Terms and Conditions of Use of Digitised Theses from Trinity College Library Dublin

Copyright statement

All material supplied by Trinity College Library is protected by copyright (under the Copyright and Related Rights Act, 2000 as amended) and other relevant Intellectual Property Rights. By accessing and using a Digitised Thesis from Trinity College Library you acknowledge that all Intellectual Property Rights in any Works supplied are the sole and exclusive property of the copyright and/or other IPR holder. Specific copyright holders may not be explicitly identified. Use of materials from other sources within a thesis should not be construed as a claim over them.

A non-exclusive, non-transferable licence is hereby granted to those using or reproducing, in whole or in part, the material for valid purposes, providing the copyright owners are acknowledged using the normal conventions. Where specific permission to use material is required, this is identified and such permission must be sought from the copyright holder or agency cited.

Liability statement

By using a Digitised Thesis, I accept that Trinity College Dublin bears no legal responsibility for the accuracy, legality or comprehensiveness of materials contained within the thesis, and that Trinity College Dublin accepts no liability for indirect, consequential, or incidental, damages or losses arising from use of the thesis for whatever reason. Information located in a thesis may be subject to specific use constraints, details of which may not be explicitly described. It is the responsibility of potential and actual users to be aware of such constraints and to abide by them. By making use of material from a digitised thesis, you accept these copyright and disclaimer provisions. Where it is brought to the attention of Trinity College Library that there may be a breach of copyright or other restraint, it is the policy to withdraw or take down access to a thesis while the issue is being resolved.

Access Agreement

By using a Digitised Thesis from Trinity College Library you are bound by the following Terms & Conditions. Please read them carefully.

I have read and I understand the following statement: All material supplied via a Digitised Thesis from Trinity College Library is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of a thesis is not permitted, except that material may be duplicated by you for your research use or for educational purposes in electronic or print form providing the copyright owners are acknowledged using the normal conventions. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone. This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Vision-based Hand Washing Gesture Recognition

A Dissertation

Submitted to the office of Graduate Studies

of

The University of Dublin

Trinity College

in fulfillment of the requirements

for the Degree of

Doctor of Philosophy

by

JIANG ZHOU, M.Eng.

April 2013

Declaration

This thesis has not been submitted as an exercise for a degree at this or any other University. Furthermore this thesis is entirely my own work and I agree that the Library may lend or copy the thesis upon request. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

JIANG ZHOU

29th April 2013

Vision-based Hand Washing Gesture Recognition

Abstract

Vision-based human gesture recognition has been studied for many years, however hand washing gesture recognition surprisingly remains untouched in the research. Strictly applying correct hand washing techniques is one of the most important ways to prevent the spread of infection and illness, especially in healthcare environment. In this thesis, we propose and evaluate a number of vision-based measuring methods as alternatives of laborious human monitoring of hand hygiene. The main goal of our research is to detect and recognize certain hand washing gesture patterns robustly and effectively.

Recognizing hand washing gestures with computer vision techniques is not an easy task. Hand washing gestures are articulated, fast-moving, bi-manual gestures. Uncertainties such as self-occlusion and lighting condition changes can also bring in large variance in gesture appearance. These factors would result in large intra-class variety and inter-class ambiguity during the recognition. To address these challenges, four vision-based methods are proposed and evaluated in order to study the hand washing gestures from different aspects. To the best of our knowledge, this is the first research of hand washing gestures using computer vision methods.

As one of our major contributions, the hand washing gestures are defined, analysed and recognized with four different types of computer vision methods. Although

these methods are developed and evaluated in the context of recognizing hand washing gestures, many conclusions are also applicable to other types of human action recognition. As another our major contribution, the newly developed method, Texton analysis with Graph-Cut Hidden Conditional Random Fields (TGC-HCRFs), provides a practical study of Conditional Random Fields (CRFs) with hidden variables for grid graphs in recognition.

Acknowledgments

I would like to thank my supervisor, Dr. Gerard Lacey, for his patient guidance throughout my Phd. study and great help during my thesis writing. I also would like to thank my parents for their constant generous support all the time.

JIANG ZHOU

The University of Dublin
Trinity College
April 2013

Contents

List of Acronyms

BOFs
Bags of Features.

CRFs
Conditional Random Fields.

DOF
Degrees of Freedom.

DOG
Difference of Gaussians.

EM
Expectation-Maximization.

ERC
Extremely Randomized Clustering.

FPGAs
Field-Programmable Gate Arrays.

fps
frames per second.

GPU
Graphics Processing Unit.

HAIs
Hospital Acquired Infections.

HCRFs
Hidden Conditional Random Fields.

HMM
Hidden Markov Model.

HOG
Histogram of Oriented Gradient.

LDA
Linear Discriminant Analysis.

MAP
Maximum A Posteriori.

MEMMs
Maximum Entropy Markov Models.

ML
Maximum Likelihood.

MRFs
Markov Random Fields.

MRSA
Methicillin-resistant Staphylococcus aureus.

PCA
Principal Component Analysis.

QP
quadratic programming.

QPBO
Quadratic Pseudo-Boolean Optimization.

ROI
Region of Interest.

s.t.
subject to.

SIFT
Scale Invariant Feature Transform.

SVM
Support Vector Machine.

TGC-HCRFs
Texton analysis with Graph-Cut Hidden Conditional Random Fields.

WHO
World Health Organization.

List of Tables

List of Figures

Chapter 1

Introduction

In our daily life, hand washing is a small but very important human activity. The very simple activity of frequent hand washing has the potential to save more lives than any single vaccine or medical intervention [55][41]. Hand washing for hand hygiene is one of the most important ways to prevent the spread of infection and illness, especially in healthcare environment. Overwhelming scientific evidence shows that Hospital Acquired Infections (HAIs) are transmitted by the hands of healthcare workers [134] and approximate 50% of these infections are preventable through better hand hygiene [170].

It has been known that parts of the hands are frequently missed during the hand washing [110] and a recent study reported that 3 - 6% of the hands of healthcare workers were contaminated with Methicillin-resistant Staphylococcus aureus (MRSA) after performing their hand-hygiene routine [40]. Therefore, correct hand washing technique is needed to avoid missing areas such as finger tips, thumbs, and between the fingers. In fact, the technique used in hand washing is more important than the amount of time spent [143]. A 2007 Swiss study identified that training in the World Health Organization (WHO) hand washing technique doubled the anti-microbial effectiveness of hand hygiene with alcohol gel [169]. Improved hand-hygiene training can also lead to a reduction in HAIs rate of between 20% and 50% [170].

Developing and maintaining a high standard of hand washing training and assessment is very difficult. Measuring the quality of the hand washing activity is one of the key steps. Several approaches have been proposed for the hand hygiene measurement [18]. Direct observation is one of the approaches, which is simple and easy to measure all types of hand hygiene including the hand washing activity. However, it is usually time consuming and costly. It can only provide information about a very low

Figure 1.1: *A hand washing recognition system*

percentage of all hand washing opportunities, and realistic comparison of rates between facilities is also impossible due to lack of standardized measurement [18]. Measuring product consumption and electronic hand hygiene compliance monitoring systems are some other approaches that can be used for measuring hand washing activity. However, those approaches are mainly for the measurement of hand washing frequency rates and WHO "5 moments of hand hygiene" [145]. None of them can provide the assessment of the hand washing technique performed during the hand washing activity.

1.1 Vision-based Hand Washing Gesture Recognition

In this thesis, we propose to measure the hand washing activity by means of hand washing gesture recognition with state-of-the-art Computer Vision techniques. Such an approach can assess the hand washing technique applied during the hand hygiene routine and enforce the correct hand washing gestures are followed when feedback is provided. It can facilitate the hand washing training and assessment with much less human resources but provide much more hand hygiene quality information. A prototype of hand washing recognition system is illustrated in figure 1.1. A top-down view camera is mounted above a sink with a certain angle. The captured hand washing activity videos are then processed locally.

Figure 1.2 outlines the main processing steps in the system. The image pre-

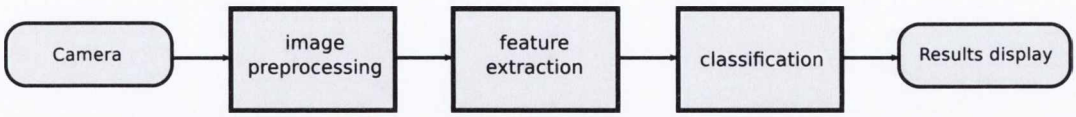


Figure 1.2: *Hand washing gesture recognition workflow*

processing step would apply some image processing techniques such as image colour adjusting or Region of Interest (ROI) detection to simplify the later classification job. The feature extraction step extracts local low-level information to characterize the hand washing gestures presented in the images. The classification step requires some machine learning [177][119][24] to build hand washing gesture models and classifies a new coming frame to be one of the predefined hand washing gestures. The classification results are displayed on a screen as real-time feedback to the person performing the hand washing. With the feedback, the system user notices what hand washing gestures are wrong and corrects them. This self-learning circle requires that the system has the ability to interpret and discriminate the multi-class hand washing gestures correctly and effectively in real-time, which is the a main goal of our study in this thesis.

1.1.1 Hand washing gestures

In order to assess the hand washing gestures, a standard needs to be set up first. In the thesis, we apply the WHO EN1500 hand washing protocol [170] as the guideline for our hand washing gesture recognition. 10 different hand washing gestures are defined as shown in figure 1.3.

- **Gesture 1** is rubbing hands palm to palm.
- **Gesture 2 and 3** are one palm over the other's dorsum with interlaced fingers moving back and forth.
- **Gesture 4** is rubbing palm to palm with fingers interlaced.
- **Gesture 5 and 6** are rotational rubbing of a thumb clasped in the other palm.
- **Gesture 7 and 8** are rotational rubbing backwards and forwards with clasped fingers of one hand in the other palm.
- **Gesture 9 and 10** are hands moving back and forth with backs of fingers opposing palms and fingers interlocked.

