectories. Therefore, in order to compare partition functions of observation sequences with different sizes, partial partition function for single observation sequence is proposed:

$$Z^n\left(y_g \left| x_{u,r}, \vec{\theta}\right.\right) = \sum_{\vec{h}} \exp\left[\Psi\left(y_g, \vec{h}, x_{u,r}|\vec{\theta}\right)\right] \Big/ n_{u,r} \tag{5.18}$$

where $n_{u,r}$ is the total number of frames in $x_{u,r}$ and y_g is a given class label. Intuitively, the partition function is normalised to average partition value per frame. Since the only trajectory feature used in the proposed framework is the movement direction, the size and location invariance are naturally achieved (Section 4.2). Also the Partition Matrix does not depend on the length of the observation sequences, which makes the proposed framework robust against the changing gesture speed.

Since the basic idea of the Partition Matrix is to take trajectories of multiple hand candidates with different frame rates into consideration, all eligible hand candidates including background distractions in the uncontrolled environments are under evaluation. Even for the same ROI, with different frame rates, the Adaptive SURF Tracking scheme can produce largely different tracking results, due to the

texture matching nature of the tracking scheme and the random moving patterns of the hand candidates. Hence, structure of the Partition Matrix of observation sequence set X is defined as,

$$\begin{pmatrix} \left(Z^p_{0,0}, L_{0,0}\right) & \cdots & \left(Z^p_{0,R-1}, L_{0,R-1}\right) \\ \vdots & \ddots & \vdots \\ \left(Z^p_{U-1,0}, L_{U-1,0}\right) & \cdots & \left(Z^p_{U-1,R-1}, L_{U-1,R-1}\right) \end{pmatrix} \tag{5.19}$$

where $(Z^p_{u,r}, L_{u,r})$ is the partition-label pair of the observation sequence of r^{th} hand candidate with u^{th} frame rate. Among all partition functions of the observation sequence $x_{u,r}$ against all class labels, $L_{u,r}$ is the class label with the highest partition function value, and $Z^p_{u,r}$ is the corresponding partition value.

$$L_{u,r} = \arg\max_{y \in Y} Z^n \left(y | x_{u,r}, \vec{\theta}\right) \tag{5.20}$$

$$Z^p_{u,r} = \max_{y \in Y} \left[Z^n \left(y | x_{u,r}, \vec{\theta}\right) \right] \tag{5.21}$$

Every column of the matrix contains observation sequences extracted from one of the hand candidates, with various frame rates, while each row of the matrix contains observation sequences extracted from all hand candidates, with the same frame rate. With the definition of the Partition Matrix, partition function for the observation sequence set X and a given class label y_g is proposed:

$$Z'\left(y_g | X, \vec{\theta}\right) = \sum_{x_{u,r} \in X} \left\{ Z^n \left(y_g | x_{u,r}, \vec{\theta}\right) \cdot w_{u,r} \right\} \tag{5.22}$$

The partition function of X is essentially a weighted sum of partial partition functions of all observation sequences in X. How to distinct the trajectory of the target signing hand from the other trajectories depends on the definition of the weight $w_{u,r}$, which is,

$$w_{u,r} = 1 + W_F\left(x_{u,r}\right) + W_R\left(x_{u,r}\right) \tag{5.23}$$

where $W_F\left(x_{u,r}\right)$ is the Frame Rate Weight Function, which gives a larger weight to the observation sequence with maximum $P\left(y|x_{u,r},\vec{\theta}\right)$ value among all ROIs with the same frame rate, namely a row in the Partition Matrix. The rationale behind this is that, for trajectories from all hand candidates under the same frame rate, the one with the highest partition value has the highest possibility to be the target sighing hand. Hence, more confidence should be assigned to the corresponding ROI. The definition of Frame Rate Weight is,

$$W_F\left(x_{u,r}\right) = \begin{cases} 1/U, & x_{u,r} = \underset{\bar{x}_{u,r}\in\left\{\bar{x}_{u',r}|u'=u\right\}}{\arg\max} P\left(y|\bar{x}_{u,r},\vec{\theta}\right) \\ 0, & otherwise \end{cases} \tag{5.24}$$

where U is the total number of frame rates that used in the Partition Matrix. The definition of $W_F\left(x_{u,r}\right)$ is based on the logic that the more frame rate adopted in the Partition Matrix, the less voting power should be assigned to each row of the matrix.

$W_R\left(x_{u,r}\right)$ is the ROI Weight function, which represents the confidence of the r^{th} ROI being the target hand, considering all frame rates. The definition is,

$$W_R\left(x_{u,r}\right) = \left|\left\{x_{u',r'}|W_F\left(x_{u',r'}\right) \neq 0, r' = r\right\}\right|/U \tag{5.25}$$

The ROI Weight of a specific ROI depends on how many row maximum value are there among the observation sequences from this ROI, namely a column of the Partition Matrix. The ROI Weight is designed to enhance the voting power of the ROI with the target hand. For the situation shown in Fig 5.8, more than one ROIs have row maximum values, the ROI Weight can further enlarge the difference of voting

power among ROIs. At last, the final gesture label is defined as $\arg\max_y Z'\left(y|X,\vec{\theta}\right)$. Namely, the gesture class with the largest weighted sum of the partition values is the final result.

To explain how Partition Matrix analyse trajectories from uncontrolled environments, an example is shown in Fig 5.8. In this example, the tracking scheme picked up 4 hand candidates, and the target hand is signing gesture "7" while the background distractions are randomly moving around. As shown in the figure, classification results for the first 2 frame rates are wrong. Namely without using the Partition Matrix, the final classification would be wrong. This example constitutes the best scenario that the Partition Matrix can distinguish the trajectory of the target hand from other trajectories of distractions with the similar scale, speed and skin-colour as the target hand. In the sample Partition Matrix, to better explain the weighting strategy, only the class labels are shown.

As shown in Fig 5.8, on the row of all ROIs with frame rate F_4, all 4 classification results are gesture "7". That indicates the possibility that all 4 ROIs overlapped at some point during the gesture, so that all four trajectories share the same level of similarity with the gesture model "7". ROI overlapping happens frequently in the testing samples of the databases used in this thesis. The high probability of region overlapping is caused by two reasons. Firstly, since the precision of hand candidates selection is not at high level by design (Section 4.1.1), some of the ROIs may occupy relatively large areas in the scene. That could lead to overlapping of ROIs. Since SURF features are extracted in every frame, when ROIs are overlapping, the textures of the ROI on the background are occluded. Hence, the overlapping ROIs extract the same texture features during the overlapping. In other words, they emerge into a single ROI. Secondly, with lower frame rates, the key point displacements between adjacent frames are relatively larger than that of the higher frame rates. The change of textures of the background distractions with random movements between frames are also more dramatically in the videos with

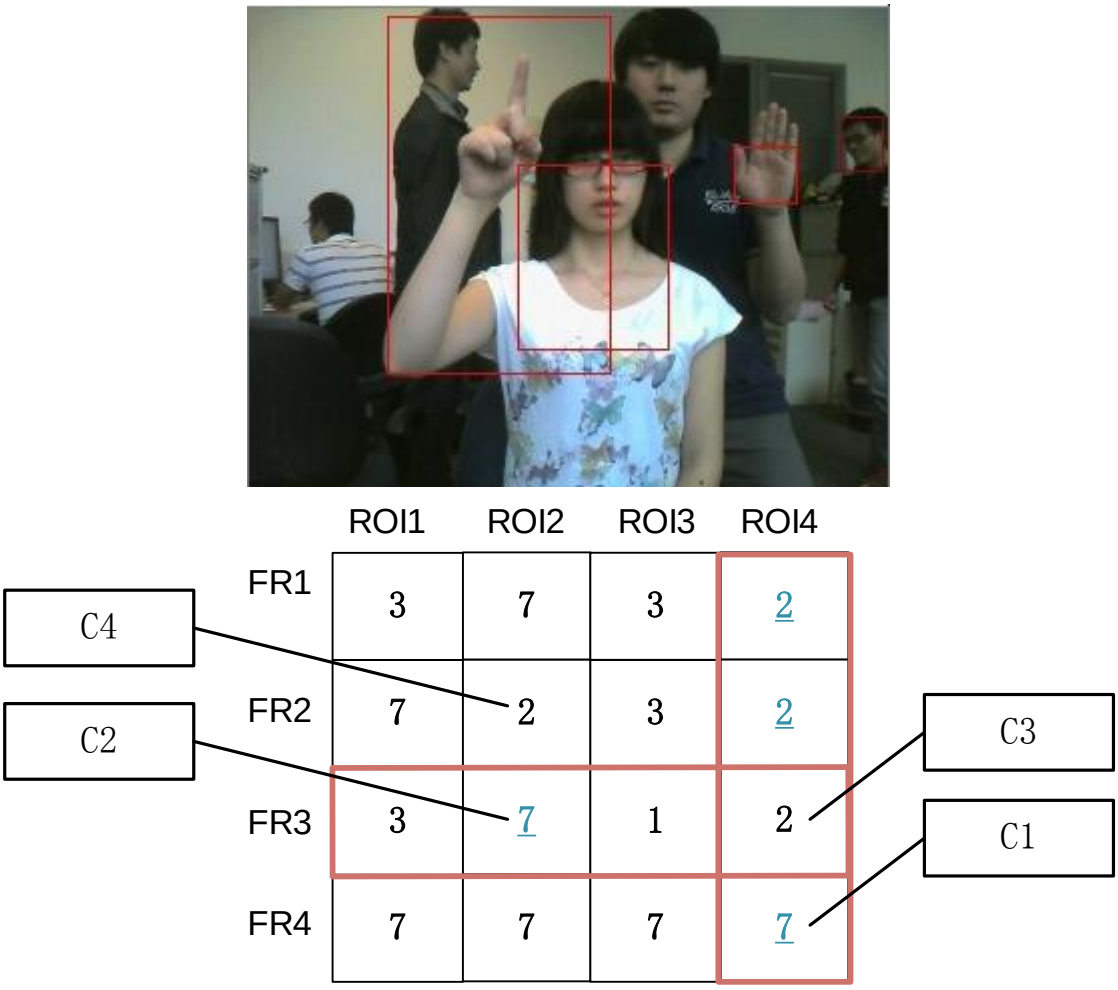

Figure 5.8: Partition Matrix of a testing sample from the Warwick Hand Gesture Database. The target hand is signing the gesture "7", while the background distractions are randomly moving around. The Partition Matrix is only showing the class labels, instead of the partition-label pairs.

lower frame rates. That leads to poor texture matching results, namely small number of matching key point pairs. When the paths of ROIs are closing together, even if the hand candidates are not overlapping, the ROIs are highly likely to overlap at low frame rates.

The Partition Matrix can use the frequent appearances of one class label due to the high probability of ROI overlapping to analyse the gesture trajectories. Frequent appearances of one class label in the Partition Matrix happens for two reasons. One is multiple hand candidates are signing this gesture. If the gesture is the ground truth gesture, this gesture will win the final score by a large margin. If the gesture is not the ground truth, and the partition value of the target hand is smaller than the partition values of the distractions, there is a possibility that this sample could be misclassified. The other possible reason is one of the hand candidates in the foreground is signing this gesture, and the trajectory of this hand candidate is large enough to overlap with others. By the nature of HGR applications, gesture performer is assumed in the foreground with relatively large scale. Hence, the hand candidate causes region overlapping is likely to be the target hand. Therefore, the frequent appearances of one class label in the matrix indicating at least one hand candidates in the foreground with relatively large area and small change of texture, are signing this gesture.

The cells with underlined class labels in the Partition Matrix represent the row maximums. The column of ROI2 and ROI4 have row maximums. The reason for not discarding the columns that do not have any row maximums is to keep the influence of ROI overlapping. As explained, the gestures class that repeatedly appearing in the Partition Matrix have high possibility to be the ground truth. Now, the problem is how to distinguish the ground truth from the others.

The cells in the Partition Matrix could be categorised into four types. As shown in Fig 5.8, the examples of the four types are labelled C1, C2, C3 and C4. C1 cells are the row maximums that located in the ROI with the largest number

of row maximums. These cells represent the partition values of the ROI that most likely to be the target hand. Also, C1 cell is the trajectory that has the highest similarity with the pre-defined gestures among all trajectories under the same frame rate. Hence, these cells should be assigned with large voting powers. For C1 cell(4,4) in the example, the weights are $W_F(x_{u,r}) = 1/4$ and $W_R(x_{u,r}) = 3/4$.

C2 cells are row maximums that are not located in the ROI with the largest number of row maximums. For example, the cell labelled C2 in Fig 5.7, it has the right classification result, but it is not from the target hand. That means in this specific frame rate, one of the background distractions has higher similarity with the gesture classes than the target hand. This could be caused by region overlapping or texture mismatching. For situation in the example, the weights are: $W_F(x_{u,r}) = 1/4$ and $W_R(x_{u,r}) = 1/4$. Intuitively, the cell is not as creditable as the C1 cells, because the ROI it came from has low credibility.

For C3 cells, they are not row maximums and they located in the ROI with the largest number of row maximum. C3 cells are possibly caused by the mis-tracked target hand under this particular frame rate. Hence for this situation, the voting power of C3 cells should be lower than C1 and C2 cells. The weights for cells(3,2) are $W_F(x_{u,r}) = 0$ and $W_R(x_{u,r}) = 3/4$.

Cell(2,2) in the example is a C4 cell. It is neither a row maximum, nor located in the ROI with most of the row maximums. Intuitively, some credibility should be assigned to this cell, since the ROI it came from once produced a row maximum. It is possible that the path of this ROI is close to the target hand trajectory enough to cause overlapping. Therefore, it is possible that other trajectories of this ROI under different frame rates also are overlapping with the target hand. Hence the weights are, $W_F(x_{u,r}) = 0$ and $W_R(x_{u,r}) = 1/4$.

For HGR databases, usually there are only few hundred of training samples. This size of training set is far from satisfactory for probabilistic models. Since the models are usually under-trained, and the intra-class variances are relatively high

in HGR problems, decent recognition rate can hardly be produced by probabilistic models. Partition Matrix provides a way to make use of the under-trained model, by providing as much tracking information as possible to the model. For HGR in the uncontrolled environments with multiple background distractions, the Partition Matrix also provides a novel method of analysing the trajectories of all hand candidates and distinguish the trajectory of the target hand from the distractions.

5.2.3 Experiments

Two experiments are conducted on two databases for testing the proposed framework. The first one is on the Palm Graffiti Digits Database used in [8]. All video samples have 240×320 pixels resolution. There are in total 300 isolated gesture training samples, collected from 10 gesture performers. The gesture performers also wear coloured gloves to label the ground truth hand positions. In the training process, ten percent of the training samples are randomly selected to be the validation set, while the rest of the training set is used to train the model. There are two testing sets, the "easy" and "hard" sets. Video clips in the easy set do not have any moving objects in the background. The videos in the hard set have 1-3 people moving in the background (Fig 5.9). Among the challenges discussed in Chapter 1, the ones that are included in this dataset are: 1. Background distractions. 2. Hand/Face overlapping.

In this experiment, the proposed framework outperformed state-of-the-art methods on the Palm Graffiti Digits Database. Correa et al. RoboCup 2009 [9] proposed a cascade of boosted Bayes classifiers with hand's positions and velocities as temporal featuers. Malgireddy et al. CIA 2011 [10] introduced a method to learn the underlying sub-gesture relationships among the predefined signs in the vocabulary, by sharing the parameters of trained generative models. Alon et al. PAMI 2009 [8] proposed a method based on DTW which is capable of perform sub-gesture reasoning, pruning off poor trajectory matches in a early stage and

Figure 5.9: Sample from the hard testing set of the Palm Graffiti Digits database.

processing multiple hand candidates in every frame. The method from Bao et al. ICEICE 2011 [2], is the only existing method similar to the proposed framework. This method also uses SURF as the texture matching spacial feature and a clustering method based on correlation analysis is proposed for trajectory classification. Our method extends the idea of [2] for uncontrolled environments where multiple hand candidates are in the scene. Hence it is implemented and tested in this experiments. Table 5.1 and 5.2 shows the results of the proposed framework on both easy and hard test sets. The comparisons of performances are shown in Table 5.3, Fig 5.10 and 5.11. The reason for the performance drop on the gesture "6" is because the absence of sub-gesture reasoning in our method (discussed in section 6.3).

In the method of Alon et al. PAMI 2009 [8], the amount of hand candidates

Table 5.1: Performance of the proposed framework on the hard set of the Palm Graffiti Digits database.

Gesture Classes	Hard Set				
	Training Samples	Testing Samples	Recognition Results		
			True	Detected	Accuracy (%)
0	30	14	11	15	78.57
1	30	14	13	13	92.86
2	30	14	13	15	92.86
3	30	14	13	14	92.86
4	30	14	13	14	92.86
5	30	14	14	16	100.00
6	30	14	6	6	42.86
7	30	14	12	13	85.71
8	30	14	13	16	92.86
9	30	14	13	18	92.86
Overall	300	140	121	140	86.43

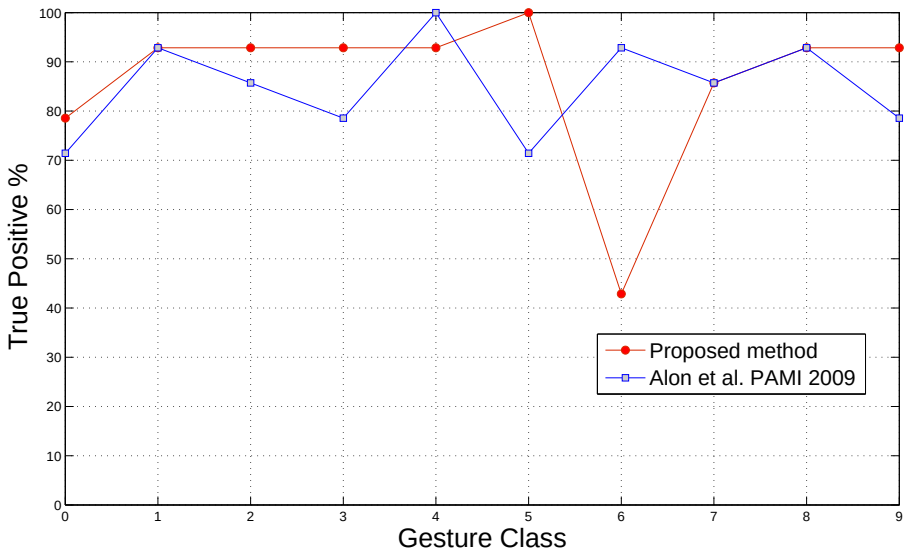

Figure 5.10: Comparison with Alon et al. PAMI 2009 [8] on the hard set of Palm Graffiti Digits database.

Table 5.2: Performance of the proposed framework on the easy set of the Palm Graffiti Digits database.

Gesture Classes	Easy Set				
	Training Samples	Testing Samples	Recognition Results		
			True	Detected	Accuracy (%)
0	30	30	30	35	100.00
1	30	30	30	31	100.00
2	30	30	28	28	93.33
3	30	30	28	28	93.33
4	30	30	30	30	100.00
5	30	30	30	32	100.00
6	30	30	24	25	80.00
7	30	30	29	30	96.67
8	30	30	30	30	100.00
9	30	30	27	31	90.00
Overall	300	300	286	300	95.33

Table 5.3: Comparison with state-of-the-art accuracies on the Palm Graffiti Digits database.

Palm Graffiti Digits Database		
	Easy Set	Hard Set
Correa et al. RoboCup 2009 [9]	75.00%	N/A
Malgireddy et al. CIA 2011 [10]	93.33%	N/A
Alon et al. PAMI 2009 [8]	94.60%	85.00%
Bao et al. ICEICE 2011 [2]	52.00%	28.57%
The proposed method	95.33%	86.43%

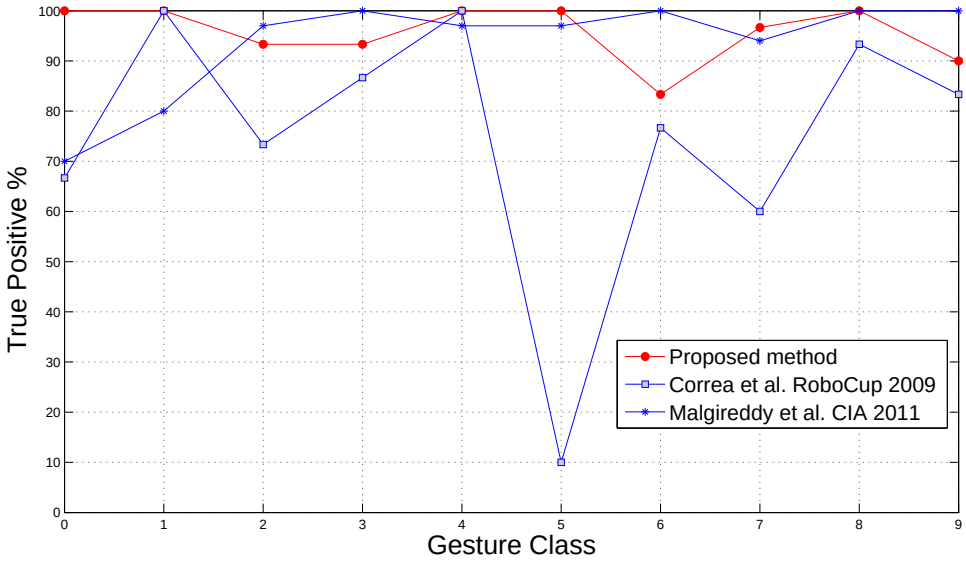

Figure 5.11: Comparison with Correa et al. RoboCup 2009 [9] and Malgireddy et al. CIA 2011 [10] on the easy set of Palm Graffiti Digits database.

must be specified, while the proposed framework does not make any assumptions or require any prior knowledge on the content of the background. Also, extra computation on estimating the location and scale of the gestures is required in [8], while the proposed framework achieved scale, speed and location invariance without any extra computation.

The method of [8] does not allow the observation states to be skipped. Also, the transition probabilities of the states are not used for classification. The Partition Matrix in the proposed framework uses transition probabilities of the hidden states as one of the three feature functions in the potential function, that is another reason why our method outperformed [8]. Since the distractions in the background are moving randomly, out of plane rotation, changing of speed and overlapping of objects are also involved. Hence the texture of the background is changing rapidly. Although there are no constrains on how subjects should perform the gestures, the gesture performers tend to remain in a relatively stationary position. That makes the changing of texture on the gesture performer in a relatively small scale. The

proposed Partition Matrix uses this characteristic of Hand Gesture Recognition, repeatedly applies the proposed Adaptive SURF Tracking method on the testing samples under different frame rates. Hence for videos in the hard set, the Partition Matrix is able to capture target gesture trajectory pattern, out of the dramatically changing background noises.

In order to demonstrate the proposed framework can perform well in arbitrary uncontrolled environments, we collected an even more challenging database called Warwick Hand Gesture Database. There are in total ten gesture classes as shown in Fig 5.3 are defined for our database. This database consists of two testing sets, the "easy" and "hard" sets. There are 360 video samples for training, 6 samples were captured from each of the 6 performers for each of the 10 gestures. There are 480 video samples in total for testing. For each testing set, 4 samples were collected from each of the 6 performers for every gesture. The "easy" set is collected in a controlled environment. The scene settings in the controlled environment include simulated natural sun light, single-coloured background, gesture performers wear long sleeve tops and no distractions in the background. The specifications of videos are the same as the Palm Graffiti Digits Database, with the same frame rate and resolution. Similar to the Palm Graffiti Digits database, the hard set of our database is captured with performers wearing short sleeve tops with cluttered backgrounds. The differences are: 1) No gloves are used in our training set; 2) Instead of 1-3 people, we have 2-4 people moving in the background; 3) The lighting conditions are random. The hard set is collected in two separate sessions.

The extent of distractions in the background of the Warwick Hand Gesture Database is much more severe than the Palm Graffiti Digits database. Fig 5.12 and 5.13 illustrate the tracking results of Adaptive SURF Tracking on samples of gesture six of hard sets from the two databases. It is obvious that the intra-class variance of our database is larger than the Palm Graffiti Digits database. In this experiment, the method of Bao et al. [2] and the original Dynamic Time Warping

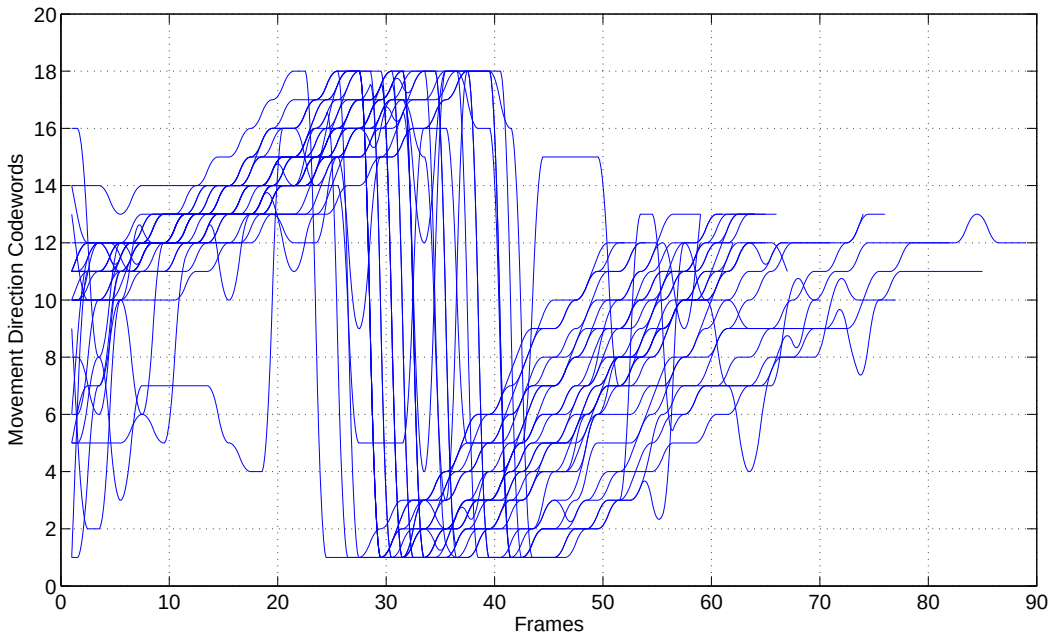

Figure 5.12: Gesture trajectories of all training samples of gesture "6" in the Palm Graffiti Digits database.

[174] are implemented and compared with our method, due to the reason that it is the foundation of [8]. The experimental results are shown in Table 5.4, Fig 5.14 and 5.15.

Since various window sizes (Eq 5.7) in the f_1 feature functions can affect the performance of the proposed framework greatly, experiments on adopting different window sizes are conducted on the hard set of the Warwick Hand Gesture Database. Table 5.5 - 5.9 illustrate the performances of the framework with window sizes from 1 - 4. Based on the results on the validation set, The number of hidden states is set to 13 and w is set to 1.

The total number of the hidden states has significant influence on the framework's performance as well. Experiments on various total number of hidden states are conducted on hard set of the Warwick Hand Gesture Database to locate the optimised hidden state number. Since the latent hidden states are serving as the label

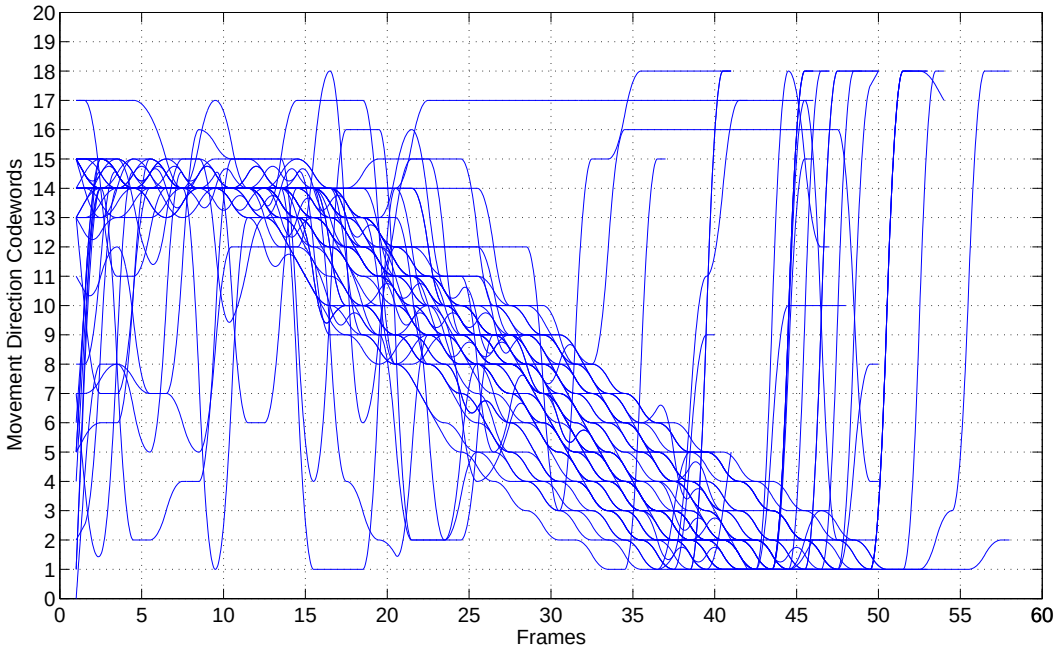

Figure 5.13: Gesture trajectories of all training samples of gesture "6" in the Warwick Hand Gesture Database.

Table 5.4: Comparison of performances with method of [2] on the Warwick Hand Gesture Database.

Warwick Hand Gesture Database		
	Easy Set	Hard Set
Bao et al. ICEICE 2011 [2]	71.00%	18.20%
Dynamic Time Warping [174]	75.00%	40.42%
The proposed method	93.00%	91.25%

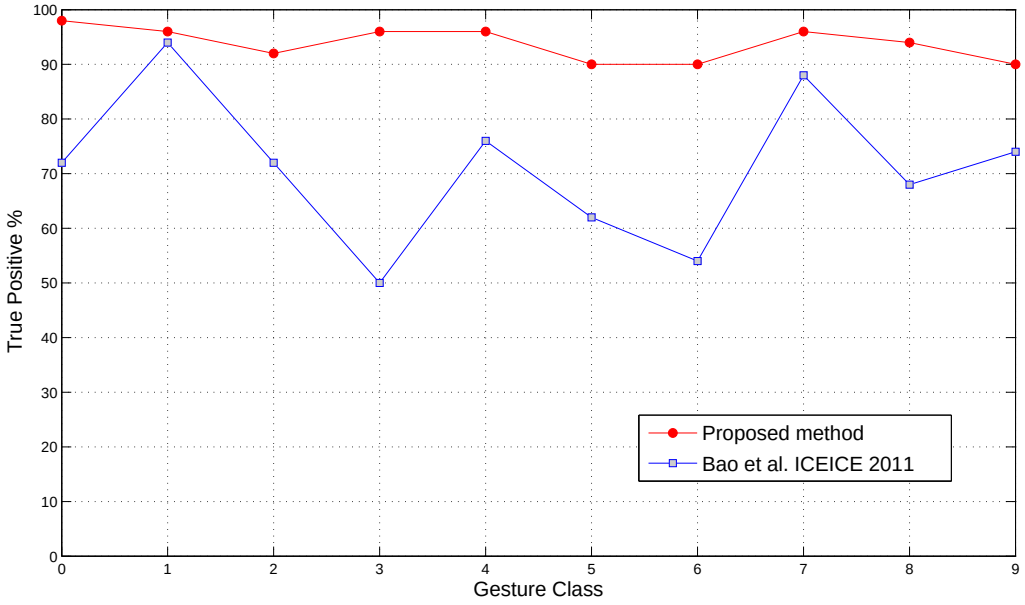

Figure 5.14: Comparison of performances with method of [2] on the esay set of Warwick Hand Gesture Database.