Figure 1.3: *Hand washing gesture definitions*

Among these gestures, some are quite dynamic such as gesture 5, 6, 7 and 8, and some are relatively static such as gesture 2, 3 and 9, 10. Thus, the vision-based methods need to take both dynamic and static sides of the hand washing gestures into account. It can also be seen that the difference between some gestures is minor such as gesture 2 and 3, and gesture 9 and 10. Gesture details may help distinguish these gestures in recognition, however they may also reduce the tolerance of gesture variety which is required in gesture 5, 6, 7 and 8. Therefore, our study needs to examine what approach is more suitable for representing and discriminating these 10 hand washing gestures, single image or sequences, locally or globally, dynamically or statically, etc.

As the hand washing gestures are non-rigid, fast, bi-manual hand movements, there are also some other challenges for the hand washing gesture recognition.

- Intra-class variations. Although 10 gesture definitions have been given above, in practice it is hard to categorize some hand movements into a same class definitively due to the large variance of a hand washing gesture. The same gesture may appear very differently when performed by different people or at different time from the same person. Gestures can also change the appearance due to the changes of lighting condition and viewpoint.

- Inter-class ambiguities. As mentioned above, some gesture definitions are close. The similar gesture definition can easily bring in class ambiguities during the

recognition. Meanwhile, our hand washing gestures are continuous hand movements. A lot of gesture transitions are captured in the videos. Hands in a single frame may thus appear belonging to many gestures if no contextual information is used.

- Unstable ROI detection. Localizing the hands in images may be required to reduce irrelevant information and simplify later classification. However, because hands are very flexible, it would be difficult to centralize hands in a ROI uniformly. Moreover, in uncontrolled environment cluttered background and illumination changes can also cause problems in hand tracking.

Figure 1.4 shows some examples of these challenges [1] . Image (a) and (b) demonstrate that the hands do not localize in ROI uniformly. In image (a), hands locate in the centre of the ROI while in image (b) the stretching out fingers push the main body of hands to the upper half of the ROI. Image (c) and (d) demonstrate the inter-class variety in gesture 8. Although the hand shapes look very different, they actually belong to the same hand washing gesture. Image (e) and (f) give an example of the inter-class ambiguity between gesture 1 and 4. Both gestures are palm to palm, and the only difference is whether the fingers are interlaced during the hand washing, which sometimes are hard to tell due to the fast hand movement. Image (g) and (h) show another example of ambiguity in gesture 10. Image (h) is captured during the gesture transition, which looks like image (g) but does not belong to gesture 10.

1.1.2 Methods

To deal with these challenges in hand washing gesture recognition, four vision-based methods, recognition with postures, recognition with sequence labelling, recognition with interest points and recognition with TGC-HCRFs, are evaluated and compared in this thesis. These four methods research into the hand washing gestures from different perspectives. The first method is adapted from methods for object recognition. The method ignores the dynamic nature of hand washing gestures and only considers the static information extracted from each frame independently. It describes the hand shape in each frame with Histogram of Oriented Gradient (HOG) features [121], and models every hand washing gesture as modelling an object class with Support Vector

[1] Some images shown in this thesis were obscure. We increase the brightness and contrast of these images for the purposes of illustration.

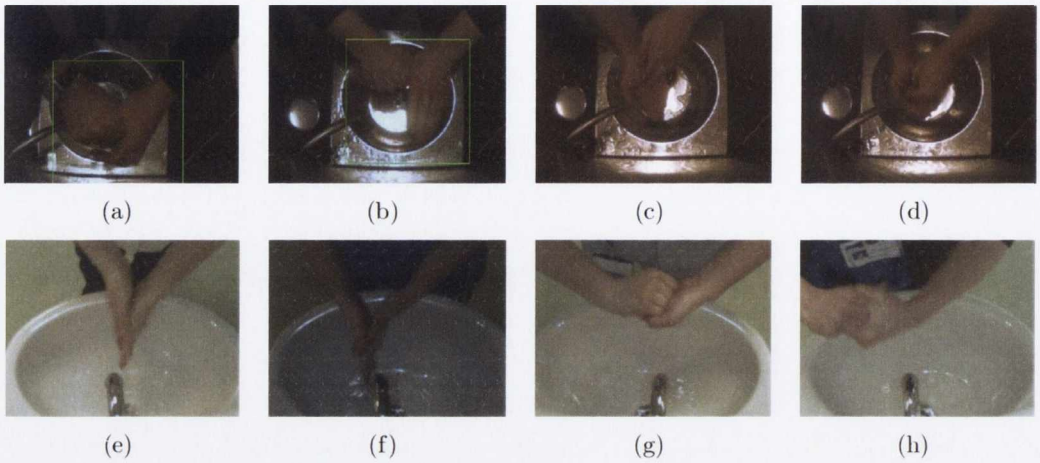


Figure 1.4: *Challenges in hand washing gesture recognition*

Machine (SVM) [24]. The method is similar to methods used in sign language gesture recognition [164][21][152]. However, in conventional sign language gesture recognition, a sign gesture is required to put in front of a camera steadily for a few of seconds to facilitate the recognition. On the contrary, hand washing gestures are continuous bi-manual hand movements involving a large number of gesture variance, transitions and self-occlusions. This makes the hand washing gesture recognition much harder than the sign gesture recognition.

The second and third evaluated methods take the dynamic characteristic of the hand washing gestures into account but analyse it from different points of view. The second method employs the HOGs from each frame as a primitive in a sequence and models the dynamic nature of gestures with sequence labelling tools [73][11]. The third method encapsulates the dynamic information into space-time "3D" HOGs descriptors [7] during feature extraction. These descriptors are clustered as visual words and "bag of words" tool is used [65][52] for recognition.

Both the second and third methods are adapted from methods for human action recognition. Human action recognition in Computer Vision has been studied for many years, and some methods are adaptable for human gesture recognition. However, we should also be aware of the difference between these two domains and be careful in our system design. Firstly, human body parts such as torso and limbs are relatively big while hand parts such as palm and fingers are relatively small, which means that details in images may be more important for gesture recognition than action recognition.

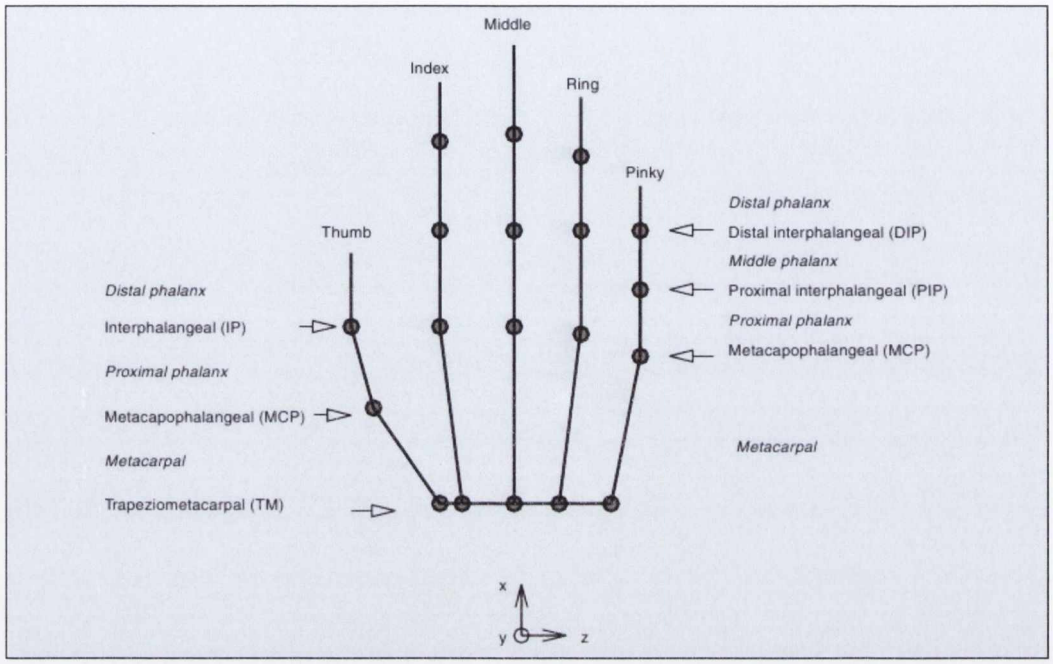


Figure 1.5: *Hand skeleton structure [172]*

Secondly, human hands are more flexible than human bodies. Figure 1.5 shows the skeleton of a hand [172]. Considering global hand pose, human hand motion has roughly 27 Degrees of Freedom (DOF) which means hand motion is highly articulate. Thirdly, the movements in hand washing gestures are small and much faster than human actions, which makes the recognition even more difficult.

Our fourth method Texton analysis with Graph-Cut Hidden Conditional Random Fields (TGC-HCRFs) further explores the computer vision techniques for hand washing gesture recognition. We unify texton analysis with Hidden Conditional Random Fields (HCRFs) in the same framework for human gesture recognition. The primal idea is that gesture textons can be modelled by the hidden states in HCRFs such that the method would be beneficial from both HCRFs and texton analysis. General structure Markov Random Fields (MRFs) and CRFs have long history in image segmentation, texture analysis and object recognition [93][78][118]. Tree-structure HCRFs have also been used for human action recognition [139][3], but the general structure HCRFs for action recognition or gesture recognition has not been studied before. This is probably because the expensive probability inference of general structure HCRFs prevents experiments with large data sets. Our TGC-HCRFs method follows the energy-based learning

framework [111] instead of conventional probability based training [73][11]. It takes advantage of the newly developed Quadratic Pseudo-Boolean Optimization (QPBO) inference [95] and max-margin training approach [158][83] such that the training time for general structure HCRFs is greatly reduced even if a large training data set is employed.

Comparing to our first method, TGC-HCRFs can capture the dynamic gesture information with gesture textons. In the evaluation, space-time "3D" HOG features from small video patches are extracted as primitives for building gesture textons. Comparing to the second method, TGC-HCRFs method offers a more flexible way to include the contextual and neighbouring information for recognition, and would be less influenced by the skin and ROI detection results. Comparing to our third method which loses all gesture spatial configuration information, TGC-HCRFs can reserve the spatial information with the grid graph.

1.1.3 Data set

A hand washing data set is prepared for evaluating aforementioned four methods. The data set has three subsets: ceramic sink set, stainless steel sink set and white board set. The ceramic sink set records nurses and doctors' hand washing activity in hospitals in real scenarios. It is assumed that the nurses and doctors are acquainted with hand washing techniques and thus there was no any supervision from hand washing technique experts during the video collection. The videos show large variance in the gestures including changes in lighting, background and hand appearance. In stainless steel sink set, participants are asked to finish all 10 hand washing gestures with a little supervision from hand washing technique experts. All participants are new to hand washing technique and therefore the hand washing gestures can be performed very differently. The white board set videos are collected in laboratory environment. Lighting condition is good for hand localization. Participants are all familiar with the hand washing technique and can finish the hand washing routine with gestures that are very close to our 10 hand washing gesture definitions.

All videos in the data set are size of 320×240 in Motion JPEG format. The ceramic sink set videos are recorded in around 12 frames per second (fps); stainless steel sink set videos are recorded at a rate of 25 frames per second and white board set videos are 30 frames per second. Different frame rates can be regarded as washing hands at different speed. 12 fps means a relatively faster hand washing speed than

Gesture	1	2	3	4	5	6	7	8	9	10
Group 1	301	277	346	424	283	289	275	356	299	259
Group 2	413	296	220	365	312	279	433	378	411	473
Group 3	280	339	316	358	348	416	328	276	317	416
Group 4	252	271	269	372	316	292	329	361	364	395
Group 5	374	549	346	511	280	246	390	326	353	391
Total	1620	1732	1497	2030	1539	1522	1755	1697	1744	1934

Table 1.1: *Hand washing data set summary. All numbers are measured in frames. All videos are separated into 5 groups for cross validation.*

the 25 fps and 30 fps. Table 1.1 summarizes the whole data set in terms of the frame numbers.

1.1.4 Evaluation criteria

The hand washing data set is separated into five groups in the evaluation. Methods are evaluated in a style of 5-fold cross validation: one group of videos are picked up as the testing set once while the rest of videos are used for training. The experiments are repeated until all groups are tested. The test results from five groups are then averaged as the final report result. In order to compare the results from different methods, three multi-class classification criteria, classification precision, macro-averaged F-measure, and micro-averaged F-measure, are computed. The classification precision is reported as averaged group accuracy with standard deviation. It is a measurement of repeatability of the method under evaluation. For some results, confusion matrices are also given as complements to the classification precision reports. Macro-averaged F-measure and micro-averaged F-measure are adaptations of F-measure for multi-class classification [10]. F-measure is a common measure of test performance in information retrieval. It considers both the precision and recall of a test. The measure scores are in the interval $(0, 1)$, in which F-measure reaches the best score at 1 and the worst score at 0. For multi-class classification problem, macro-averaged F-measure calculates the F-measure score as an unweighted average of the precision and recall over all classes, regardless of class frequencies. Thus, the measure could be influenced more by the classifier's performance on rare categories. On the contrary, micro-averaged F-measure is an average score over instances. Classes which have many instances are given more importance in the measurement. In our evaluation, the macro-averaged F-measure and

micro-averaged F-measure are computed by formulas 1.1,

$$\text{Macro-average F-measure:} \quad 2 \cdot \frac{P_{macro} \cdot R_{macro}}{P_{macro} + R_{macro}}$$

$$\text{Micro-average F-measure:} \quad 2 \cdot \frac{P_{micro} \cdot R_{micro}}{P_{micro} + R_{micro}}$$

$$P_{macro} = \frac{1}{M} \sum_{i=1}^{M} \frac{TP_i}{TP_i + FP_i} \quad , \quad R_{macro} = \frac{1}{M} \sum_{i=1}^{M} \frac{TP_i}{TP_i + FN_i}$$

$$P_{micro} = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M} (TP_i + FP_i)} \quad , \quad R_{micro} = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M} (TP_i + FN_i)} \tag{1.1}$$

where M is the number of classes, P stands for the precision and R stands for the recall. TP_i is the true positive number which is the count of correctly classified frames in gesture i, and the false positive FP_i is the count of frames that are misclassified to be gesture i. The true negative TN_i indicates the number of correctly classified frames of all other gestures except gesture i, and false negative FN_i indicates the number of frames that belong to gesture i but misclassified to be other gestures.

In some evaluations statistics student's t-test is performed. We randomly pick 20% videos without replacement from the whole data set as the testing set and use the rest for training. This procedure is repeated 15 times, and all testing results are pooled together to form the t-test samples for hypothesis testing.

1.2 Contributions

In this thesis, hand washing gesture recognition is studied as a new type of gesture recognition task. Hand washing gestures are articulate, fast, bi-manual hand movements which introduce a lot of intra-class variety and inter-class ambiguity. Aiming to tackle these challenges, several contributions have been made in the research.

1.2.1 Major contributions

Three major contributions are dissertated in the thesis:

- A new type of gesture recognition, hand washing gesture recognition which has hardly been tackled before, is studied in this thesis. 10 hand washing gesture pattens are defined, and various properties of hand washing gestures are analysed.
- Four different computer vision methods are developed and evaluated for hand washing gesture recognition, which research into the hand washing gestures from distinct perspectives.
- A new method TGC-HCRFs is proposed, which unifies the texton analysis and HCRFs within the same framework. The evaluation of the TGC-HCRFs method also provides a practical study of HCRFs with grid graph for gesture recognition, which has not been seen in literature.

1.2.2 Minor contributions

Along with the major contributions, some minor contributions are also made:

- The sequence labelling tool HCRFs is applied for gesture recognition. The key parameters such as the context window size, number of hidden variables, etc. are evaluated.
- Different test sequence length is evaluated in the linear-chain HCRFs. It is found that the classification performance grows along with the increase of the test sequence length.
- The depth of trees in Extremely Randomized Clustering (ERC)-Forests [52][147] is evaluated for controlling the discriminative power of visual vocabulary.
- A bootstraping [167][120] strategy is applied in the max-margin training of TGC-HCRFs, which alleviates the local optimum problem of TGC-HCRFs and reduces the overall training time.
- Several findings of applying HCRFs with grid graph for gesture recognition have been drawn. It has been found that gesture recognition favours fine-grid sampling and rich description of video patches. The vertical neighbours of video patches also carry more information than the horizontal neighbours in recognition.
- The TGC-HCRFs method has the ability of automatically determining the visual vocabulary size for gesture recognition.

1.3 Thesis Structure

In the first chapter, we have given a brief introduction of hand washing gestures and an overview of this thesis. In the following chapters, we will elaborate the details of our research on the vision-based hand washing gesture recognition:

- **Chapter 2** A literature review of gesture recognition, object recognition and human action recognition is given. The theory background for developing our TGC-HCRFs method is also provided.

- **Chapter 3** The hand washing gestures are recognized using static HOG features extracted within every frame. A number of parameter settings in HOG extraction are evaluated with linear SVM. The image processing for ROI detection, which is also used in chapter 3 and 5, is described in this chapter as well.

- **Chapter 4** The linear-chain HCRFs is introduced for hand washing gesture recognition. A hand washing gesture is modelled as a sequence with a number of hidden states. Multiple parameters in HCRFs are evaluated including the context window size, hidden variable number, and sequence temporal resolution. Different time-shift window size for a frame-by-frame prediction is assessed.

- **Chapter 5** Hand wash gestures are described by a bag of visual words. Each visual word represents a cluster of space-time interest points which are detected as spatial-temporal energy hot-spots. The Extremely Randomized Decision Trees [147] is used to generate the visual vocabulary, and the SVM with a χ^2 kernel is applied for classification. No ROI detection is applied in this method, and no spatial information of hand washing gestures is conserved.

- **Chapter 6** Our TGC-HCRFs model is elaborated including the training algorithm for a large scale dataset. A number of parameters in TGC-HCRFs are evaluated and discussed for hand washing gesture recognition.

- **Chapter 7** Our four hand washing gesture recognition methods are reviewed and compared. The conclusion of our whole study is then drawn and possible future work is given at the end of the thesis.

Chapter 2

Background

Gesture recognition has gained considerable interest in computer vision community in recent decades. The problem of human gesture recognition can be decoupled into two levels: the low level hand posture detection and the high level hand gesture recognition [129][142][172][21]. A static hand posture, usually called a "**posture**", is a certain hand pose or configuration without any representation of movements, whereas dynamic hand gesture or "**gesture**" is defined as dynamic hand movement referring to a sequence of postures connected by continuous motions over a short time span. In fact, the gesture recognition has close relationship to the object recognition and human action recognition. Many methods from both object recognition and human body action recognition can be applied for recognizing the postures and gestures.

In this chapter, a literature review of feature extraction and classification used in gesture recognition, object recognition and human action recognition is presented first in section 2.1. In section 2.2 and 2.3, some background of Hidden Conditional Random Fields and texton analysis applied in this thesis is provided.

2.1 Feature Extraction and Classification

Many methods have been proposed for human gesture recognition in recent years. Moreover, a lot of methods in object recognition, face recognition and human action recognition can also be adapted for recognizing human gestures. Following Garg's taxonomy [129], all these methods can be grouped into two categories: 3D model based approaches and appearance based approaches. Our following review will mainly focus on the appearance based approaches to which the thesis belongs.