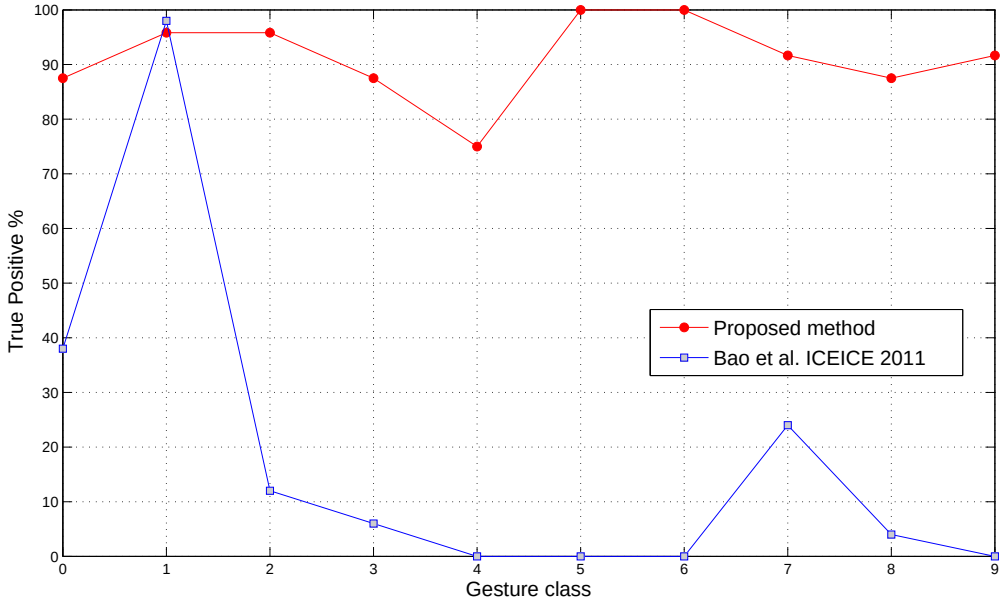

Figure 5.15: Comparison of performances with method of [2] on the hard set of Warwick Hand Gesture Database.

Table 5.5: Performance with $w = 1$ on the hard set of Warwick Hand Gesture Database.

Gesture Classes	Hard Set				
	Training Samples	Testing Samples	Recognition Results		
			True	Detected	Accuracy (%)
0	36	24	22	25	91.67
1	36	24	22	22	91.67
2	36	24	20	28	83.33
3	36	24	17	20	70.83
4	36	24	17	23	70.83
5	36	24	23	29	95.83
6	36	24	20	24	83.33
7	36	24	24	25	100.00
8	36	24	21	21	87.50
9	36	24	21	23	87.50
Overall	360	240	207	240	86.25

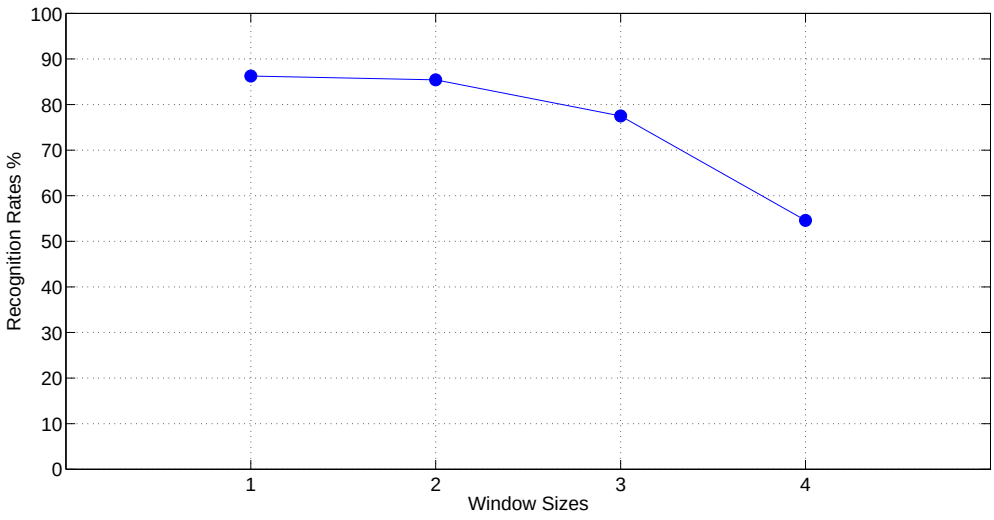

Figure 5.16: Performance with different window sizes.

Table 5.6: Performance with $w = 2$ on the hard set of Warwick Hand Gesture Database.

Gesture Classes	Hard Set				
	Training Samples	Testing Samples	Recognition Results		
			True	Detected	Accuracy (%)
0	36	24	21	23	87.50
1	36	24	23	25	95.83
2	36	24	18	26	75.00
3	36	24	18	19	75.00
4	36	24	15	19	62.50
5	36	24	23	24	95.83
6	36	24	20	23	83.33
7	36	24	22	28	91.67
8	36	24	24	24	100.00
9	36	24	21	29	87.50
Overall	360	240	205	240	85.42

Table 5.7: Performance with $w = 3$ on the hard set of Warwick Hand Gesture Database.

Gesture Classes	Hard Set				
	Training Samples	Testing Samples	Recognition Results		
			True	Detected	Accuracy (%)
0	36	24	18	21	75.00
1	36	24	18	21	75.00
2	36	24	21	29	87.50
3	36	24	13	16	54.17
4	36	24	21	31	87.50
5	36	24	23	26	95.83
6	36	24	20	29	83.33
7	36	24	21	27	87.50
8	36	24	13	17	54.17
9	36	24	18	23	75.00
Overall	360	240	186	240	77.50

Table 5.8: Performance with $w = 4$ on the hard set of Warwick Hand Gesture Database.

Gesture Classes	Hard Set				
	Training Samples	Testing Samples	Recognition Results		
			True	Detected	Accuracy (%)
0	36	24	17	33	70.83
1	36	24	18	21	75.00
2	36	24	4	17	16.67
3	36	24	11	34	45.83
4	36	24	3	11	12.50
5	36	24	22	23	91.67
6	36	24	9	21	37.50
7	36	24	19	34	79.17
8	36	24	16	22	66.67
9	36	24	12	24	50.00
Overall	360	240	131	240	54.58

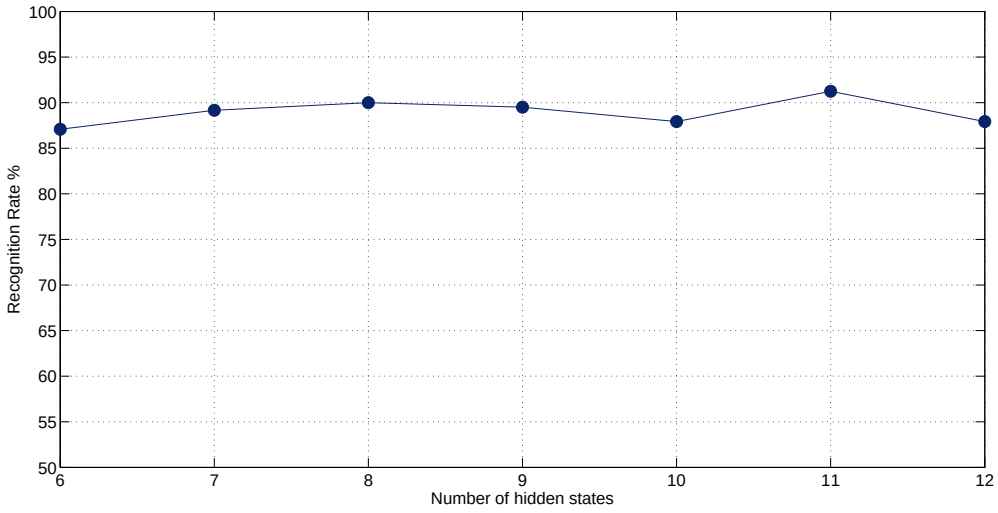

Figure 5.17: Performance with different number of hidden states.

for different parts or components of the pre-defined patterns, the suitable number of hidden states may not be complying with common senses. In Quattoni et al. (CVPR 2006) [149], it is proven that the optimised definition of hidden states is not always fitting with the natural structure of the pre-defined patterns. For example, to recognise different types of cars, the best hidden state definition is not according to segmentation of parts in the cars. In other words, a wheel or a door may not be a suitable individual hidden state, and the area contains a wheel and a door maybe is. Based on the performances on the validation set, which is one tenth of the training set, the best amount for hidden states is 11. In this series of experiments, the window size is set to 1. To better demonstrate the sensitivity of our method on the amount for hidden states and the window size on large number of testing samples, the testing results on the testing set are shown in Table 5.9 - 5.15 and Fig 5.17, instead of results on the validation set.

As for the computational costs, the proposed framework can run in average at: 55.00 ms/frame for easy sets, 56.75ms/frame for hard sets, on both experiments. That is 18.18 frames/sec and 17.64 frames/sec, respectively. Experiments were

Table 5.9: Performance with $|h| = 6$ on the hard set of Warwick Hand Gesture Database.

Gesture Classes	Hard Set				
	Training Samples	Testing Samples	Recognition Results		
			True	Detected	Accuracy (%)
0	36	24	22	26	91.67
1	36	24	23	23	95.83
2	36	24	22	27	91.67
3	36	24	16	18	66.67
4	36	24	19	20	79.17
5	36	24	22	29	91.67
6	36	24	21	25	87.50
7	36	24	22	23	91.67
8	36	24	23	24	95.83
9	36	24	19	25	79.17
Overall	360	240	209	240	87.08

Table 5.10: Performance with $|h| = 7$ on the hard set of Warwick Hand Gesture Database.

Gesture Classes	Hard Set				
	Training Samples	Testing Samples	Recognition Results		
			True	Detected	Accuracy (%)
0	36	24	22	25	91.67
1	36	24	22	22	91.67
2	36	24	19	26	79.17
3	36	24	20	22	83.33
4	36	24	23	23	95.83
5	36	24	22	22	91.67
6	36	24	19	22	79.17
7	36	24	24	26	100.00
8	36	24	22	27	91.67
9	36	24	21	25	87.50
Overall	360	240	214	240	89.17

Table 5.11: Performance with $|h| = 8$ on the hard set of Warwick Hand Gesture Database.

Gesture Classes	Hard Set				
	Training Samples	Testing Samples	Recognition Results		
			True	Detected	Accuracy (%)
0	36	24	24	26	100
1	36	24	23	23	95.83
2	36	24	24	26	100.00
3	36	24	17	18	70.83
4	36	24	23	29	95.83
5	36	24	23	31	95.83
6	36	24	22	22	91.67
7	36	24	21	24	87.50
8	36	24	20	20	83.33
9	36	24	19	21	79.17
Overall	360	240	216	240	90.00

Table 5.12: Performance with $|h| = 9$ on the hard set of Warwick Hand Gesture Database.

Gesture Classes	Hard Set				
	Training Samples	Testing Samples	Recognition Results		
			True	Detected	Accuracy (%)
0	36	24	20	22	83.33
1	36	24	24	26	100.00
2	36	24	23	30	95.83
3	36	24	20	22	83.33
4	36	24	22	27	91.67
5	36	24	22	25	91.67
6	36	24	23	27	95.83
7	36	24	23	23	95.83
8	36	24	20	20	83.33
9	36	24	18	18	75.00
Overall	360	240	215	240	89.5

Table 5.13: Performance with $|h| = 10$ on the hard set of Warwick Hand Gesture Database.

Gesture Classes	Hard Set				
	Training Samples	Testing Samples	Recognition Results		
			True	Detected	Accuracy (%)
0	36	24	20	22	83.33
1	36	24	22	22	91.67
2	36	24	21	27	87.50
3	36	24	17	20	70.83
4	36	24	20	22	83.33
5	36	24	21	25	87.50
6	36	24	23	28	95.83
7	36	24	23	26	95.83
8	36	24	22	25	91.67
9	36	24	22	23	91.67
Overall	360	240	211	240	87.92

Table 5.14: Performance with $|h| = 11$ on the hard set of Warwick Hand Gesture Database.

Gesture Classes	Hard Set				
	Training Samples	Testing Samples	Recognition Results		
			True	Detected	Accuracy (%)
0	36	24	21	21	87.50
1	36	24	23	25	95.83
2	36	24	23	28	95.83
3	36	24	21	21	87.50
4	36	24	18	19	75.00
5	36	24	24	28	100.00
6	36	24	24	30	100.00
7	36	24	22	23	91.670
8	36	24	21	22	87.50
9	36	24	22	23	91.67
Overall	360	240	219	240	91.25

Table 5.15: Performance with $|h| = 12$ on the hard set of Warwick Hand Gesture Database.

Gesture Classes	Hard Set				
	Training Samples	Testing Samples	Recognition Results		
			True	Detected	Accuracy (%)
0	36	24	20	24	83.33
1	36	24	23	24	95.83
2	36	24	23	24	95.83
3	36	24	18	18	75.00
4	36	24	19	23	79.17
5	36	24	21	25	87.50
6	36	24	20	25	83.33
7	36	24	24	27	100.00
8	36	24	24	27	100.00
9	36	24	19	23	79.17
Overall	360	240	211	240	87.92

performed on a common 3.3GHz 4-core 8GB RAM Windows machine with C++ implementation with openMP.

The criteria of real time computing has various versions in different research fields. According to Real-Time Digital Signal Processing: Fundamentals, Implementations and Applications [175], by SM Kuo et al.: A real-time DSP system demands that the signal processing time, t_p, must be less than the sampling period, T, in order to complete the processing task before the new sample comes in. If t_o is the overhead of I/O operations, the criteria is:

$$t_p + t_o < T \tag{5.26}$$

This is referred as the hard criteria [175]. It is mostly used to evaluate methods for time sensitive tasks, including time recording, stock exchanging, etc. In the context of HGR, hard criteria means the processing frame rate must larger than the frame rate of the original video stream. However for interactive service applications such as HGR, the widely used real time criteria is the soft criteria. There is no unified definition of soft criteria in computer vision community. In the field of HGR, as long as the method does not cause severe time delay to affect user experience, the method could be called real time. The traditional minimum processing speed for real time computing is the half of the original video frame rate. Hence our framework is able to perform comfortably in real time.

5.3 Robustness

Following the discussion in Section 4.2, some of the challenges listed in Section 1.3 are tackled by the gesture classifier. The challenges are discussed individually in this section, with explanation of the corresponding countermeasures. The unsolved challanges are also discussed in this section.

5.3.1 Gesture Similarity

For the challenge of high gesture similarity, such as gesture "6" and "0", the proposed framework presented decent results in the experiments, due to the definition of the partition function. The partition function is the normalised summation of all weighed feature function values (Eq 5.18). Hence for testing samples of the gesture "6", the feature functions extracted from the first stroke, namely the vertical down stroke in the beginning of the gesture "6", would be assigned with large weights. That is because the distinctiveness of these feature functions for separating gesture "6" from "0". The feature functions that are shared by gesture "6" and "0", would be assigned with low weights. In other words, the proposed method can detect the minor difference between the similar gestures, then assign them with large voting powers.