2.1.1 3D model based approaches

One approach to recognizing hand gestures is to build a model of the 3D kinematic hand structure. Kuch and Huang [98] represent the hands with 3D cubic B-splines and estimate the hand configurations via "analysis-by-synthesis". The method does not restrict the gestures for the hand tracker and there is no need for the user to wear a special glove or other physical items. However, the method requires interactive selection during model calibration to locate all the hand joints and remove background. The automatic model fitting in the calibration phase shares a common problem with many 3D model-based approaches, which is searching for the optimal hand posture in a huge hand configuration space. Such a search process is computationally expensive and the optimization is prone to local maxima. Shimada [146] applied inequality constraints to reduce the search space. These constraints include: (a) shape parameters are constant over the sequence; (b) pose parameters change continuously; (c) each parameter is within a certain range and has relations with the other parameters. Lin [75] proposed a learning approach without representing the constraints explicitly. The redundancy of the hand configuration space is eliminated by finding a lower-dimensional subspace which is called C-Space. The approach focuses on the analysis of local finger motions and constraints, and eliminating unnatural hand configurations. A drawback of this method is that the motion data used to form the constraints implicitly are collected by special gloves.

Although 3D model based approaches potentially allows a wide class of hand gestures, the optimization of models could be very computationally complex due to the high DOF of the hand geometry. A large training database is also required to cover all the hand shapes under different views. Moreover, properly modelling the optimization constraints either implicitly or explicitly is not a easy task.

2.1.2 Appearance based approaches

Alternatives to 3D model based approaches are the appearance based approaches. Appearance based approaches aim to extract abstract information from the static or dynamic visual appearance for recognition. These approaches have the advantage of real time performance due to the fact that appearance based features are easier to compute [129]. Depending on how the features are extracted and processed, we are going to review the appearance based approaches from two perspectives [122]: *dense represen-*

tation approaches and *sparse representation approaches.*

Dense representation approaches

Dense representation approaches generally extract features densely over an entire image, video, or a detection window, and collect them into a high-dimensional vector that can be used for subsequent classification. Oren and Papageorgiou [117][27] extract an over-complete set of Haar wavelet coefficients at different orientations and scales in a detection window. The wavelets encode visually significant patterns and provide a reasonable degree of translation invariance for recognition. The relationship between the wavelets is then learned by SVM. The methods are evaluated on face and pedestrian detection problems. The results show that they can be used to robustly define rich and complex classes of objects, and they are invariant to changes in colour and texture. Viola and Jones [163][133] also extract the Haar-like features and use the integral image to speed up feature extraction. These features are then fed into Adaboost [171] as attentional cascade to achieve real-time performance.

Another well-known dense representation features are HOG [121] [122] [123]. The HOG features are reminiscent of edge orientation histograms [165] but computed on a dense grid of uniformly spaced cells. It is believed that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions, even without precise knowledge for the corresponding gradient or edge positions. Evaluation of the technique by detecting various classes of objects demonstrates that the classifiers trained with HOG features are invariant to certain degrees of translation, rotation and deformation, and is robust to colour, texture, illumination changes and cluttered backgrounds. However, object detection or recognition may require the HOG calculation in multiple regions, and the computation is expensive. Inspired by the integral images [163], integral histograms [53][47][136] are proposed to alleviate this problem. Chandrasekhar [160] also designed compressed histograms of gradients (CHOG) as a type of low bit rate feature descriptor to speed up the matching process.

Extending the dense representation to spatial-temporal domain, human actions can be recognized analogue to language analysis [33][34][61][25] by describing human movements as simpler movement primitives [60] with HOG descriptors. Mauthner [155] computes the HOG features on both appearance and motion fields, and show that a combination of shape and motion information can improve the human action detec-

tion and classification performance significantly. Schindler [90] also uses dense shape and motion features for action recognition, and obtains the same conclusion. Instead computing HOG features on images, Junejo [71] calculates HOG on self-similarities matrices computed from video segments, and then the extracted HOG features become view-independent. Davis and Bobick [5] compute 7 Hu moments [74] from Motion-Energy Image (MEI) and Motion-History Image (MHI) to build a temporal template. Bradski [58] further generalized the MHI as timed Motion History Image (tMHI) which directly encodes the actual time in a floating-point format. However, these methods are view-dependent and require well segmented foreground figures which can be easily effected by the illumination changes and cluttered background. Efros [4] uses smoothed and aggregated optical flow computed at low resolutions for action recognition. The method is robust to jitters introduced by the tracking [3] but requires that the actions are observed from a distance. This is because the dense flow estimation becomes unreliable when the observed acting person is large.

Weinland [43] extends Davis' method as 3D Motion History Volumes (MHV). The volumes are built from multi-view silhouettes from multiple cameras and form a free-viewpoint representation of the human actions. Fourier based features are used with cylindrical coordinates to express motion patterns. Achard [20] also builds the 3D volumes but from a single camera binary silhouette sequence instead. The volumes are termed as "space-time micro-volumes" and are used to extract the semi-global moment features. Gorelick [101] generalizes the Poisson equation used in object recognition [102] for action recognition. Global moment features are extracted via applying different types of Poisson equations for the characteristic function in moments computation. The method is fast, robust to partial occlusions, non-rigid deformations and significant changes of both scale and viewpoints but requires preprocessing to extract the foreground figures in order to form the space-time silhouette volumes.

Instead of building training models, dense representation approaches can also be correlation based. The test samples correlate with exemplars using image intensities or dense features for classification. Barrow [16] uses "chamfer matching" for shape comparison but can only tolerate slight misalignment or distortion of two collections of shape fragments. Shechtman [48] measures the texture correlation between test images and example images. The method accounts for local and global geometric distortions, and gives matching capabilities of complex visual data in real cluttered images. Sullivan [80] uses the key frames as exemplars for matching. The recognition is based on

quantitative similarity that computes point to point correspondence between shapes, however, manual marking of some interior points on human bodies for correspondence is required during the training. Shechtman and Irani [49] construct 3×3 Gram matrix taking pixel space-time gradients of 3D patches from both test and example volumes. Motion consistency is measured as the rank of the Gram matrix. No foreground subtraction and no prior learning of activities are required. However, the method is mainly designed for action detection. It may have difficulty in discriminating similar actions such as walking and running. Zelnik-Manor [107] builds a temporal pyramid of videos by taking different temporal resolution. Built on the temporal pyramid, empirical distributions for every action event is constructed using normalized space-time gradients. Actions are detected by comparing the empirical distributions between the sample clip and a test video segment. The dense features extracted from moving trajectories can also be used for test and exemplar video matching [29][137][144]. However, these methods may not be able to capture the large variety of actions.

Although dense representation based approaches have achieved very promising recognition performance, most methods require ROIs which are bounding boxes of the targets. This usually involves tracking and segmentation. Many methods can be used for the detection of ROI such as mixture of Gaussian [28][149][44], kernel density estimation [50][9], skin colour detection [150][161][103] and flocks of features [100][68]. However, these methods usually have one limitation or another, and are not universally applicable for most situations. For example, mixture of Gaussian needs to concern the adaptation rate, especially in uncontrolled clutter background; skin detection is sensitive to the illumination changes and skin-colour distractors. More precisely, the detected box can be inaccurate because of occlusion and may keep changing the size due to the non-rigid human movements. This will bring in difficulties when dealing with the intra-class variety and inter-class ambiguity. On the other hand, the cost of the exhaustive scan in images or videos with a fixed-size box is very computationally expensive.

Sparse representation based approaches

Sparse representation based approaches are based on local descriptors of relevant local image regions or video patches. It was motivated by the physiological studies that the visual information human received is redundant [51] and that human gaze preferentially fixates on image regions with corners and multiple superimposed orientations [122]. In

object recognition, the approaches first search for salient regions in images through interest point detectors. These detectors include Harris detector [67], Harris-Laplace detector [88], Hessian-Laplace detector [9], Difference of Gaussians (DOG) detector [42], Harris-Affine detector and Hessian-Affine detector [89]. The detected regions can be corners, blobs, ridges or entropy. The hypothesis is that these detected regions are informative to represent the image contents.

Various descriptors are designed to describe the detected regions. Local Jet [30] as point descriptor characterizes the local geometry in the neighbourhood of the detected point. Shape context descriptor [140] reflects the positional uncertainty of useful coarse shape cues. It expresses the configuration of the entire shape relative to a reference point with a set of vectors originating to all other sample points on the shape. Scale Invariant Feature Transform (SIFT) [42] delineates the points as voted orientation histograms rectified by local scale and dominant orientation based on the scale-space theory [113]. Considering its invariant to scale, orientation and affine distortion, and partially invariant to illumination changes, Wang [36] employs SIFT in hand posture recognition. PCA-SIFT [91] provides a more compact and more distinctive representation of SIFT by applying Principal Component Analysis (PCA) to the local gradient patches. Speeded Up Robust Features (SURF) [63] improves the SIFT using Haar-like features in the descriptor and obtains faster processing time than SIFT. Bao [84] employs the SURF for dynamic hand gesture tracking and recognition.

Extending to the space-time domain, the spatio-temporal interest points can be detected and described similarly to the 2D interest points. Scovanner [132] extends the 2D SIFT to the 3D SIFT for action recognition. Laptev [108][31] generalizes the Harris detector with Local Jet description for the spatio-temporal interest points. However, the method is criticized for rare detection in aperiodic movements like rodent behaviour or facial expressions [46][72]. Attempting to overcome that problem, Dollar [46] detects the interest points via the extrema of linear filter response and describes them by Cuboids which are composed of gradients and flow vectors. The greater amount of information embedded in an image sequence over a 2D image also enables the interest points to be constructed differently. Klaser [7] describes spatio-temporal interest points as 3D HOGs which is suitable for describing sport actions in Wang's evaluation [65]. Yilmaz [13] constructs action volumes with silhouettes. Interest points are then detected as curvature extrema and described with various surface types such as peak, pit, valley and saddle valley. The detected points are examined for their view-invariance

to human actions.

Using sparse features for recognition is usually based on the Bags of Features (BOFs) method [57][148]. All detected interest points in the training set are pooled together and clustered as "visual words". A codebook or visual vocabulary is built thereof and each interest point can be assign to a word by some similarity measurement. The word occurrence histogram of an image or an image sequence is then constructed for subsequent statistics analysis and the results produced are usually "semantics-oriented".

Comparing to the dense representation based methods, sparse representation based methods generally do not require any foreground subtraction and thus can handle complex scenes with cluttered backgrounds. However, this merit can also turn to be a drawback as many detected points may be unreliable and uninformative. In order to obtain reliable and informative 2D and 3D interest points, a possible way is applying feature pruning in the regions of interest [76]. Moreover, interest points based methods generally do not preserve the spatial configuration of the targets. To overcome this limitation, Boiman [126] localizes irregular action behaviour with local video patch ensembles at the price of heavy computation. Niebles [125] combines the hierarchical model with the BOFs method to capture the spatial relations of human parts. However, the human part layer of the hierarchical model is built on the relative positions of groups of interest points, and may have difficulty in assigning human parts on cluttered background.

2.1.3 Classification methods

Most approaches, including dense representation based approaches and sparse representation based approaches, require to train classifiers for recognition. In this section, we review several popular classifiers for pattern recognition in literature.

Eigen analysis classifier

Eigen analysis provides a simple but efficient way to summarize the data. It seeks an orthogonal basis that spans a low-ordered subspace that accounts for most of the variance in a set of data [129]. Coogan [152] represents the hand shapes in a subspace created by PCA to handle the small rotations and translations. Belhumeur [130] builds "fisherfaces" to account for variation in lighting and facial expression in face recognition

by explicitly modelling the difference between classes with Linear Discriminant Analysis (LDA) [168]. Li [116] proposes 2D LDA which performs eigen analysis directly on 2D image matrix. Park [64] considers the substructure of each class as hierarchical LDA which shows better performance than original LDA. Original LDA has an assumption of equal within-class covariance for all classes, Chen [138] proposes heteroscedastic[1] LDA to relax such an assumption.

However, eigen analysis usually requires a well prepared data set. In face recognition, the training images are all well cropped and scaled, and only contain people's faces. Meanwhile, eigen analysis methods are sensitive to the outliers presented in the training set [106].

Support Vector Machine classifier

SVM [24] has been widely used for object recognition and human action recognition due to its good generalization capability. It finds separating hyperplanes that maximises the margin between classes in either the input feature space or a transformed high dimensional space. Dalal [122] applies the SVM classifier for object recognition with HOG features. Felzenszwalb [128] improves the approach by building a multi-scale, deformable part model with Latent SVM [26] such that different parts of an object can be loosely positioned. Recently using SVM kernel methods with sparse features has become popular in object recognition and action recognition [35]. The SVM kernel is substituted by a similarity measurement which obeys the Mercer's condition [24] and accepts unequal numbers of interest points detected between images or videos. Wang [65] applies the χ^2 kernel in the evaluation of local spatio-temporal features and shows promising results for action recognition. Grauman [86] designs pyramid match kernel such that partial match correspondences can be measured between two interest point sets.

Adaboost classifier

Adaboost is an algorithm for constructing a "strong" classifier by combination of weighted "weak" classifiers that are called repeatedly in a series of rounds during training. Many features can be combined with Adaboost to build efficient classifiers. Viola [163][133] uses Adaboost to train cascades of weak classifiers for face and pedestrian

[1] simply put, "heteroscedastic" means all classes have different covariance matrices

detection. A positive result from the current classifier in the chain triggers the evaluation of the next classifier aiming to reject as many of the remaining negative cases as possible while still retaining all the positives. In each training stage, Adaboost selects a small number of critical visual features from a larger set. Levi [87] uses local edge orientation histograms in the Adaboost in order to train the classifier with a small number of examples. Laptev [70] also applies local edge orientation histograms to train the weak learner, Weighted Fischer Linear Discriminant which projects multi-dimensional features to 1-dimensional manifolds, to overcome the problems of limited training sets. Sabzmeydani [131] run the Adaboost in local regions with low-level gradient features to construct a new mid-level shapelet feature which is the weighted sum of weak classifiers. The experiment run on pedestrian detection shows that the boosted feature can capture more information than fixed features sets. Torralba [12] focused on detecting a large number of different classes of objects in cluttered scenes and learned the shared features with a modified Adaboost algorithm in which the weak learners are expressed as regression "stumps" [56]. Zheng [166] proposes Realboost with a novel image strip features that are calculated via the mean intensities of the single strip regions. However, the method is only suitable for the object detection like cars which have edge-like and ridge-like strip patterns.

Graphical model classifier

A graphical model is a probabilistic model using a graph-based representation for a probability distribution. A well-known graphical model is the Hidden Markov Model (HMM) [105] which has been widely used for sequence segmentation and labelling. Mendoza [115] employs a continuous HMM with contour histograms to model the dynamic structure of human actions. An action class is assigned to a video sequence by the trained HMM which maximizes the likelihood of the observation sequence. Ahmad [114] also applies HMM to model different human actions. The method takes an action from multiple views into account so that a set of HMM models are trained for each human action. However, HMM is a generative model which assigns a joint probability to paired observation and label sequences [73]. Two conditional probability distributions are required in the joint probability computation: a state transition probability from previous state s' to current state s, $P(s|s')$ $s, s' \in S$, and an observation probability, $P(o|s)$ $o \in O, s \in S$, where S is a finite set of states and O is a set of possible observations [8]. Usually the observations are multinomial distributions and in many cases

tasks would benefit from a rich representation of observations [8], but in order to have a tractable inference, the computation of $P(o|s)$ in HMM needs to enumerate all possible observations. Therefore, strong independence assumptions among the observation variables are made in HMM [92]. Nevertheless, it is not necessary to expend modelling effort on the observations that are fixed anyway at test time. Therefore, McCallum [8] proposes a discriminative model Maximum Entropy Markov Models (MEMMs) which compute a conditional probability $P(s|s', o)$ instead of the HMM transition and observation probabilities. The MEMMs model relaxes the independence assumption in observations made in HMM and allows arbitrary features from observations which may overlap or interact with each other. However the MEMMs suffer a label bias problem [73] in practice. The local probabilities with fewer transitions have advantages over those with many transitions in MEMMs due to the per-state normalisation of transition scores. In other words, when a state has only one single outgoing transition, the learning of MEMMs will effectively ignore the observations. To overcome this problem, Lafferty [73] proposes the CRFs which are also discriminative models. There is no observation independence assumed in CRFs and the model accounts for state sequences globally rather than locally to overcome the label bias problem. With these advantages, Sminchisescu [32] employs CRFs for human action recognition. Silhouettes extracted from frames surround current frame are used as contextual information in the classification. Shimosaka [2] considers the situation where multiple symbols or labels are present in a single frame and comes up with Multi-Task CRFs solution. The method incorporates the interaction between action labels as well as the Markov property of actions to improve the accuracy of all label assignments at a specific time.

However, CRFs lack of the ability of representing the internal structure of actions. Wang [139] therefore employs HCRFs for gesture recognition. The HCRFs model learns distributions of hidden states for different gestures in a discriminative manner and outputs a single class label to a sequence in prediction. Liu [54] applies Neighborhood Preserving Embedding (NPE) for observation dimension reduction before employing HCRFs and the local neighbourhood structure of data is preserved by NPE. In order to capture both extrinsic dynamics and internal structure of actions, Morency [104] proposes the Latent Dynamic Conditional Random Fields (LDCRFs) for continuous gesture recognition. By preparing an exclusive hidden variable set for each class label, the model is able to give prediction to videos frame by frame. Build on LDCRFs, Ning [66] develops Latent Pose CRFs (LPCRFs) which substitutes explicitly calculated ob-

servations with latent pose estimators such that feature extraction is jointly optimized with the random fields.

2.2 Hidden Conditional Random Fields

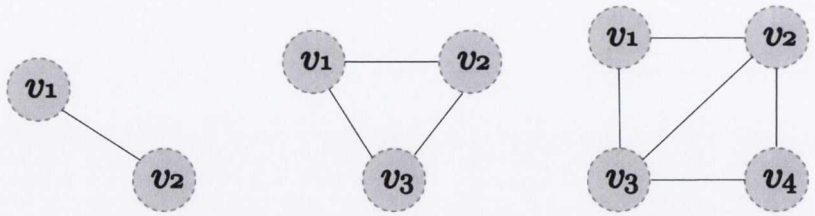
In the rest of this chapter, some theory background that our evaluated methods build on in the thesis will be given. In this section, we first dissertate the HCRFs model which is used in chapter 4 and chapter 6.

2.2.1 Undirected graphical model

HCRFs are undirected graphical models. An undirected graphical model is a family of probability distributions that factorize according to an undirected graph [151]. An undirected graph has a set of nodes each of which corresponds to a variable or group of variables, as well as a set of undirected links each of which connects a pair of nodes [17]. An very important property of undirected graphical models is the conditional independence which can be used to factorize a complex probability distribution into a product of functions defined over sets of variables that are local to a graph. The factorization of the probability distribution should be performed in a way that conditionally independent nodes do not appear within the same factor, that means the nodes belong to different cliques [92].

In an undirected graphical model, a *clique* is a subset of the nodes in a graph, which is fully connected. Figure 2.1 shows some examples of graph clique. Example (a) in figure 2.1 is a clique of two nodes and example (b) is a clique of three nodes, but example (c) is not a clique as there is a missing link from node v_1 to node v_4. If a clique can not be extended to include any other nodes from the graph without ceasing full connection between all pairs of nodes, this clique is a *maximal clique* as example (b) in figure 2.1. An arbitrary function can be defined over a maximal clique such that the complex distribution defined over a graph could be represented by a product of local functions.