Although in some experiments the recognition rate for gesture "6" is lower than other gestures, the over-all performance is still satisfactory. Without dedicated algorithms to learn the similarity among gestures, the ability of the proposed framework to recognise similar gestures in videos with continuous gestures is compromised to certain extent. This will be discussed in Section 7.3.

5.3.2 Gesture Complexity

The challenge of high gesture complexity is tackled by the factorisation of the potential function in the classifier. Since the classifier only monitors local trajectory patterns, if the gestures are long with large number of local trajectory patterns, the only negative influence is the long off-line training time. Although there are HGR applications with more complex gesture vocabularies, such as Sign Language Recognition, the proposed framework is designed for manipulative HGR, instead of communicative HGR (Section 2.1). The task of recognising 10 hand signed digits is already considered one of the complex gesture vocabularies for manipulative HGR. Thereby, as long as the HCRF models are thoroughly trained, the proposed method

can perform well with gesture vocabulary with high level of complexity.

5.3.3 Gesture Size Variance

For the gesture size variance challenge, the countermeasure in the proposed framework is the normalised partition function (Eq 5.38). Since the partition function is normalised to average partition value per frame, the sizes of the testing samples is irrelevant to gesture classification. Also, for all class labels, the final partition value is the weighed sum of all partition values in the Partition Matrix (Eq 5.26). Hence, if the testing sample has a relatively large scale gesture trajectory, which means the sample has a large number of frames, that would enlarge the final partition values for all class labels. But the ratios among the final partition values of all gesture classes are not changed by the varying sizes of the testing samples. Therefore, the proposed method can accurately classify the samples with different sizes.

5.3.4 Unsolved Challenges

There are still three challenges on the list in Section 1.3, the proposed framework is still unable to overcome.

The first one is gesture angle variance, namely gesture rotations. As discussed in Section 5.2.1, one of the main causes of the intra-class variance in HGR applications is distorted samples, as shown in Fig 5.8. Essentially the distortions can be seen as in-plane rotations. The proposed framework is capable of tolerating the rotations to certain extent. The level of tolerance highly depends on the variety of the training samples. For the rotations have appeared in the training set frequently, the probabilistic model can learn the pattern and treat the rotation as one of the distortion patterns of the gesture class. However, for unfamiliar rotations in the testing samples, the proposed framework could fail. Hence, the proposed framework is sensitive to unfamiliar rotations.

The second one is large vocabularies. There are only 10 gestures in the

vocabulary of this thesis. However, similar to the challenge of high gesture complexity, 10 gestures are more than enough for controlling HGR applications. Hence the experimental setting is still adequate to validate the performance of the proposed framework for manipulative HGR applications. For communicative HGR, such as Sign Language Recognition, the vocabularies are usually with thousands of words. For these vocabularies, the performance of the proposed framework remains unknown. Due to the absence of sub-gesture reasoning methods, the possibility for the framework to deliver unsatisfactory results is relatively high. For large vocabularies, the inter-class variance is likely to be low. In other words, there will be large number of similar gestures. Without the ability to learn the relationships between the gestures, the proposed framework could fail to perform robust HGR for large vocabularies.

The third one is the double handed gestures. Double handed gestures are common in Sign Language Recognition. But for manipulative HGR, vocabularies with single handed gestures are adequate for the task. Hence in the experiments of this thesis, double handed gestures are not considered.

5.4 Conclusions

In this chapter, a novel weighting scheme is proposed for performing HGR in uncontrolled environments. The main advantages of this method are: 1. It can monitor temporal features on different scales. 2. It is capable of dealing with multiple hand candidates at the same time. Similar with deep learning methods, the proposed Partition Matrix monitors temporal features on different scales. But the proposed method still has few drawbacks: 1. It can only monitor the temporal features on pre-defined scales, and currently it has only been tested for using 4 scales. 2. The temporal features are trained on a single scale. Unlike deep learning methods, the features are not trained on different scales. Hence, training a connected multi-layer

HCRF model similar with the RNN model could be potentially beneficial for improving the Partition Matrix.

Chapter 6

Hand Gesture Spotting in Uncontrolled Environments

In Chapters 5 the proposed HGR framework (Fig 1.3) is working under one assumption that the video samples are already segmented into isolated gesture videos. That means all training and testing samples are manually segmented to contain only one gesture. In other words, the method treats the first frame in the video as the start of the single gesture, and the last frame as the end of it. However, in real world HGR applications, the users usually need to send multiple commands in a row. For example, the user wants to dial a number on the phone with HGR control. The task of recognising continuous gestures without the prior knowledge about the starting and ending points of isolated gestures is called Hand Gesture Spotting (HGS) or Hand Gesture Segmentation. The main challenge of HGS is to determine the starting and ending point of each gesture in the video stream, with inter-connecting meaningless hand movements between the predefined gestures. In this thesis, a novel HGS method is proposed, as part of the proposed HGR framework, for segmenting and recognising hand gestures in the uncontrolled environments. Experimental results on the Warwick Hand Gesture Databases prove that this method is capable of tackling severe background distractions.

The proposed spotting scheme is a forward spotting scheme that uses Partition Matrix (introduced in Chapter 5) to evaluation whether the current frame is the starting or ending point of a meaningful gesture. The main contributions of the proposed spotting scheme over other existing spotting schemes are: 1. It can monitor temporal features on different scales. 2. it can perform gesture spotting in uncontrolled environments with multiple hand candidates.

6.1 Garbage Model

To perform HGS in uncontrolled environments, the first task is hand tracking. The proposed HGS method uses the Adaptive SURF Tracking method introduced in Chapter 4. Since the tracking scheme is able to monitor all eligible hand candidates in the scene to tackle the challenge of moving objects in the background, a novel spotting scheme based on HCRF is proposed specifically for working with our tracking scheme and accomplishing hand gesture spotting in uncontrolled environments.

With the inter-connecting meaningless hand movements, the task of HGS in uncontrolled environments can be seen as HGR for isolated gestures with the possibility that the target signing hand itself is producing meaningless hand movements in the video. Namely, at some parts of the video stream, none of the hand candidates are signing meaningful predefined hand gestures. This scenario is similar with speech recognition or speaker recognition in uncontrolled environments. The voice signal also contains words and background noises. There have been some attempts to tackle the influence of the background noise [176, 177, 178]. The general idea of these methods is to build a filter based on prior knowledge of the target human voice signal, or a probabilistic model dedicatedly for the background noise. The proposed HGS method takes this idea one step forward in the context of HGS. A dedicated Non-Sign Model (or garbage model) is comprised of feature functions that are derived from the feature functions of the gesture vocabulary.

HCRF models are undirected graph models that can learn the transition probabilities between different modules of the observation sequences, with hypothetical states. In our experiments, the task is recognising 10 hand-signed digits $Y = \{y_0, y_1, \ldots, y_{m-1}\}$, similar with the experiments for isolated gestures in Section 5.2.3. For a testing sample, a series of observation sequences $X = \{x_{u,r} | u = 0, 1, \ldots U-1, r = 0, 1, \ldots R-1$ are extracted by the Adaptive SURF Tracking method. Each movement direction vector $x_{u,r} = \{o_0, \ldots, o_{l-1}\}$ contains l observation states as l is the number of frames in the video fragment corresponding to $x_{u,r}$. The hidden states $H = \{H_0, H_1, \ldots, H_{n-1}\}$, defined for the HGS experiments on the Warwick Hand Gesture Database. Same as the definition in Section 5.2.1, for each observation sequence $x_{u,r}$, a vector of hidden states $\vec{h} = \{h_0, h_1, \ldots, h_{l-1}\}$ is assgined to it. Each element of the hidden state vector $\vec{h}$ is one of the hidden states in H and it is corresponding to an observation state in the observation sequence $x_{u,r}$. Similar with the "easy" and "hard" HGR testing sets in the Warwick Hand Gesture Database used in Section 5.2.3, there are also two testing sets for HGS in the Warwick Hand Gesture Database, namely the "easy" and "hard" HGS testing sets. The optimisation method for training the weight vector $\vec{\theta}$ of the potential function is Limited Memory BroydenFletcherGoldfarbShanno (Limited Memory BFGS) method, also the same as in Section 5.2.3. The weight vector is initialised with the mean value and the regularisation factors set to zero.

Given a class label y_g, a observation sequence $x_{u,r}$, a hidden state vector $\vec{h}$ and the weight vector $\vec{\theta}$, the standard HCRF model (Section 5.2.1) can be represented as,

$$P\left(y_g | x_{u,r}, \vec{\theta}\right) = \sum_{\vec{h}} P\left(y_g, \vec{h} | x_{u,r}, \vec{\theta}\right) = \frac{\sum_{\vec{h}} \exp\left\{\Psi\left(y_g, \vec{h}, x_{u,r} | \vec{\theta}\right)\right\}}{\sum_{y} \sum_{\vec{h}} \exp\left\{\Psi\left(y, \vec{h}, x_{u,r} | \vec{\theta}\right)\right\}} \tag{6.1}$$

where the numerator that can be understood as the score of the observation sequences $x_{u,r}$ given the class label y_g is defined as the partition function:

$$Z\left(y_g|x_{u,r},\vec{\theta}\right)=\sum_{\vec{h}}\exp\left\{\Psi\left(y_g,\vec{h},x_{u,r}|\vec{\theta}\right)\right\} \tag{6.2}$$

The potential function is comprised of three types of feature functions, as introduced in Section 5.2.1.

$$\begin{aligned}\Psi\left(y_g,\vec{h},x_{u,r}|\vec{\theta}\right)=&\sum_{j=0}^{l-1}\sum_{i=0}^{D\times n}\theta_{1,i}\cdot f_{1,i}\left(x_{u,r},h_j\right)+\sum_{j=0}^{l-1}\sum_{i=0}^{m\times n}\theta_{2,i}\cdot f_{2,i}\left(y_g,h_j\right)+\\&\sum_{(j,k)\in E}\sum_{i=0}^{m\times n^2}\theta_{3,i}\cdot f_{3,i}\left(y_g,h_j,h_k\right)\end{aligned} \tag{6.3}$$

where j and k are hidden state index, i is the feature function index and E is the set of adjacent hidden states in the hidden state vector $\vec{h}$, as mentioned in Section 5.2.1. To segment meaningful gestures from the transitional hand movements, we define a dedicated class label y_G for the garbage gestures. Hence, the gesture class set becomes $Y=\{y_0,y_1,\ldots,y_9,y_G\}$. To build correspondent feature functions for this non-sign gesture class, two series of new feature functions are defined for y_G based on the trained weights of the existing feature functions.

The reasons for not including garbage gesture samples in the training stage for learning non-sign gesture patterns in the HCRF model are twofold. Firstly, the strategy of treating garbage hand movements as familiar gesture patterns in the training stage is unrealistic. It is nearly impossible to collect all meaningless hand movement patterns for training, due to the infinite possibilities of random movements. Secondly, if samples of garbage hand movements are included in the training process, and all frames of the samples are labelled for supervised learning, the tracking results would be biased towards the background distractions and interconnecting hand movements in the training set, which makes the weights of states and transition feature functions also biased to the familiar noise patterns in the

training set. Hence the most accurate way of learning the weights of non-sign feature functions is estimating them from the known weights of meaningful gesture feature functions. In other words, learning meaningless hand movements as the features that are not consistent with the meaningful gestures.

Two new series of feature functions for non-sign gestures are used in the proposed HGS method. One is a set of state feature $f'_{2,i} = \{f_{2,i} \,|i \in (\, L_2, y_G \cdot H_j] \,, H_j \in H \,\}$ where the class lable of the garbage model y_G is used as its numerical value. This new set of feature function represents the compatibility between the hidden states and y_G. The weights of these new state features ${\theta_{2,i}}' = \{\theta_{2,i} \,|i \in (\, L_2, y_G \cdot H_j] \,, H_j \in H \,\}$ are calculated from the trained weights of the existing state features. For a given hidden state H_j:

$$\theta_{2,\, y_G \cdot H_j} = \mu_2 \left(H_j\right) + T \cdot \sqrt{v_2 \left(H_j\right)} \tag{6.4}$$

where T is a scale factor that set to 1.2 empirically. $\mu_2 \left(H_j\right)$ is the average weight of the state features of H_j and all meaningful gesture classes,

$$\mu_2 \left(H_j\right) = \frac{\sum_{g=0}^{m-1} \theta_{2,\, y_g \cdot H_j}}{m} \tag{6.5}$$

and $v_2 \left(H_j\right)$ is the variance of the weights θ_2 for state features of H_j and all meaningful gesture classes,

$$v_2 \left(H_j\right) = \frac{\sum_{g=0}^{m-1} \left(\theta_{2,\, y_g \cdot H_j}\right)^2}{m} - \left[\mu_2 \left(H_j\right)\right]^2 \tag{6.6}$$

The other series of new feature functions is transition feature functions $f'_{3,i} = \{f_{3,i} \,|i \in (\, L_3, H_j \cdot H_k \cdot y_G] \,; H_j, H_k \in H; (j,k) \in E \,\}$, as defined before E is the set of adjacent hidden states. ${\theta'}_{3,i} = \{\theta_{3,i} \,|i \in (\, L_3, H_j \cdot H_k \cdot y_G] \,; H_j, H_k \in H; (j,k) \in E \,\}$ are the corresponding weights of the new transition feature functions and they are

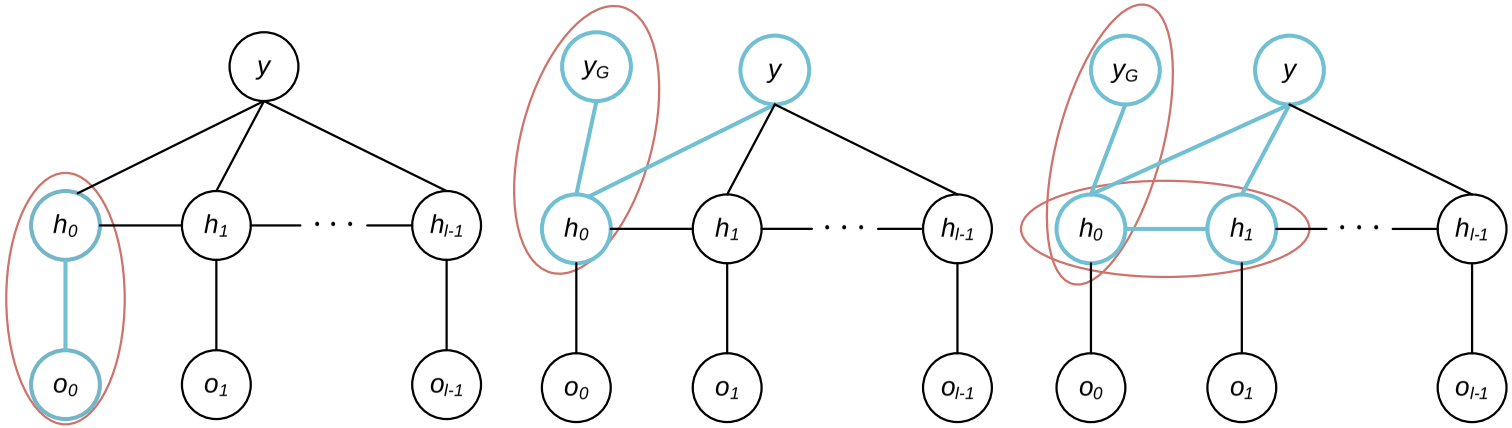

Figure 6.1: Three types of feature functions for the garbage model. Left: the f_1 features, remain the same as in isolated gesture recognition; Middle: the new f_2 feature functions represent the compatibility of hidden states and garbage hand movements; Right: the new f_3 feature functions indicating the compatibility of the transitional hidden states and garbage hand movements.

calculated based on existing transition features. For a given pair of hidden state H_j and H_k:

$$\theta_{3,H_j \cdot H_k \cdot y_G} = \mu_3 (H_j, H_k) + T \cdot \sqrt{v_3 (H_j, H_k)} \tag{6.7}$$

T is still set to 1.2. $\mu_3 (H_j, H_k)$ and $v_3 (H_j, H_k)$ are the mean and variance of weights of existing transition feature functions respectively,

$$\mu_3 (H_j, H_k) = \frac{\sum_{i=0}^{m-1} \theta_{3,H_j \cdot H_k \cdot y_i}}{m} \tag{6.8}$$

$$v_3 (H_j, H_k) = \frac{\sum_{i=0}^{m-1} \left(\theta_{3,H_j \cdot H_k \cdot y_i}\right)^2}{m} - \left[\mu_3 (H_j, H_k)\right]^2 \tag{6.9}$$

These weights are measures for the significance of certain state in the non-sign movements. In other words, given the different probabilities of all meaningful gesture classes that contain a certain hidden state H_j, $\theta_{2,y_G \cdot H_j}$ represents the probability of this hidden state appearing in garbage hand movements. If the hidden state H_j is only appearing in a meaningful gesture classes y_i, the average weight $\mu_2 (H_j)$ would be relatively low, but the variance $v_2 (H_j)$ would be high. Due to the uniqueness of this hidden state to y_i, the state features $\theta_{2,y_i \cdot H_j}$ must be assigned a large weight during the training process, and the state feature for H_j with other gesture classes would have relatively smaller weights to enhance the voting power of $\theta_{2,y_i \cdot H_j}$. Hence the variance $v_2 (H_j)$ would be high. Therefore the final value of $\theta_{2,y_G \cdot H_j}$ depends on the value of $v_2 (H_j)$ which represents the extent of effectiveness of $\theta_{2,y_i \cdot H_j}$ as a feature function. The more effective the feature function is, namely the rarer the feature function is, the larger the probability of this feature function appearing in a garbage gesture is. Intuitively, this strategy of calculating the garbage model is valid. If a feature function is rarely seen in pre-defined gesture classes in the training

set, it should be treated as a non-meaningful feature function.