Let us denote a clique by $\mathcal{C}$ and the vertex set of a graph by $\mathcal{V}$. According to the Hammersley-Clifford theorem [112], the probability distribution over an undirected



(a) (b) (c)

Figure 2.1: *Graph clique examples:* (a) *pair-site clique;* (b) *triple-site clique;* (c) *not a clique*

graph $P(\mathcal{V})$ can be approximated as form 2.1,

$$P(\mathcal{V}) \quad \propto \quad \prod_{\mathcal{C}} \Psi_{\mathcal{C}}(\mathcal{V}_{\mathcal{C}}) \tag{2.1}$$

where $\mathcal{V}_{\mathcal{C}}$ is the set of variables in a maximal clique $\mathcal{C}$ and $\Psi_{\mathcal{C}}(\mathcal{V}_{\mathcal{C}})$ is a *potential function* defined on the maximal clique. In general, it is required that the potential functions are strictly positive such that an undirected graph can be factorized as formula 2.1 and it is certain that $P(\mathcal{V}) > 0$. As the potential functions are all positive, it is also convenient to express these functions as exponentials as formula 2.2,

$$\Psi_{\mathcal{C}}(\mathcal{V}_{\mathcal{C}}) = \exp(-E(\mathcal{V}_{\mathcal{C}})) \tag{2.2}$$

where $E(\mathcal{V}_{\mathcal{C}})$ is called an *energy function.* The choice of the positive potential functions are arbitrary. It is not necessary to have specific probabilistic interpretations for the functions. However, one consequence of the generality of the potential functions is that their product can not guarantee satisfying the axioms of probability. Therefore a normalization constant $\mathcal{Z}$ calculated by formula 2.3, which is also called *partition function*, is introduced to ensure a proper probability output.

$$\mathcal{Z} = \sum_{\mathcal{V}} \prod_{\mathcal{C}} \Psi(\mathcal{V}_{\mathcal{C}}) \tag{2.3}$$

Computing the partition function requires to sum over all possible assignments to the variables in the $\mathcal{V}$. This computation can be very expensive, even intractable in many cases. For example, given a model with M discrete nodes each having K states, the evaluation of the normalization term involves summing over K^M states in the worst case, which is exponential in the size of the model [17].

2.2.2 Conditional random fields model

Given an observed sequence X with length T and the corresponding class label sequence $Y = \{Y_1, Y_2, \cdots, Y_t, \cdots, Y_T\}$, a linear-chain undirected graph structure can be depicted as figure 2.2. Modelling a joint probability $P(X, Y)$ over this simple linear-chain structure could be intractable [162] as it is impossible to sum over all possible assignments to the variable X with Y in most cases. To overcome this problem, the CRFs model is proposed [73]. The CRFs model computes the conditional probability $P(Y|X)$ instead of the joint probability $P(X, Y)$, and leaves the observation distribution $P(X)$ unspecified. In fact, $P(X)$ is not needed for classification anyway and we are only interested in the output structure Y conditioned on the input X. To model the conditional distribution $P(Y|X)$, CRFs only represent Y as an undirected graph in which each vertex of the graph corresponds to a variable Y_t. The joint variable Y, when conditioned on X, admits the Markov property in that the conditional distribution of Y_t given its neighbours, defined by the graph, does not depend on other variables outside the neighbourhood [157]. The formal definition of CRFs is defined as below:

> *"Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph such that $Y = (Y_t)_{t \in \mathcal{V}}$ is indexed by the vertices of $\mathcal{G}$. Then (X, Y) is a* ***conditional random field*** *in case, when conditioned on X, the random variables Y_t obey the Markov property with respect to the graph: $P(Y_t|X, Y_{t'}, t' \neq t) = P(Y_t|X, Y_{t'}, t' \in \mathcal{N}(t))$, where $\mathcal{N}(t)$ is the neighbourhood of Y_t in $\mathcal{G}$"* [73].

With the above CRFs definition, the conditional probability $P(Y|X)$ defined over the linear-chain graph would be computed by formulae 2.4 and 2.5.

$$P(Y|X) = \frac{1}{\mathcal{Z}} \prod_{\mathcal{C}} \Psi_{\mathcal{C}}(Y_{\mathcal{C}}, X) \tag{2.4}$$

$$\mathcal{Z} = \sum_{Y'} \prod_{\mathcal{C}} \Psi_{\mathcal{C}}(Y'_{\mathcal{C}}, X) \tag{2.5}$$

where $Y_{\mathcal{C}}$ denotes label variables Y involved in clique $\mathcal{C}$, and Y' denotes all possible label sequences. Note that the normalization constant $\mathcal{Z}$ is computed by only summing over Y', and can be efficiently calculated via dynamic programming [105] for a linear-chain graph structure.

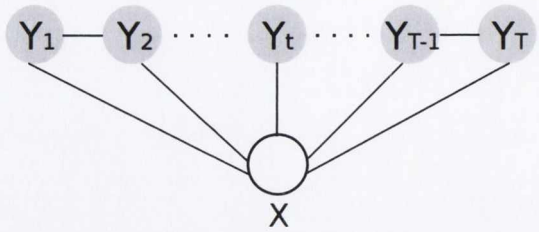


Figure 2.2: *Linear chain CRFs*

Feature functions

The clique potential Ψ_C specifies how local variables interact and how much the interaction contributes to the global distribution [33]. Since a clique only contains two nodes (Y_{t-1}, Y_t) for a first-order linear-chain graph, the clique potential can be written as form 2.6,

$$\Psi_C(Y_C, X) = \exp\left(\sum_{i=1}^{m} \lambda_i f_i(Y_{t-1}, Y_t, X)\right) \tag{2.6}$$

where f_i is called *feature function* which encodes prior belief about dependency between the conditioning variable X and the local variables (Y_{t-1}, Y_t) , and λ_i is corresponding feature weight. Every clique potential can have its own weight vector to specify how the local features contribute to the global distribution. In other words, the feature weights are position dependent. However, it is more common to use the same weight vector across all clique potentials, which means the weights are position independent. This is also known as *parameter tying.*

2.2.3 Hidden conditional random fields model

The CRFs model has shown to be a powerful discriminative model for sequence labelling [2][162]. It outputs a class label for every node in a sequence. However, the model lacks of the ability to represent the internal structure of a sequence. There are many cases that the categorization is based on the whole sequence and only one class label is output for the entire sequence. For example, a sequence of hand movements over a span of time may present only one class of gesture. To meet the challenge of intrinsic structure representation, the HCRFs model is proposed [11]. HCRFs model is an extension of CRFs model with hidden variables. It can describe the internal structure of a sequence

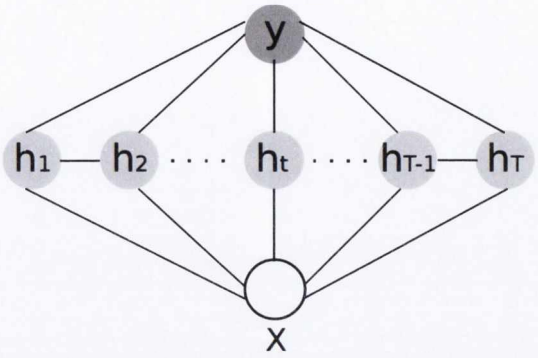


Figure 2.3: *Linear chain HCRFs*

by many intermediate unobserved states. These hidden states will be inferred from the observation X and jointly represent the sequence class y [2] .

With the HCRFs model, the linear-chain structure of a sequence would be depicted as figure 2.3, where each node of the sequence is assigned with a hidden state h_t from a finite set of hidden variables $\mathcal{H}$. The sequence of hidden states $h = \{h_1, h_2, \cdots, h_t, \cdots, h_T\}$ is modelled as an undirected graph and admits the Markov property. Thus, similarly to CRFs, for a linear-chain HCRFs model, we can compute its conditional probability $P(y, h|X)$ by formulae 2.7 and 2.8

$$P(y, h|X) = \frac{1}{\mathcal{Z}} \prod_{C} \Psi_C(y, h_C, X) \tag{2.7}$$

$$\mathcal{Z} = \sum_{y'} \sum_{h'} \prod_{C} \Psi_C(y', h'_C, X) \tag{2.8}$$

where h_C represents the hidden variables h involved in clique C, h' denotes all possible hidden state sequences, and y' denotes all possible class labels for the sequence. Further, we are able to compute the conditional probability $P(y|X)$ with formula 2.9 for classification,

$$P(y|X) = \sum_{h} P(y, h|X) \tag{2.9}$$

where $P(y|X)$ is obtained by marginalizing over the hidden variables h.

[2] In order to differentiate the sequence class label in HCRFs from the sequence labels in CRFs, we use the small letter y to represent the sequence class

If the exponential form 2.10 is used for the potential function $\Psi_{\mathcal{C}}(y, h_{\mathcal{C}}, X)$,

$$\Psi_{\mathcal{C}}(y, h_{\mathcal{C}}, X) = \exp\left(\sum_{i=1}^{m} \lambda_i f_i(y, h_{t-1}, h_t, X)\right) \tag{2.10}$$

for a first-order linear-chain sequence, it would be possible to push the product $\prod_{\mathcal{C}}$ along the cliques into the potential function as a summation as formula 2.11.

$$\prod_{\mathcal{C}} \Psi_{\mathcal{C}}(y, h_{\mathcal{C}}, X) = \exp\left(\sum_{t=1}^{T}\sum_{i=1}^{m} \lambda_i f_i(y, h_{t-1}, h_t, X)\right) \tag{2.11}$$

$$= \exp\left(\sum_{t=1}^{T}\sum_{i=1}^{m} \lambda_{1,i} f_{1,i}(y, h_t, X) + \sum_{t=2}^{T}\sum_{i=1}^{m} \lambda_{2,i} f_{2,i}(y, h_{t-1}, h_t, X)\right) \tag{2.12}$$

In equation 2.12, the feature function f_i is further decomposed into two types of features: node feature $f_{1,i}$ and edge feature $f_{2,i}$. The node feature function $f_{1,i}$ extracts local features based on information from a single node while the edge feature $f_{2,i}$ depends on a pair of nodes. For simplicity, we use term $\Phi(y, h, X)$ to denote the expression inside the exponentiation.

$$\Phi(y, h, X) = \sum_{t=1}^{T}\sum_{i=1}^{m} \lambda_{1,i} f_{1,i}(y, h_t, X) + \sum_{t=2}^{T}\sum_{i=1}^{m} \lambda_{2,i} f_{2,i}(y, h_{t-1}, h_t, X) \tag{2.13}$$

2.2.4 Parameter estimation

The parameter λ_i in equation 2.11 needs to be learned from training samples. Many methods can be used for estimating the HCRFs parameters [62][3], but there are two general approaches: Maximum Likelihood (ML) based approach and Expectation-Maximization (EM) based approach.

ML based approach

The most popular approach for HCRFs training is based on the maximum likelihood principle, which selects the parameters that maximise the conditional likelihood as 2.14

and 2.15,

$$\hat{\lambda} = \arg\max_{\lambda} \mathcal{L}(\lambda) \tag{2.14}$$

$$\begin{aligned}\mathcal{L}(\lambda) &= \sum_{l=1}^{n} \log P(y^{(l)}|X^{(l)}, \lambda) \\ &= \sum_{l=1}^{n} \left(\log \left(\sum_{h} \exp\left(\Phi(y^l, h, X^l)\right) \right) - \log \left(\sum_{y',h'} \exp\left(\Phi(y'.h', X^l)\right) \right) \right)\end{aligned} \tag{2.15}$$

where $\mathcal{L}(\lambda)$ is the data log-likelihood, and l indexes the training instance in a training set $\mathcal{D} = \{y^{(l)}, X^{(l)}\}_{l=1}^{n}$. In general, a regularisation term is added to the log-likelihood to prevent the values of parameters from going wild. If a L_2 regularization is applied, the objective function 2.15 would become formula 2.16,

$$\mathcal{L}(\lambda) = \sum_{l=1}^{n} \left(\log \left(\sum_{h} \exp\left(\Phi(y^l, h, X^l)\right) \right) - \log \left(\sum_{y',h'} \exp\left(\Phi(y'.h', X^l)\right) \right) \right) - \frac{\|\lambda\|^2}{2\delta^2} \tag{2.16}$$

where δ specifies how much the quadratic penalty $\frac{\|\lambda\|^2}{2\delta^2}$ is applied.

If gradient-based optimization is employed in the training, the optimal parameters would be found when the gradient of the penalised log-likelihood is zero. Let us consider the parameters of node features first. Taking the partial derivative of the objective function 2.16 with respect to the parameter $\lambda_{1,i}$, we obtain equation 2.12.

$$\begin{aligned}\frac{\partial \mathcal{L}(\lambda)}{\partial \lambda_{1,i}} &= \sum_{l=1}^{n} \left\{ \sum_{h} P(h|y^{(l)}, X^{(l)}, \lambda) \sum_{t=1}^{T} f_{1,i}(y^{(l)}, h_t, X^{(l)}) \right. \\ &\quad \left. - \sum_{y',h'} P(y', h'|X^{(l)}) \sum_{t=1}^{T} f_{1,i}(y', h'_t, X^l) \right\} - \frac{\lambda_{1,i}}{\delta^2} \\ &= \sum_{l=1}^{n} \left\{ \sum_{t=1}^{T} \sum_{h_t} P(h_t|y^{(l)}, X^{(l)}) f_{1,i}(y^{(l)}, h_t, X^{(l)}) \right. \\ &\quad \left. - \sum_{t=1}^{T} \sum_{y',h'_t} P(y', h'_t|X^{(l)}) f_{1,i}(y', h'_t, X^{(l)}) \right\} - \frac{\lambda_{1,i}}{\delta^2}\end{aligned} \tag{2.17}$$

For a linear-chain structure, the probability $P(h_t|y^{(l)}, X^{(l)})$ and $P(y', h'_t|X^{(l)})$ can be efficiently calculated using forward-backward algorithm [105]. Similarly, the partial

derivative of the objective function with respect to the parameter $\lambda_{2,i}$ of the edge feature would be calculated as equation 2.18.

$$\frac{\partial \mathcal{L}(\lambda)}{\partial \lambda_{2,i}} = \sum_{l=1}^{n} \left\{ \sum_{t=2}^{T} \sum_{h_{t-1},h_t} P(h_{t-1}, h_t | y^{(l)}, X^{(l)}) f_{2,i}(y^{(l)}, h_{t-1}, h_t, X^{(l)}) \right. \\ \left. - \sum_{t=2}^{T} \sum_{y',h'_{t-1},h'_t} P(y', h'_{t-1}, h'_t | X^{(l)}) f_{2,i}(y^{(l)}, h_{t-1}, h_t, X^{(l)}) \right\} - \frac{\lambda_{2,i}}{\delta^2} \quad (2.18)$$

Note that setting the gradients 2.17 and 2.18 to zeros does not result in any closed form solution. Thus, the parameter estimation typically resort to iterative methods such as the conjugate gradients method and the limited memory quasi-Newton method [62].

EM based approach

The parameters of HCRFs can also be estimated in a Expectation-Maximization style. The conventional EM algorithm attempts to maximise the data log-likelihood $\mathcal{L}(\lambda)$ by iteratively applying two steps: the Expectation step (E-step) and the Maximization step (M-step). The E-step calculates the expected value of the log-likelihood function using current estimate of the parameters, with respect to the conditional distribution of hidden variables h given observed data. The M-step computes parameters maximizing the expected log-likelihood found on the E-step. These parameters are then used to determine the distribution of the hidden variables h in the next E-step. It has been proved that the iterative procedure would increase log-likelihood $\mathcal{L}(\lambda)$ until the training converges to a local maximum [157].

The EM approach for HCRFs parameter estimation can be used as Kumar's and Korc's work [141][97], where a drastic approximation of the partition function $\mathcal{Z}$ needs to be computed with a single Maximum A Posteriori (MAP) labelling configuration. However, it is more convenient to apply the EM approach within a Max-Margin training framework [118][3] where the computation of $\mathcal{Z}$ is not required.

Recently, there has been an explosion of interest in structured output learning with maximum margin training [15][118][3]. The idea of the max-margin training is that maximizing the margin of the SVM scores can magnify the difference between the true label and the best runner-up, increasing the "confidence" of the classification [15]. For our classification, we aims to find the MAP labelling $\hat{y} = \arg\max_y P(y|X, \lambda)$ of a test input X where $\hat{y}$ is the estimated class label. This implies the constraint 2.19

during the training. If this were successful, the estimated label $\hat{y}$ during the training inference would be equal to the true training label $y^{(l)}$.

$$P(y^{(l)}|X^{(l)}, \lambda) \geq P(y|X^{(l)}, \lambda) \quad y \neq y^{(l)}, \forall l \tag{2.19}$$

According to equations 2.7 and 2.8, calculating the posteriori $P(y|X, \lambda)$ would require the computation of the normalization constant $\mathcal{Z}$ which can be a bottleneck during the inference. However, as both sides of the constraint 2.19 have the normalisation term, it is possible to cancel the normalisation constant and rewrite the constraint 2.19 as 2.20, where $\Phi(y, h, X) = \lambda f(y, h, X)$ as shown in expression 2.13.

$$\Phi(y^{(l)}, h, X^{(l)}) \geq \Phi(y, h', X^{(l)}) \quad y \neq y^{(l)}, \forall l \tag{2.20}$$

With constraint 2.20, the model parameters would be able to be estimated within a max-margin framework as 2.21,

$$\begin{aligned} &\max_{\|\lambda\|=1} \gamma \\ s.t. \quad &\Phi(y^{(l)}, h, X^{(l)}) - \Phi(y, h', X^{(l)}) \quad y \neq y^{(l)}, \forall l \end{aligned} \tag{2.21}$$

where γ denotes the margin between the true label $y^{(l)}$ and the best runner-up. In formula 2.21, $\|\lambda\|$ is set to 1 to prevent weights from growing without bounds. Using the transformation $\|\lambda\| \leftarrow \frac{1}{\gamma}$, formula 2.21 can be written as a standard quadratic form as 2.22,

$$\begin{aligned} &\min_{\lambda,\xi} \frac{1}{2}\|\lambda\|^2 + \frac{C}{n}\sum_{l=1}^{n}\xi_l, \\ s.t. \quad &\Phi(y^{(l)}, h, X^{(l)}) - \Phi(y, h', X^{(l)}) \geq 1 - \xi_l \\ &y \neq y^{(l)}, \quad \xi_l \geq 0, \quad \forall l \end{aligned} \tag{2.22}$$

where C is the trade-off parameter for soft-margin SVM, and ξ_l is the slack variable measuring the misclassification of sample l.

It can be seen that optimization 2.22 contains unsolved hidden variables h in the constraints. In order to perform the optimization, an EM style training can be used [3]. As listed below, the training algorithm is composed of two steps. The first step is analogue to an expectation step where hidden variables h are estimated with current parameter value. The second step is similar to the maximization step, in which model parameters λ are computed with fixed hidden structures.

1. Holding λ, ξ fixed, optimize the hidden variables h for sample $\{y^{(l)}, X^{(l)}\}$,

2. Holding h fixed, optimize λ, ξ by optimizing form 2.22.

Summary

Both ML based and EM based approaches can be used for HCRFs parameter estimation. However, due to the computation cost, ML based approach is usually applied with simple graph structures such as linear-chain or tree graph. On the contrary, the EM based approach can be used for arbitrary graph structures as the heavy computation of normalisation constants is omitted. As the HCRFs model involves hidden variables, the objective of HCRFs 2.16 has multiple local maxima. In other words, both ML training and EM training approaches can not guarantee to reach a globally optimal point [11].