6.2 Multiple Sliding Windows Forward Spotting Scheme

A forward spotting scheme is proposed in this thesis to determine the starting and ending frames of the meaningful gestures in continuously sighed gesture videos in uncontrolled environments. Fig 6.2 demonstrates the structure of the proposed spotting scheme. As shown in Fig 6.2, to perform gesture spotting on an input video, starting from the first frame, the proposed spotting scheme takes a series of sliding windows with different sizes to extract video fragments from the input video. The video fragments start from the current frame f_c with the sliding window size L_g, are defined as:

$$S_{c,g} = \{f_i | i = c, c+1, c+2, \ldots, c+L_g\} \tag{6.10}$$

where L_g is the average length of all the training samples of gesture g. Hence there are in total 10 different sliding window sizes $S_{c,g}$. Unlike the standard CRF model, HCRF model do not produce gesture labels for every frame. For evaluating the probabilities of the current frame being part of each gesture classes, the sliding window is designed to has various sizes, instead of fixed size [3].

For the proposed framework, as explained in Chapter 5, the length of gesture trajectories are irrelevant for classification, due to two reasons. One is utilising the proposed normalised Partition Function (Eq 5.18), the other is factorisation of the potential function (Eq 5.5). Sizes of the sliding windows are also irrelevant for gesture spotting. Despite the size invariance property of the proposed framework, the purpose of using average sizes of training samples of each gesture classes as sizes of the sliding windows is to utilise information in the training set. When the gesture trajectory has similar size with one of the sliding window sizes, the difference between the partition value of the correct class label and other labels

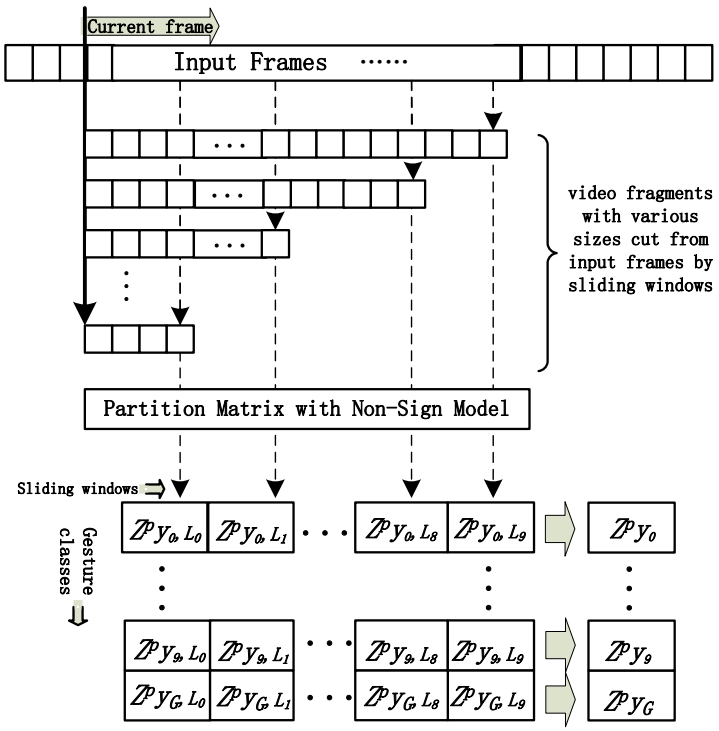

Figure 6.2: Structure of the forward spotting scheme with Partition Matrix for videos with multiple hand candidates. A series of video fragments are cut from the input frames by sliding windows with different sizes. Then the series of video fragments are put through Partition Matrix with Non-Sign Model introduced in the last section. The resutls of the Partition Matrix are used to form a matrix that produces the final spotting results for the current frame.

would be relatively large. In other words, those video fragments that are similar to the training samples can induce the best performance of the trained HCRF model. Although different choices of the sliding window sizes have small influence on the recognition rate, it is still to the advantage of the proposed framework to use the average sizes of training samples of every gesture class.

As shown in Fig 6.2, after video fragment set $S_{c,0-9}$ is extracted, the Partition Matrix with Non-Sign Model is used to evaluate every video fragment. Then a matrix is formed as shown in the bottom of Fig 6.2. Every column of this matrix is the normalised partition values of one video fragment against all gesture classes:

$$\begin{pmatrix} ({Z^p}_{y_0,L_0}) & \cdots & ({Z^p}_{y_0,L_9}) \\ \vdots & \ddots & \vdots \\ ({Z^p}_{y_G,L_0}) & \cdots & ({Z^p}_{y_G,L_9}) \end{pmatrix} \tag{6.11}$$

where ${Z^p}_{y,L_g}$ indicates value of the partition function of the video fragment which was defined in Eq 5.21, with sliding window size L_g and the class label y. The final partition values of all gesture classes F_y of the current frame f_c are calculated as:

$$F_y = \sum_g {Z^p}_{y,L_g} \tag{6.12}$$

which are the sums of all elements in each row of the matrix. For a specific gesture trajectory, partition values for different sliding windows represent the local similarities of different parts of the trajectory with the predefined gestures. Namely, by adding up the partition values for different parts of the gesture trajectory, the final score F_y can capture the over all similarity between the gesture trajectory and the meaningful gestures classes.

As the tracking scheme combined with the Partition Matrix, the proposed spotting scheme is able to capture the target meaningful gesture trajectories from

multiple hand candidates in uncontrolled environments, while the vast majority of the HGR community only consider HGS in controlled environments with unified background, no distractions appearing in the scene [3].

With the sliding window mechanism, the only remaining step to complete the HGS task is segmenting meaningful gestures with partition values produced by the sliding windows. A Differential Probability (DP) function is proposed to determine the starting and ending frames of the meaningful gestures. For all gesture classes, including the garbage gesture class, the summations of all normalised partition values from different sliding window sizes, namely F_0 to F_9 are used to calculate the Differential Probability function. Given the starting position of the sliding windows f_c, Differential Probability function is defined as:

$$D^P(f_c) = \max_{y \in (Y - y_G)} (F_y - F_{y_G}) \tag{6.13}$$

If the value of the Differential Probability is positive at a certain frame, namely the condition $\exists y_i : F_{y_i} > F_{y_G}, y_i \in (Y - y_G)$ is satisfied, the current frame is treated as a starting point of gesture y_i. That means for frame f_c, one of the video fragments start at f_c is silimar with one of the meaningful gesture classes more than similar with the garbage model. At a later frame f_{c+k}, when the DP value returns to negative, namely the condition $\forall y_i : F_{y_i} < F_{y_G}, y_i \in (Y - y_G)$ is satisfied, f_{c+k} is treated as the ending point of the gesture y_i. The starting point of the sliding windows is re-initialised to f_{c+k} as well. As shown in Fig 7.3, the proposed method is not for real-time HGS. The video with continuous gestures must be completed, and fed into the proposed HGS method as the input video. Essentially, the proposed HGS method is a forward spotting method, which means the sliding window is moving forward to detect the starting point of meaningful gesture first, before the ending point is detected. The opposite strategy is backward spotting, which detects the ending point of the gesture first, then move the sliding window backwards to

locate the starting point. Hence, for real-time HGS, backward spotting method is the common choice. The forward spotting methods apparently require a "buffer" to store the video fragments from the starting point to the current frame.

The proposed HGS scheme is not designed for real-time HGS, and the reasons are twofold. Firstly, the proposed HGR framework produces large amount of feature functions for the potential function. For real-time HGS, all the feature function values of all video fragments from different sliding windows need to be stored in the buffer, then evaluated to determine the starting point of the meaningful gestures. Secondly, between the meaningful gestures, if a long period of meaningless hand movements exists, the video fragments could have large number of frames. That could make the computation for the DP function intractable. Also, if the number of hand candidates in the scene is large, the calculation of DP function would be even more computational intensive.

6.3 Experiments

For testing the proposed method, a database for 10 hand-signed digits is collected to provide the proposed spotting scheme a uncontrolled environment with severely distracted unconstrained background. The training set contains 6 gesture performers. For each gesture class, each gesture performer signs the gesture 6 times. Hence, there are in total 360 training samples (the training samples are manually segmented). In the proposed HGS method, only the HCRF models for predefined gesture classes need to be trained, while the garbage model and Partition Matrix are calculated during the inference process based on the trained HCRF models. Therefore, the training process only requires isolated gesture samples. All the training samples are collected with perfectly controlled environments, with controlled lighting and unified single colour background, without any kind of distraction in the background. Two testing sets are collected, one 'easy' set with the same scene setting as the

training set, and one 'hard' set with uncontrolled background. In the hard set, the background is normal office scene under natural sun light without any artificial lightings, and the performers wear short sleeve tops. There are 2-4 people constantly and randomly walking around in the background, and deliberately making meaningless hand movements beside the gesture performer. For both testing sets, each of the 6 performers signs gesture 0-9 continuously in one video sample for 4 times. Hence there are 240 gesture samples in both easy and hard sets. The method of [3] is also a forward spotting method which based on the original CRF model. Hence it is implemented and tested in this experiment to show the unique advantages of our method over the state-of-the-art forward hand gesture spotting method. Comparing with [3] our method has additional advantages: 1) Our method is for uncontrolled environments with multiple hand candidates in the scene, while [3] was not tested on uncontrolled scene settings. 2) In each frame, instead of one fixed-size sliding window, our method segment the video with multiple sliding windows. Then these video segments are fed into the Partition Matrix for classification, instead of the original CRF model. That gives our method the ability to deal with background distractions (multiple hand candidates). 3) Our method proposed a new DP function for detecting starting and ending frames of the meaningful gestures. The tracking results that are fed into our implementation of [3] are from our own tracking scheme, and the tracking results of the target hand ROI is manually picked out. That makes the experiments fair for comparison. In this experiment, the total number of hidden states in our method is 13.

Fig 6.4 shows how meaningful gesture can be detected from distracted background with other hands moving. Fig 6.3 illustrates the trajectories of all samples of gesture 6 in the training set, while Fig 6.4 shows the tracking results of all hand candidates in a testing sample of hard set. The red line indicating the target ROI, and we can see from frame 4 to 63, a gesture 6 trajectory is detected.

The experimental results of the proposed spotting scheme on the hard and

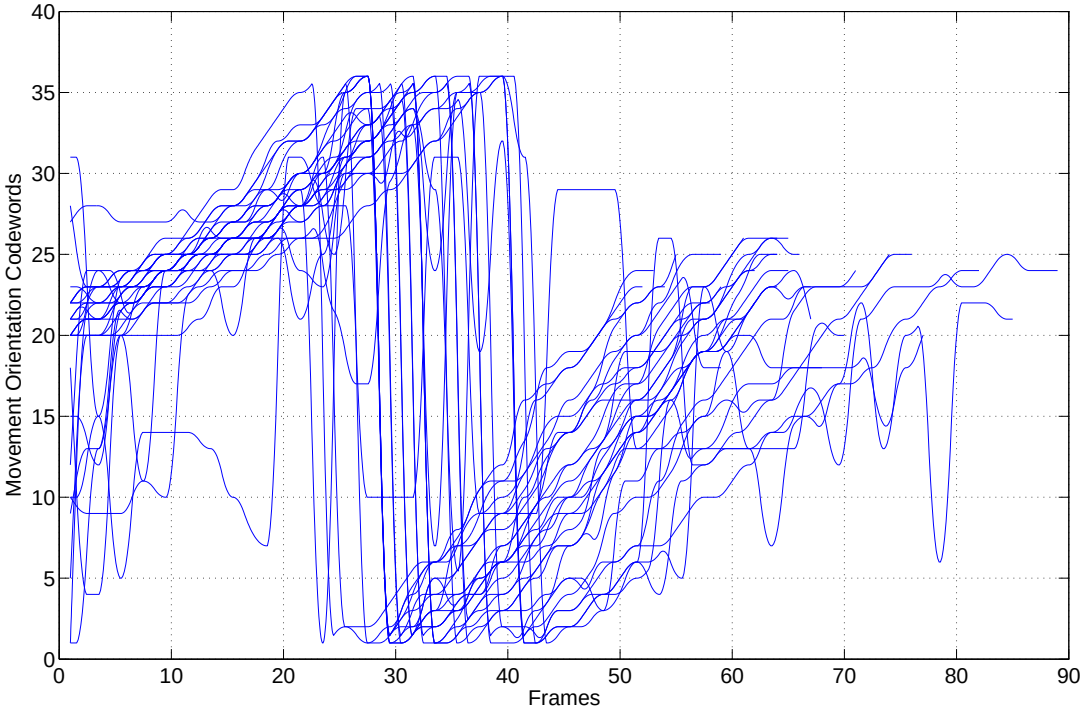

Figure 6.3: Trajectories of all training samples of gesture "6" in the Warwick Hand Gesture Database.

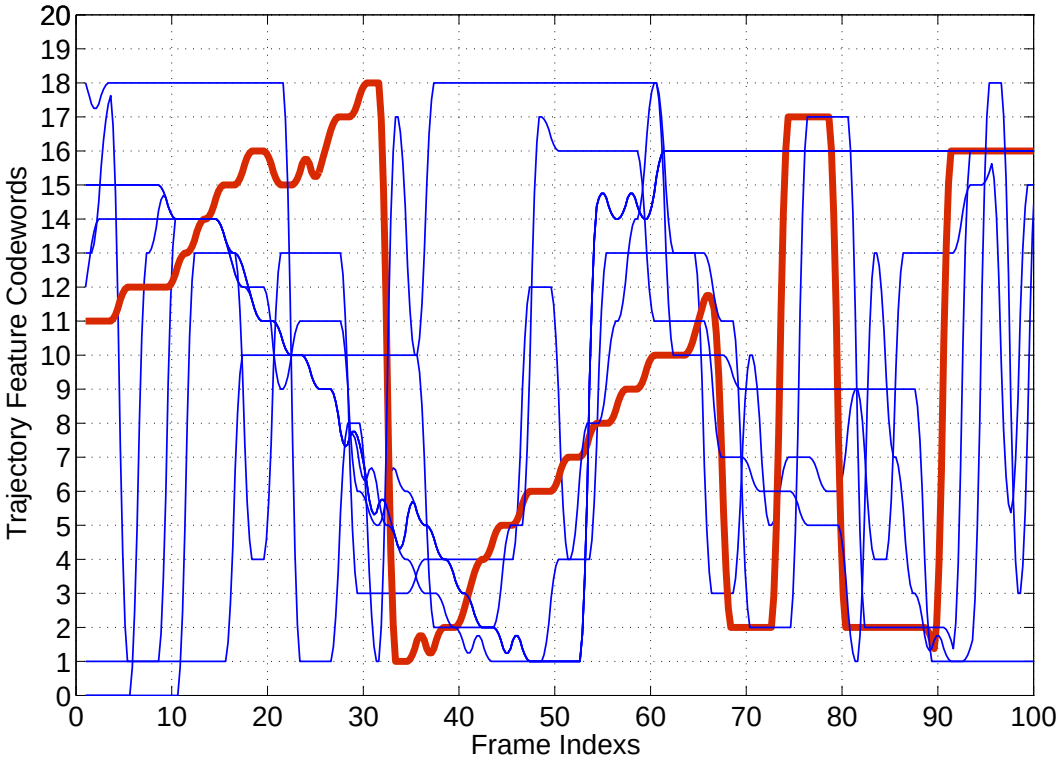

Figure 6.4: Trajectories of all hand candidates, including background distractions, from a fragment of a testing sample in Warwick Hand Gesture Database. The target hand is signing gesture "6" in this fragment.

easy testing sets are shown in Table 6.1 and Table 6.2. For the training process, there are in total 13 hidden states, the optimisation method is Limited Memory BroydenFletcherGoldfarbShanno (Limited Memory BFGS) algorithm, the same as the isolated gesture recognition experiments in Chapter 5. The weight vector is initialised with the mean value and the regularisation parameters are set to zero. A sample of uncontrolled scene setting is shown in Fig 7.6. There are in total 12 movement directions as the trajectory feature codewords.

The comparisons with the method in [3] on both easy and hard testing sets are shown in Table 6.3, Fig 6.6 and 6.7. The two methods share similar performances on different labels on the hard set (Fig 6.6 and 6.7), which means both method failed to beat the similarity among gestures. For the overall performance, the proposed method out-performed [3] with relatively small margins. The reason for decreased performances on gesture "6" and "0" for both the proposed method and the method of [3] is that no sub-gesture reasoning mechanism is included in both methods. Gesture "6" has high level of similarity with gesture "0". The sub-gesture reasoning methods are algorithms that learn the similarity and containing relationships among gesture classes. Hence, for a circle-like trajectory fragment that could be gesture "6" or "0", the sub-gesture reasoning methods can determine whether the it is part of a gesture "6", or a finished gesture "0". However, the over-all performance of the proposed HGS method is still satisfactory. For the starting vertical stroke of the gesture "6", gesture performs tend to draw the vertical stroke much shorter than the vertical stroke in gesture "9". Hence, samples of gesture "9" have enough amount of temporal features from the vertical strock to distinguish the samples from gesture "0" and "6". That is a possible reason for the worse performance on gesture "6" than gesture "9".

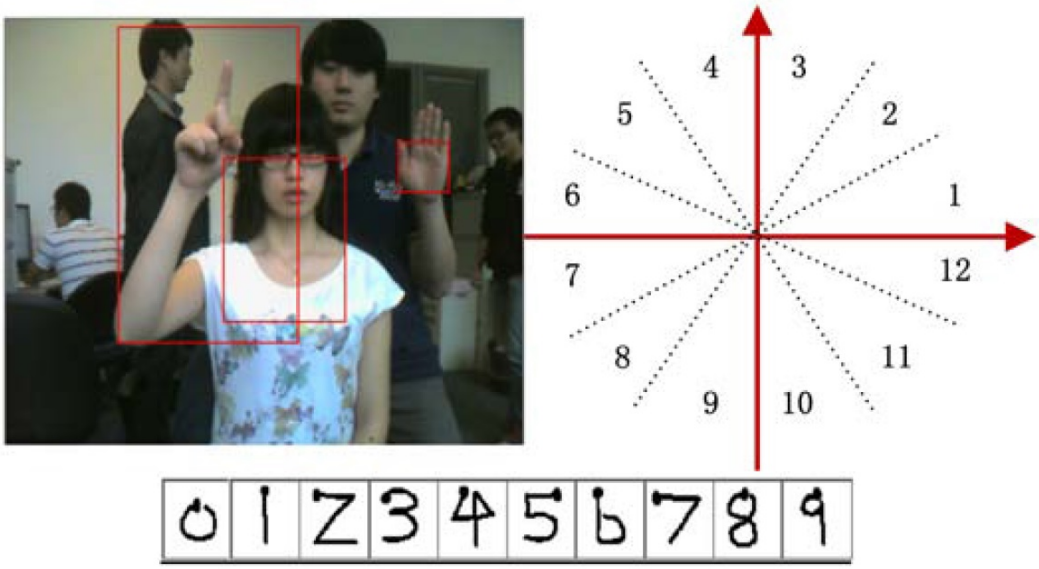

Figure 6.5: Upper left: Sample of testing video in the "hard" testing set with uncontrolled environments; Upper right: The movement directions codewords, and there are in total 12 directions. Bottom: the definition of the gesture set in the experiments.

Table 6.1: Results of the proposed method on the "hard" gesture spotting set of Warwick Hand Gesture Database.

Gesture Classes	Hard Set				
	Training Samples	Testing Samples	Recognition Results		
			True	Detected	Accuracy (%)
0	36	24	22	25	91.67
1	36	24	22	22	91.67
2	36	24	20	28	83.33
3	36	24	17	20	70.83
4	36	24	17	23	70.83
5	36	24	23	29	95.83
6	36	24	20	24	83.33
7	36	24	24	25	100.00
8	36	24	21	21	87.50
9	36	24	21	23	87.50
Overall	360	240	207	240	86.25

Table 6.2: Results of the proposed method on the "easy" gesture spotting set of Warwick Hand Gesture Database.

Gesture Classes	Easy Set				
	Training Samples	Testing Samples	Recognition Results		
			True	Detected	Accuracy (%)
0	36	24	23	23	95.83
1	36	24	22	22	91.67
2	36	24	23	24	95.83
3	36	24	22	22	91.67
4	36	24	21	26	87.50
5	36	24	22	24	91.67
6	36	24	21	25	87.50
7	36	24	24	26	100.00
8	36	24	24	25	100.00
9	36	24	23	23	95.83
Overall	360	240	225	240	93.75

Table 6.3: Comparison of performances with method in [3].

Warwick Hand Gesture Database		
	Easy Set	Hard Set
Eleezain et al. ICPR 2010 [3]	92.08%	82.08%
The proposed method	93.75%	86.25%

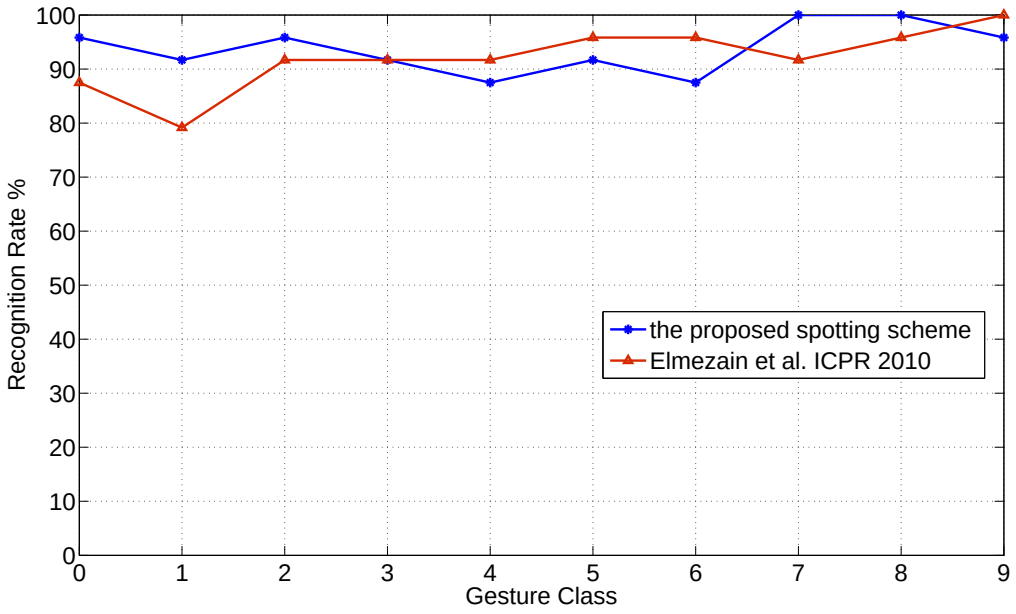

Figure 6.6: Comparison of performances on the ten gesture classes in the "easy" gesture spotting set of Warwick Hand Gesture Database.

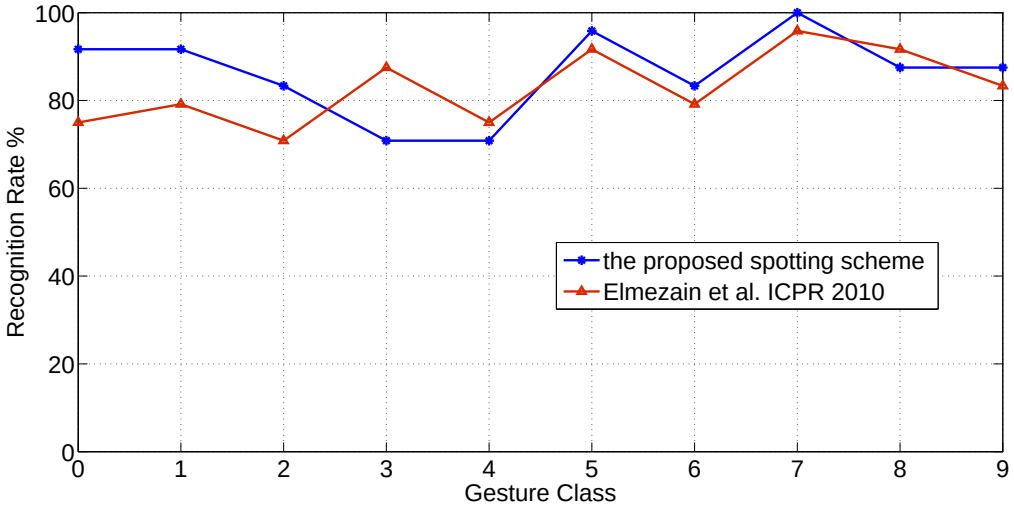

Figure 6.7: Comparison of performances on the ten gesture classes in the "hard" gesture spotting set of Warwick Hand Gesture Database.

6.4 Conclusions

A novel gesture spotting scheme is proposed in this chapter. The spotting scheme uses a sliding window mechanism conbined with the Partition Matrix introduced in Chapter 5. The main advantages of this spotting scheme are: 1. It can monitor temporal features on different scales. 2. It can perform gesture spotting in uncontrolled environments with multiple hand candidates. Since it is a forward spotting scheme, it is not able to perform real-time gesture spotting. A backward searching scheme which can detect the ending point of the meaningful gestures firstly, then searching backward for the starting point could potentially make the spotting scheme capable of real-time gesture spotting.

Chapter 7

Conclusions and Future Works

Computer vision method that enables people to communicate with computers intuitively on the semantic level is one of the most promising directions for the next generation of Human Computer Interaction. Using computer vision techniques to recognise hand gestures is getting accepted by the general population as an alternative way to control machines. However, the existing Hand Gesture Recognition technologies are confined by various challenges from the unconstrained environments.

In this thesis, a general framework for Hand Gesture Recognition is introduced to tackle the challenges from uncontrolled environments, include changing illumination, multiple skin-coloured regions moving in the background with complex texture, performers wearing short-sleeve and frontal occlusion. Gesture scale, speed and location invariance are also achieved. This framework is capable of performing Hand Posture Recognition, Hand Gesture Recognition and Hand Gesture Spotting in the uncontrolled environments.

7.1 Conclusions

A comprehensive framework for HGR, HPR and HGS in uncontrolled environments is proposed in this thesis. For Hand Posture Recognition, a novel boosting-based method is proposed. The main contribution of this method over the existing methods is the capability of selecting optimised set of texture features to represent the pre-defined postures against each other and the cluttered background. For Hand Posture Recognition with complex backgrounds, the Speeded Up Robust Features are used as the texture key points. Due to the large number of candidate texture key points, Adaptive Boosting is also used as a feature selection method. In this way, among all local texture features, the feature selection method can pick out effective local features through iterative validations on the training set. The selected local features are then combined to synthesise the strong classifiers for the gesture classes. This method is tested on the Triesch Hand Posture Database, and outperformed the state-out-the-art methods.

A novel tracking scheme called Adaptive SURF Tracking is introduced to extract hand trajectories for gesture classification in the uncontrolled environments with multiple distractions in the background. The main advantages of this method over the existing methods are: 1. No need for any segmentation process. 2. Capablility of dealing with multiple hand candidates. 3. It can adapt to various lighting conditions, gesture scales, speed and locations. The tracking method detects human facial regions in the first frame to estimate the skin-colour tone under the current lighting condition. Then large connected skin-coloured regions are picked out as the hand candidates. For the rest of the video stream, the trajectories of all hand candidates are recorded. This real-time tracking scheme is robust against speed, location and scale variance of the hand trajectories. Trajectory features are then extracted from the trajectories.

The trajectory features are then put into a novel gesture classifier called

Partition Matrix as the input observation sequences. The main advantages of this method over the existing methods are: 1. It can monitor temporal features on different scales. 2. Capablity of dealing with multiple hand candidates. Partition Matrix utilises the partition functions within the Hidden Conditional Random Fields to evaluate all hand candidates. Partition values are calculated from trajectories of all hand candidates under different frame rates, to distinguish the trajectory features of the signing hand. The key concept is making use of varying tracking results of randomly moving background distractions, by applying different frame rates on the original video. Compared with the background distractions, the variance of texture and trajectory patterns of the signing hand under different frame rates is relatively small. Hence, in the matrix of all partition values, the target signing hand would present a relatively consistent pattern. The classifier is tested on two datasets with severely distracted background settings and produced satisfactory experimental results.

As a branch of Hand Gesture Recognition, Hand Gesture Spotting is the task of detecting and segmenting single hand gestures within a continuously signed gesture sequence. A forward Hand Gesture Spotting method is introduced specifically for uncontrolled environments. The main advantages of this spotting method over the existing spotting schemes are: 1. Capability of monitoring temporal features on different scales. 2. It can perform gesture spotting in uncontrolled environments with multiple hand candidates in the scene. The spotting method inherits the Partition Matrix to evaluate the possibilities of the current frame being part of the predefined gesture classes. A garbage model is built after the training process. The garbage model is used to evaluate the possibility of the current hand trajectories being meaningless hand movements. This spotting method is tested on the Warwick Hand Gesture Database, and produced decent accuracy.

7.2 Limitations and Future Works

There are still many unsolved issues in the proposed framework for HGR, HPR and HGS in uncontrolled environments. Main limitations of the framework are listed below with possible corresponding solutions as the future works.

- Vocabulary Structure: Currently the proposed framework has only been tested on databases with small vocabularies, namely datasets of 10 hand-signed digits. Robustness of the framework against large vocabulary is unknown. Sub-gesture reasoning methods could potentially enhance the framework's ability to handle vocabularies with high level of intra-class variance and low level of inter-class variance. Hand-signed letters could potentially be a viable choice of vocabulary.

- Scene Settings: More challenges should be considered for Hand Posture Recognition in the unconstrained scene settings. Currently the framework's Hand Posture Recognition method is vulnerable against background distractions and partial frontal occlusions. Mechanisms that are capable of adaptively matching the non-occlusion hand regions with the predefined posture classes should be included in the framework. Concepts of some Face Recognition methods that are specifically designed to handle partial occlusions could be used as solutions.

- Texture Features: More texture features could be tested in the Adaptive SURF Tracking method, including BRISK, ORB and FREAK (more details please see Section 2.2.3.2), for less computational cost on feature extraction and texture matching process.

- Optimisation in the CRF Models: Various optimisation methods could be adopted in the training process of the CRF models. In the HCRF model, the likelihood functions are no longer guaranteed to be convex. The initialisation

of the optimisation search becomes vital to the over-all performance of the model. A strategy of estimating the optimised initialisation of the weight vector should be considered.

- Latent Dynamic Condition Random Fields (LDCRF): LDCRF should be tested as an alternative model of HCRF. LDCRF is capable of producing class label for every observation state. Compared with HCRF, it is more suitable for the HGS task. A forward spotting method for the LDCRF model could be potentially powerful for HGS in uncontrolled environments.

- Real-time Backward Spotting Method: The HGS method of the proposed framework is currently unable to perform real-time HGS in uncontrolled environments, since it is a forward spotting method. The sliding window structure and the garbage model in the proposed HGS method can be improved to form a robust real-time backwards HGS method.

- Estimation on Number of Hidden States in CRF Models: The number of hidden states in the HCRF and LDCRF models has great influence on the over-all model performance. Currently various values have to be tested through experiments to locate the optimum. The experiment process is rather time-consuming due to the long off-line training time. A single training session could take dozens of hours to converge. A method that can estimate the optimised amount of hidden states could be developed to narrow down the possible choices. One of the possible solutions is to analyse the feature functions in the potential function of the HCRF and LDCRF models. Similar with the calculation of the garbage model in the proposed framework, a predicted structure of the predefined gesture classes could be calculated from the training samples. Thereby the estimated number of "strokes" in the vocabulary could be computed as part of the off-line training process.

- Tang et al. [179] proposed a forest-based method called Latent Regression

Forest for real-time 3D hand pose estimation from single depth image. The problem of 3D articulated hand pose estimation is treated as a structured coarse-to-fine search for the skeletal joints. Latent tree model is then used to learn the granularity of each search stage. The Latent Regression Forest is used on the whole image instead of individual pixels, which largely improves the run-time speed. The Latent Tree Model which the Latent Regression Forest method built upon is modelling the underlining dependencies of the observations in the similar way as the HCRF model. It has been used in many areas in the computer vision community. It should be tested for the HGR in uncontrolled environments task as well.

- Deep Learning methods: Firstly, currently the Partition Matrix (Chapter 5) is monitoring the temporal features on different scales. But the temporal features are man-made and trained only on a single scale. Deep Learning methods generate optimised features on different scales in the training stage. The Partition Matrix could benefit from a multi-layer HCRF model similar with RNN, which trains the latent variables on different scales. Secondly, a score lever fusion method could be beneficial for the Partition Matrix. Instead just using one initial classifier to calculate the scores in each cell of the Partition Matrix, multiple classifiers could be used including deep learning methods to generate multiple initial scores for each cell.

Bibliography

[1] Agnes Just, Yann Rodriguez, and Sebastien Marcel. Hand posture classification and recognition using the modified census transform. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 351–356. IEEE, 2006.

[2] Chieh-Chih Wang and Ko-Chih Wang. Hand posture recognition using adaboost with sift for human robot interaction. In *Recent progress in robotics: viable robotic service to human*, pages 317–329. Springer, 2008.

[3] Mahmoud Elmezain, Ayoub Al-Hamadi, Samy Sadek, and Bernd Michaelis. Robust methods for hand gesture spotting and recognition using hidden markov models and conditional random fields. In *Signal Processing and Information Technology (ISSPIT), 2010 IEEE International Symposium on*, pages 131–136. IEEE, 2010.

[4] Yi Yao and Chang-Tsun Li. Real-time hand gesture recognition for uncontrolled environments using adaptive surf tracking and hidden conditional random fields. *Advances in Visual Computing*, pages 542–551, 2013.

[5] American Sign Language Dictionary. word Bicycle in ASL. `http://www.lifeprint.com/index.htm`, 2013. [Online; accessed July-2014].

[6] Sean Chen and Evan Levine. Mister Gloves - A Wireless USB Gesture In-

put System. https://courses.cit.cornell.edu/ee476/FinalProjects/s2010/ssc88_egl27/References, 2013. [Online; accessed July-2014].

[7] Jochen Triesch and Christoph Von Der Malsburg. Robust classification of hand postures against complex backgrounds. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 170–170. IEEE Computer Society, 1996.

[8] Jonathan Alon, Vassilis Athitsos, Quan Yuan, and Stan Sclaroff. A unified framework for gesture recognition and spatiotemporal gesture segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9):1685–1699, 2009.

[9] Mauricio Correa, Javier Ruiz-del Solar, Rodrigo Verschae, Jong Lee-Ferng, and Nelson Castillo. Real-time hand gesture recognition for human robot interaction. In *RoboCup 2009: Robot Soccer World Cup XIII*, pages 46–57. Springer, 2010.

[10] Manavender R Malgireddy, Ifeoma Nwogu, Subarna Ghosh, and Venu Govindaraju. A shared parameter model for gesture and sub-gesture analysis. In *Combinatorial Image Analysis*, pages 483–493. Springer, 2011.

[11] Shackel B. and Richardson S. Human factors for informatics usability. *Cambridge University Press*, 1991.

[12] Diaper Dan and Sanger Colston. Tasks for and tasks in humancomputer interaction. *Interacting with Computers*, 18(7):117–138, 2006.

[13] B. Shackel. Ergonomics for a computer. *Design*, 120:36–39, 1959.

[14] Columbia University Computing History. The IBM 610 Auto-Point Computer. http://www.columbia.edu/cu/computinghistory/610.html, 2014. [Online; accessed July-2014].

[15] CATHERINE G. WOLF. A comparative study of gestural, keyboard, and mouse interfaces. *Behaviour and Information Technology*, 11(1):13–23, 1992.

[16] Microsoft. Kinect. `http://www.microsoft.com/en-us/kinectforwindows/`, 2009. [Online; accessed July-2014].

[17] ThalmicLabs. MYO armband. `https://www.thalmic.com/en/myo/`, 2014. [Online; accessed July-2014].

[18] LeapMotion Inc. Leap Motion Controller. `https://www.leapmotion.com/`, 2010. [Online; accessed July-2014].

[19] A.Jaimes and N. Sebe. Multimodal human computer interaction: A survey. *Computer Vision and Image Understanding*, 108:116 – 134, 2007.

[20] J. Nespoulous, P. Perron, and A. R. Lecours. The biological foundations of gestures: Motor and semiotic aspects. *New Jersey London: Lawrence Erlbaum associates*, 1986.

[21] William C Stokoe, Dorothy C Casterline, and Carl G Croneberg. *A dictionary of American Sign Language on linguistic principles*. Linstok Press Silver Spring, 1976.

[22] David Brien, British Deaf Association, et al. *Dictionary of British Sign Language English*. Faber & Faber, 1992.

[23] Chunli Wang, Wen Gao, and Shiguang Shan. An approach based on phonemes to large vocabulary chinese sign language recognition. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 411–416. IEEE, 2002.

[24] Quan Yuan, Stan Sclaroff, and Vassilis Athitsos. Automatic 2d hand tracking in video sequences. In *Application of Computer Vision, 2005.*

WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on, volume 1, pages 250–256. IEEE, 2005.

[25] Alejandro Jaimes and Nicu Sebe. Multimodal human–computer interaction: A survey. *Computer vision and image understanding*, 108(1):116–134, 2007.

[26] Kai Nickel, Edgar Scemann, and Rainer Stiefelhagen. 3d-tracking of head and hands for pointing gesture recognition in a human-robot interaction scenario. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 565–570. IEEE, 2004.

[27] Toshiyuki Kirishima, Kosuke Sato, and Kunihiro Chihara. Real-time gesture recognition by learning and selective control of visual interest points. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(3):351–364, 2005.

[28] Sébastien Carbini, Jean Emmanuel Viallet, and Olivier Bernier. Pointing gesture visual recognition for large display. In *FG Net Workshop on Visual Observation of Deictic Gestures*, 2004.

[29] Sanparith Marukatat, Thierry Artieres, and Patrick Gallinari. A generic approach for on-loine handwriting recognition. 2004.

[30] Beat Signer, Ueli Kurmann, and Moira C Norrie. igesture: a general gesture recognition framework. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 954–958. IEEE, 2007.

[31] Jochen Triesch and Christoph Von Der Malsburg. A system for person-independent hand posture recognition against complex backgrounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(12):1449–1453, 2001.

[32] Sébastien Marcel. Hand posture recognition in a body-face centered space. In *CHI'99 Extended Abstracts on Human Factors in Computing Systems*, pages 302–303. ACM, 1999.

[33] Sotiris Malassiotis and Michael G Strintzis. Real-time hand posture recognition using range data. *Image and Vision Computing*, 26(7):1027–1037, 2008.

[34] Lars Bretzner, Ivan Laptev, and Tony Lindeberg. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 423–428. IEEE, 2002.

[35] Agnes Just, Yann Rodriguez, and Sebastien Marcel. Hand posture classification and recognition using the modified census transform. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 351–356. IEEE, 2006.

[36] Xiaoming Yin and Ming Xie. Finger identification and hand posture recognition for human–robot interaction. *Image and Vision Computing*, 25(8):1291–1300, 2007.

[37] Yuelong Chuang, Ling Chen, and Gencai Chen. Saliency-guided improvement for hand posture detection and recognition. *Neurocomputing*, 2014.

[38] Yea Shuan Huang and Yun Jiun Wang. A neural-network-based hand posture recognition method. In *Transactions on Engineering Technologies*, pages 187–201. Springer, 2014.

[39] William T Freeman and Michal Roth. Orientation histograms for hand gesture recognition. In *International Workshop on Automatic Face and Gesture Recognition*, volume 12, pages 296–301, 1995.

[40] Lars Bretzner, Ivan Laptev, and Tony Lindeberg. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 423–428. IEEE, 2002.

[41] Feng-Sheng Chen, Chih-Ming Fu, and Chung-Lin Huang. Hand gesture recognition using a real-time tracking method and hidden markov models. *Image and Vision Computing*, 21(8):745–758, 2003.

[42] Arnaud Lemoine, Sean Mcgrath, Anthony Vernon Walker Smith, and Alistair Ian Sutherland. Hand gesture recognition system and method, October 3 2000. US Patent 6,128,003.

[43] Zhou Ren, Jingjing Meng, Junsong Yuan, and Zhengyou Zhang. Robust hand gesture recognition with kinect sensor. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 759–760. ACM, 2011.

[44] 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 the 19th ACM international conference on Multimedia*, pages 1093–1096. ACM, 2011.

[45] Sung-Ho Im, Dong-Sun Lim, Tae-Joon Park, Kee-Koo Kwon, Man-Seok Yang, and Heung-Nam Kim. User interface apparatus using hand gesture recognition and method thereof, April 20 2010. US Patent 7,702,130.

[46] KS Chidanand Kumar. Segmentation and feature space analysis for hand gesture recognition under complex background. *Science*, 3(3), 2012.

[47] Thad E Starner. Visual recognition of american sign language using hidden markov models. Technical report, DTIC Document, 1995.

[48] Richard A Bolt. *Put-that-there: Voice and gesture at the graphics interface*, volume 14. ACM, 1980.

[49] Thomas G Zimmerman, Jaron Lanier, Chuck Blanchard, Steve Bryson, and Young Harvill. A hand gesture interface device. In *ACM SIGCHI Bulletin*, volume 18, pages 189–192. ACM, 1987.

[50] David J Sturman, David Zeltzer, and Steve Pieper. Hands-on interaction with virtual environments. In *Proceedings of the 2nd annual ACM SIGGRAPH symposium on User interface software and technology*, pages 19–24. ACM, 1989.

[51] Alexander G Hauptmann. Speech and gestures for graphic image manipulation. In *ACM SIGCHI Bulletin*, volume 20, pages 241–245. ACM, 1989.

[52] David L Quam. Gesture recognition with a dataglove. In *Aerospace and Electronics Conference, 1990. NAECON 1990., Proceedings of the IEEE 1990 National*, pages 755–760. IEEE, 1990.

[53] Alexander G Hauptmann and Paul McAvinney. Gestures with speech for graphic manipulation. *International Journal of Man-Machine Studies*, 38(2):231–249, 1993.

[54] Mahmoud Elmezain, Ayoub Al-Hamadi, and Bernd Michaelis. A robust method for hand gesture segmentation and recognition using forward spotting scheme in conditional random fields. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3850–3853. IEEE, 2010.

[55] Bozon, Mark. Nintendo Sets the Record Straight. `http://www.nintendo.com/wiiu;jsessionid=D5F2C11D626712CE04E2F54043807248`, 2006. [Online; accessed July-2014].

[56] Anbumani Subramanian, Vinod Pathangay, and Dinesh Mandalapu. Hand gesture recognition. 2010.

[57] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[58] Neidle, C. Boston ASL dataset. `http://www.bu.edu/asllrp/cslgr/`, 2006. [Online; accessed July-2014].

[59] J Kenneth Salisbury and John J Craig. Articulated hands force control and kinematic issues. *The International Journal of Robotics Research*, 1(1):4–17, 1982.

[60] James M Rehg, Daniel D Morris, and Takeo Kanade. Ambiguities in visual tracking of articulated objects using two-and three-dimensional models. *The International Journal of Robotics Research*, 22(6):393–418, 2003.

[61] James M Rehg and Takeo Kanade. Digiteyes: Vision-based hand tracking for human-computer interaction. In *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on*, pages 16–22. IEEE, 1994.

[62] William C Stokoe. Sign language structure. 1978.

[63] Philippe Dreuw, Thomas Deselaers, David Rybach, Daniel Keysers, and Hermann Ney. Tracking using dynamic programming for appearance-based sign language recognition. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 293–298. IEEE, 2006.

[64] Kikuo Fujimura and Xia Liu. Sign recognition using depth image streams. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 381–386. IEEE, 2006.

[65] Kai Nickel, Edgar Scemann, and Rainer Stiefelhagen. 3d-tracking of head and hands for pointing gesture recognition in a human-robot interaction scenario.

In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 565–570. IEEE, 2004.

[66] Richard Bowden, David Windridge, Timor Kadir, Andrew Zisserman, and Michael Brady. A linguistic feature vector for the visual interpretation of sign language. In *Computer Vision-ECCV 2004*, pages 390–401. Springer, 2004.

[67] Richard Bowden, Andrew Zisserman, Timor Kadir, and Mike Brady. Vision based interpretation of natural sign languages. 2003.

[68] Sushmita Mitra and Tinku Acharya. Gesture recognition: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(3):311–324, 2007.

[69] Rogerio Feris, Matthew Turk, Ramesh Raskar, Kar-Han Tan, and Gosuke Ohashi. Recognition of isolated fingerspelling gestures using depth edges. In *Real-Time Vision for Human-Computer Interaction*, pages 43–56. Springer, 2005.

[70] Omar Al-Jarrah and Alaa Halawani. Recognition of gestures in arabic sign language using neuro-fuzzy systems. *Artificial Intelligence*, 133(1):117–138, 2001.

[71] Scott K Liddell. *Grammar, gesture, and meaning in American Sign Language*. Cambridge University Press, 2003.

[72] Chung-Hsien Wu, Yu-Hsien Chiu, and Kung-Wei Cheng. Error-tolerant sign retrieval using visual features and maximum a posteriori estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(4):495–508, 2004.

[73] Hee-Deok Yang and Seong-Whan Lee. Simultaneous spotting of signs and

fingerspellings based on hierarchical conditional random fields and boostmap embeddings. *Pattern Recognition*, 43(8):2858–2870, 2010.

[74] Hee-Deok Yang and Seong-Whan Lee. Robust sign language recognition with hierarchical conditional random fields. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2202–2205. IEEE, 2010.

[75] Itay Katz. EyeSight Technology. `http://eyesight-tech.com/`, 2014. [Online; accessed July-2014].

[76] DANIEL VAN NIEUWENHOVE. Softkinetic. `http://www.softkinetic.com/en-us/softkinetic.aspx`, 2014. [Online; accessed July-2014].

[77] Mahmoud Elmezain and Ayoub Al-Hamadi. Ldcrfs-based hand gesture recognition. In *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, pages 2670–2675. IEEE, 2012.

[78] PointGrab Ltd. pointgrab. `http://www.pointgrab.com/`, 2014. [Online; accessed July-2014].

[79] Faisal Yazadi and Mark Schelbert. Cyber Glove Systems - worldwide leader in data glove technology. `http://www.cyberglovesystems.com/index.php`, 2010. [Online; accessed July-2014].

[80] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *Computer Vision–ECCV 2012*, pages 852–863. Springer, 2012.

[81] Stan Melax, Leonid Keselman, and Sterling Orsten. Dynamics based 3d skeletal hand tracking. In *Proceedings of Graphics Interface 2013*, pages 63–70. Canadian Information Processing Society, 2013.

[82] Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys.

Motion capture of hands in action using discriminative salient points. In *Computer Vision–ECCV 2012*, pages 640–653. Springer, 2012.

[83] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[84] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. Real time hand pose estimation using depth sensors. In *Consumer Depth Cameras for Computer Vision*, pages 119–137. Springer, 2013.

[85] Xiaojiang Peng, Limin Wang, Zhuowei Cai, and Yu Qiao. Action and gesture temporal spotting with super vector representation. In *ECCV Workshops*, 2014.

[86] Yong Pei, Bingbing Ni, and Indriyati Atmosukarto. Mixture of heterogeneous attribute analyzers for human action detection. In *Computer Vision-ECCV 2014 Workshops*, pages 528–540. Springer, 2014.

[87] Andrew D Bagdanov, Alberto Del Bimbo, Lorenzo Seidenari, and Lorenzo Usai. Real-time hand status recognition from rgb-d imagery. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2456–2459. IEEE, 2012.

[88] Ross Girshick, Jamie Shotton, Pushmeet Kohli, Antonio Criminisi, and Andrew Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 415–422. IEEE, 2011.

[89] Sangheon Park, Sunjin Yu, Joongrock Kim, Sungjin Kim, and Sangyoun Lee. 3d hand tracking using kalman filter in depth space. *EURASIP Journal on Advances in Signal Processing*, 2012(1):1–18, 2012.

[90] Matthieu Bray, Esther Koller-Meier, Pascal Müller, Luc Van Gool, and Nicol N Schraudolph. 3d hand tracking by rapid stochastic gradient descent using a skinning model. In *In 1st European Conference on Visual Media Production (CVMP*. Citeseer, 2004.

[91] Michael J Jones and James M Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.

[92] Vladimir Vezhnevets, Vassili Sazonov, and Alla Andreeva. A survey on pixel-based skin color detection techniques. In *Proc. Graphicon*, volume 3, pages 85–92. Moscow, Russia, 2003.

[93] Jure Kovac, Peter Peer, and Franc Solina. *Human skin color clustering for face detection*, volume 2. IEEE, 2003.

[94] Alberto Albiol, Luis Torres, and Edward J Delp. Optimum color spaces for skin detection. In *ICIP (1)*, pages 122–124, 2001.

[95] Maricor Soriano, Birgitta Martinkauppi, Sami Huovinen, and Mika Laaksonen. Skin detection in video under changing illumination conditions. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 1, pages 839–842. IEEE, 2000.

[96] Benjamin D Zarit, Boaz J Super, and Francis KH Quek. Comparison of five color models in skin pixel classification. In *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 1999. Proceedings. International Workshop on*, pages 58–63. IEEE, 1999.

[97] Kenny Morrison and Stephen J McKenna. An experimental comparison of trajectory-based and history-based representation for gesture recognition. In *Gesture-Based Communication in Human-Computer Interaction*, pages 152–163. Springer, 2004.

[98] Benjamin D Zarit, Boaz J Super, and Francis KH Quek. Comparison of five color models in skin pixel classification. In *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 1999. Proceedings. International Workshop on*, pages 58–63. IEEE, 1999.

[99] Stan Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 232–237. IEEE, 1998.

[100] Wen-Hsiang Lai and Chang-Tsun Li. Skin colour-based face detection in colour images. In *Video and Signal Based Surveillance, 2006. AVSS'06. IEEE International Conference on*, pages 56–56. IEEE, 2006.

[101] D. Tang, H.J. Chang, A. Tejani, and T-K. Kim. Latent regression forest: Structural estimation of 3d articulated hand posture. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014.

[102] Sameh Khamis12, Jonathan Taylor, Jamie Shotton, Cem Keskin, Shahram Izadi, and Andrew Fitzgibbon. Learning an efficient model of hand shape variation from depth images. 2015.

[103] Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32–40, 1975.

[104] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 142–149. IEEE, 2000.

[105] Changjiang Yang, Ramani Duraiswami, and Larry Davis. Efficient mean-shift tracking via a new similarity measure. In *Computer Vision and Pattern Recog-*

nition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 176–183. IEEE, 2005.

[106] Gary R Bradski. Computer vision face tracking for use in a perceptual user interface. 1998.

[107] John G Allen, Richard YD Xu, and Jesse S Jin. Object tracking using camshift algorithm and multiple quantized feature spaces. In *Proceedings of the Pan-Sydney area workshop on Visual information processing*, pages 3–7. Australian Computer Society, Inc., 2004.

[108] Zhaowen Wang, Xiaokang Yang, Yi Xu, and Songyu Yu. Camshift guided particle filter for visual tracking. *Pattern Recognition Letters*, 30(4):407–413, 2009.

[109] O-D Nouar, Ganoun Ali, and CANALS Raphael. Improved object tracking with camshift algorithm. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 2, pages II–II. IEEE, 2006.

[110] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe. A boosted particle filter: Multitarget detection and tracking. In *Computer Vision-ECCV 2004*, pages 28–39. Springer, 2004.

[111] Changjiang Yang, Ramani Duraiswami, and Larry Davis. Fast multiple object tracking via a hierarchical particle filter. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 212–219. IEEE, 2005.

[112] Zia Khan, Tucker Balch, and Frank Dellaert. An mcmc-based particle filter for tracking multiple interacting targets. In *Computer Vision-ECCV 2004*, pages 279–290. Springer, 2004.

[113] Caifeng Shan, Tieniu Tan, and Yucheng Wei. Real-time hand tracking using a mean shift embedded particle filter. *Pattern Recognition*, 40(7):1958–1970, 2007.

[114] Björn Stenger, Arasanathan Thayananthan, Philip HS Torr, and Roberto Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(9):1372–1384, 2006.

[115] Björn Stenger, Arasanathan Thayananthan, Philip HS Torr, and Roberto Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(9):1372–1384, 2006.

[116] Berthold K Horn and Brian G Schunck. Determining optical flow. In *1981 Technical Symposium East*, pages 319–331. International Society for Optics and Photonics, 1981.

[117] John L Barron, David J Fleet, and Steven S Beauchemin. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77, 1994.

[118] James M Rehg and Takeo Kanade. Visual tracking of high dof articulated structures: an application to human hand tracking. In *Computer VisionECCV'94*, pages 35–46. Springer, 1994.

[119] Shan Lu, Dimitris Metaxas, Dimitris Samaras, and John Oliensis. Using multiple cues for hand tracking and model refinement. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–443. IEEE, 2003.

[120] Timothy P Wallace and Paul A Wintz. An efficient three-dimensional air-

craft recognition algorithm using normalized fourier descriptors. *Computer Graphics and Image Processing*, 13(2):99–126, 1980.

[121] Guangyi Chen and Tien D Bui. Invariant fourier-wavelet descriptor for pattern recognition. *Pattern recognition*, 32(7):1083–1088, 1999.

[122] GC-H Chuang and C-CJ Kuo. Wavelet descriptor of planar curves: Theory and applications. *Image Processing, IEEE Transactions on*, 5(1):56–70, 1996.

[123] Chia-Hung Wei, Yue Li, Wing-Yin Chau, and Chang-Tsun Li. Trademark image retrieval using synthetic features for describing global shape and interior structure. *Pattern Recognition*, 42(3):386–394, 2009.

[124] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[125] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[126] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.

[127] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010*, pages 778–792. Springer, 2010.

[128] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

[129] Stefan Leutenegger, Margarita Chli, and Roland Yves Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555. IEEE, 2011.

[130] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1508–1515. IEEE, 2005.

[131] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517. Ieee, 2012.

[132] Takuichi Nishimura and Ryuichi Oka. Spotting recognition of human gestures from time-varying images. In *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, pages 318–322. IEEE, 1996.

[133] Jonathan Alon, Vassilis Athitsos, Quan Yuan, and Stan Sclaroff. Simultaneous localization and recognition of dynamic hand gestures. In *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*, volume 2, pages 254–260. IEEE, 2005.

[134] Lawrence Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[135] Thad Starner and Alex Pentland. Real-time american sign language recognition from video using hidden markov models. In *Motion-Based Recognition*, pages 227–243. Springer, 1997.

[136] Feng-Sheng Chen, Chih-Ming Fu, and Chung-Lin Huang. Hand gesture recognition using a real-time tracking method and hidden markov models. *Image and Vision Computing*, 21(8):745–758, 2003.

[137] Matthew Brand, Nuria Oliver, and Alex Pentland. Coupled hidden markov models for complex action recognition. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 994–999. IEEE, 1997.

[138] Thad Starner, Joshua Weaver, and Alex Pentland. Real-time american sign language recognition using desk and wearable computer based video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12):1371–1375, 1998.

[139] Mahmoud Elmezain, Ayoub Al-Hamadi, Jörg Appenrodt, and Bernd Michaelis. A hidden markov model-based continuous gesture recognition system for hand motion trajectory. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[140] Matthew Brand, Nuria Oliver, and Alex Pentland. Coupled hidden markov models for complex action recognition. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 994–999. IEEE, 1997.

[141] Andrew D Wilson and Aaron F Bobick. Parametric hidden markov models for gesture recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(9):884–900, 1999.

[142] Dan C Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1237, 2011.

[143] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[144] Patrice Y Simard, Dave Steinkraus, and John C Platt. Best practices for convolutional neural networks applied to visual document analysis. In *2013 12th International Conference on Document Analysis and Recognition*, volume 2, pages 958–958. IEEE Computer Society, 2003.

[145] Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.

[146] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001.

[147] Roland Wilson and Chang-Tsun Li. A class of discrete multiresolution random fields and its application to image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(1):42–56, 2003.

[148] Ariadna Quattoni, Sybor Wang, Louis-Philippe Morency, Michael Collins, and Trevor Darrell. Hidden conditional random fields. *IEEE transactions on pattern analysis and machine intelligence*, 29(10):1848, 2007.

[149] Sy Bor Wang, Ariadna Quattoni, L Morency, David Demirdjian, and Trevor Darrell. Hidden conditional random fields for gesture recognition. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1521–1527. IEEE, 2006.

[150] L Morency, Ariadna Quattoni, and Trevor Darrell. Latent-dynamic discriminative models for continuous gesture recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[151] Gang Yu, Junsong Yuan, and Zicheng Liu. Propagative hough voting for human activity recognition. In *Computer Vision–ECCV 2012*, pages 693–706. Springer, 2012.

[152] MS Ryoo. Human activity prediction: Early recognition of ongoing activities

from streaming videos. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1036–1043. IEEE, 2011.

[153] Juergen Gall, Angela Yao, Nima Razavi, Luc Van Gool, and Victor Lempitsky. Hough forests for object detection, tracking, and action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11):2188–2202, 2011.

[154] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.

[155] Yikai Fang, Jian Cheng, Jinqiao Wang, Kongqiao Wang, Jing Liu, and Hanqing Lu. Hand posture recognition with co-training. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[156] P Pramod Kumar, Prahlad Vadakkepat, and Loh Ai Poh. Graph matching based hand posture recognition using neuro-biologically inspired features. In *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*, pages 1151–1156. IEEE, 2010.

[157] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[158] Michael J Jones and James M Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.

[159] Axel Bernal, Koby Crammer, Artemis Hatzigeorgiou, and Fernando Pereira. Global discriminative learning for higher-accuracy computational gene prediction. *PLoS computational biology*, 3(3):e54, 2007.

[160] MS Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1036–1043. IEEE, 2011.

[161] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 129–136, 2011.

[162] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics, 2011.

[163] Xingjie Wei, C Li, Zhen Lei, Dong Yi, and Stan Li. Dynamic image-to-class warping for occluded face recognition. 2014.

[164] Xuming He, Richard S Zemel, and MA Carreira-Perpindn. Multiscale conditional random fields for image labeling. In *Computer vision and pattern recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE computer society conference on*, volume 2, pages II–695. IEEE, 2004.

[165] Ariadna Quattoni, Michael Collins, and Trevor Darrell. Conditional random fields for object recognition. In *Advances in neural information processing systems*, pages 1097–1104, 2004.

[166] Sy Bor Wang, Ariadna Quattoni, L Morency, David Demirdjian, and Trevor Darrell. Hidden conditional random fields for gesture recognition. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1521–1527. IEEE, 2006.

[167] Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. Maximum entropy markov models for information extraction and segmentation. In *ICML*, pages 591–598, 2000.

[168] Adwait Ratnaparkhi et al. A maximum entropy model for part-of-speech tagging. In *Proceedings of the conference on empirical methods in natural language processing*, volume 1, pages 133–142. Philadelphia, PA, 1996.

[169] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[170] Charles Sutton and Andrew McCallum. An introduction to conditional random fields for relational learning. *Introduction to statistical relational learning*, pages 93–128, 2006.

[171] Charles Sutton, Andrew McCallum, and Khashayar Rohanimanesh. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *The Journal of Machine Learning Research*, 8:693–723, 2007.

[172] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[173] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2009.

[174] Richard Bellman and Robert Kalaba. On adaptive control processes. *Automatic Control, IRE Transactions on*, 4(2):1–9, 1959.

[175] Sen M Kuo, Bob H Lee, and Wenshun Tian. *Real-Time Digital Signal Processing: Fundamentals, Implementations and Applications*. John Wiley & Sons, 2013.

[176] Richard C Rose, Edward M Hofstetter, and Douglas A Reynolds. Integrated models of signal and background with application to speaker identification in noise. *Speech and Audio Processing, IEEE Transactions on*, 2(2):245–257, 1994.

[177] A_P Varga and RK Moore. Hidden markov model decomposition of speech and noise. In *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, pages 845–848. IEEE, 1990.

[178] Steven Boll. Suppression of acoustic noise in speech using spectral subtraction. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 27(2):113–120, 1979.

[179] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3786–3793. IEEE, 2014.