2.3 Textons

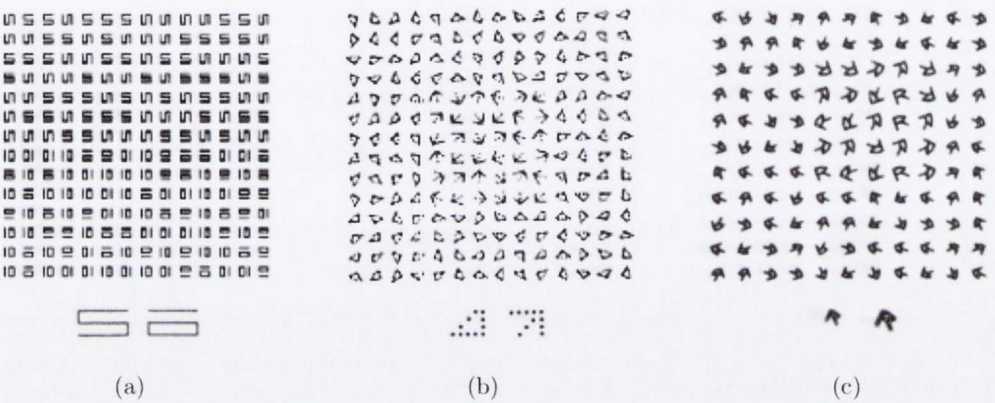
Visual texture has been extensively studied in Computer Science [159][37][1]. Recently texture analysis based on textons has also used for image retrieval and object recognition.

2.3.1 Texton theory

Textons are used for explanation of texture discrimination. They are introduced by Julesz [85] as the putative units of preattentive human texture perception, but what is the "preattentive"?

> *"**Preattentive processing** of visual information is performed automatically on the entire visual field detecting basic features of objects in the display. Such basic features include colours, closure, line ends, contrast, tilt, curvature and size. These simple features are extracted from the visual display in the preattentive system and later joined in the focused attention system into coherent objects. Preattentive processing is done quickly, effortlessly and in parallel without any attention being focused on the display"* [156].

Typically, tasks that can be performed on large multi-element displays in less than 200 to 250 milliseconds are considered preattentive [23]. Julesz investigates order statics of texture patterns and believes that only a difference in textons or in their density can be detected preattentively. These textons are groups of features detected by the early visual system, such as elongated blobs, terminators (ends of line segments), and crossings. Figure 2.4 [85] gives an example of preattentive discrimination of textures via textons. The upper and lower regions in the left image can not be told apart



(a) (b) (c)

Figure 2.4: (a) *two objects are actually the same texton;* (b) *two textons are different in the number of terminators;* (c) *the first-order statistic is different globally in the two textures*

although two components appear different in isolation. The two objects are actually the same texton as they have the same size, number of terminators and join points. Two regions in the middle image are discriminable due to the difference in terminator numbers. The texton in the upper half has three terminators while the one in the lower half has four. Two textures are also distinguishable in the right image by the global difference in their first-order statistics. The first-order statistic (or probability distribution) is simply the probability that randomly thrown dots will land on a certain colour (for example, black) of the texture [85].

2.3.2 Computation model

The texton theory developed by Julesz is typically constructed on black-and-white dot or line patterns, and is not directly applicable to gray-scale images. Malik [153][78] presents an alternative model that favours the texton analysis on gray-level images.

The model relies on filter banks on arbitrary images and identifies textons as clusters of vectors of filter outputs. By mapping pixels to textons, an image can be analysed into texton channels, each of which is a point set. Figure 2.5 presents the computation diagram of the model.

The model can be summarized into three stages.

- The first stage approximates the output of primary visual cortex V1 cells. An image $I(x, y)$ is convolved with a bank of filters tuned to various orientation and spatial frequencies. The filters can be linear filters like Gaussian derivatives, Difference of Gaussian (DOG), or Gabor filters. The output characterizes an image patch centred at (x_0, y_0) by a set of values. This is similar to characterizing an analytic function by its derivatives at a point like Taylor series approximation. The choice of the filters had been studied in Malik's early work [77], where he pointed out the need for essential nonlinearities in texture perception. Linear filters may produce identical first-order global statistics but the texture is preattentively discriminable. Malik chose half-wave rectification as the nonlinearity for biological evidence and its conservation of filter signs. Moreover, Malik applied nonlinear inhibition as a second nonlinearity to suppress spurious responses in nonoptimally tuned filters.

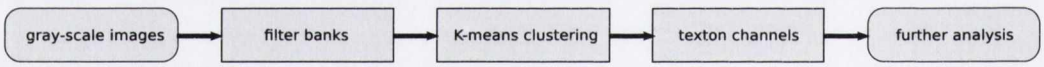


Figure 2.5: *Texton computational model for gray-scale images*

- After applying filter banks, each pixel is transformed to a vector of filter responses. In the second stage, vectors from all the images in the entire training set are aggregated and clustered using K-means. The criterion is to find K centres such that after assigning each data vector to the nearest centre, the sum of the squared distance is minimized [78]. The set of estimated cluster centres then form the textons and a visual vocabulary. If the distance in K-means is measured by Mahalanobis distance, associated covariances of clusters can also be used for the vocabulary definition [81]. The size of the vocabulary usually has an effect on the texton performance and model discrimination. Small size vocabulary may suffer from discrimination while a large vocabulary may result in overfitting. A common way to determine the number of clusters is to set a large number of

clusters initially, then the vocabulary is pruned down by merging cluster centres [153][81].

- In the third stage, each pixel in the testing image is mapped exactly to one texton by measuring the euclidean distance. A collection of all pixels with same textons constitutes a texton channel. With texton channels, a texton histograms can be build for recognition [78] [81]. It is also possible to extract contextual information by building more complex features using texton channels, for example, the texture-layout features in TextonBoost [79].

Same as interest points based methods, simply building texton histograms will lose the spatial configuration information. Graphical models HCRFs have the ability of modelling spatial or temporal structures of actions. In this thesis, we propose an extension of HCRFs to retain the spatial information in images.

Chapter 3

Recognition using Static Postures

3.1 Introduction

Hand washing gestures are very articulate bi-manual hand movements. According to our hand washing gesture definition in section 1.1.1, which follows the WHO hand washing recommendation, some gestures are quite dynamic and some gestures are relatively static. In this chapter, we ignore the dynamic nature of the hand washing gestures, and analyse the gestures with only static information. In other words, all gestures are treated as postures. The analysis follows the dense representation approach for object recognition. Hand shape is the only cue used in the recognition and the shape configuration is encoded in a dense grid style. The classifiers subsequently learn posture models for every hand washing gesture class and run a one-versus-one multi-class classification [69] for recognition.

3.2 Methodology

Recognition using static postures considers each frame in the video independently. Figure 3.1 illustrates the workflow of the method. It first localizes the hands into a square box in the image preprocessing step. Subsequently, HOG features are extracted from each frame to capture the shape information of gestures. HOG are well-known feature descriptors for object detection. Here we apply them for gesture recognition task. In the classification, linear SVM [24] is used to build gesture models. As there are 10 gestures rather than 2, a one-versus-one multi-class classification is proposed for the recognition.

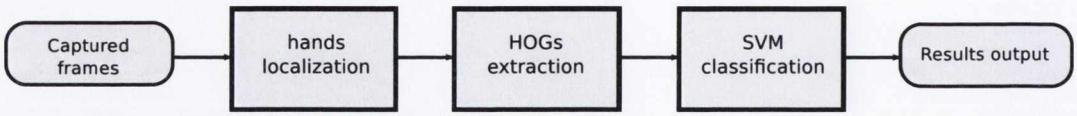


Figure 3.1: *Recognition using static postures workflow*

3.2.1 ROI detection

After capturing a frame from the camera, the first step is to localize the hands in the image, which is also called ROI detection. Background subtraction [149][50][38] and skin detection [150][161] can be used to find the hand area, however both techniques may have severe false detections due to the splashing water, skin-tone disturbers or lighting condition changes. In order to reduce the false detections, a more robust ROI detection method is developed in our evaluation, which combines the skin information and hand motion information together for the detection.

Our ROI detection calculates both skin probability and motion probability. ROI is detected as strong skin-motion area. Figure 3.2 illustrates the detection workflow. Given an input colour image, adaptive lighting compensation is employed first to adjust the image colour in order to remove the adverse effect of various lighting conditions. Then a skin mask is obtained by thresholding the multiplication result of skin probability and motion probability. After simple morphology operations which remove spurs and noise on the mask, a square box framing hands inside is allocated.

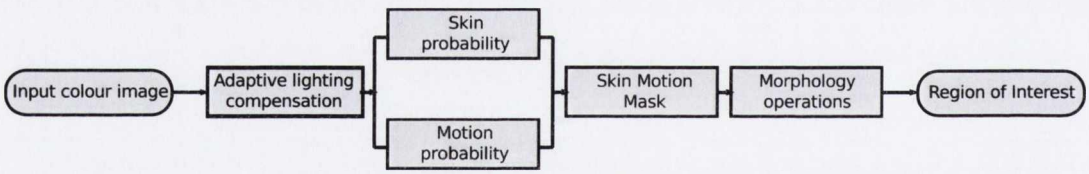


Figure 3.2: *A diagram of ROI detection*

Adaptive light compensation

The appearance of the skin-tone colour strongly depends on the lighting conditions, therefore a lighting compensation is indispensable to obtain robust skin detection result. The grey-world algorithm presented in [103] is applied for adaptive light compensation. The algorithm is based on the assumption that the spatial average of surface reflectance in a scene is achromatic. Since the light reflected from an achromatic surface is changed

equally at all wavelength, it follows that the spatial average of the light leaving the scene will be of the colour of the incident illumination.

The grey-world algorithm calculates a scale factor s_i ($i \in R, G, B$) for each colour component of every pixel. Then adjusted pixel colour could be calculated by formula 3.1:

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix}_{new} = \begin{pmatrix} s_R \\ s_G \\ s_B \end{pmatrix} \otimes \begin{pmatrix} r \\ g \\ b \end{pmatrix}_{old} \tag{3.1}$$

where $\otimes$ means element wise multiplication. The scale factor calculation of standard grey-world algorithm does not fit well for images with dark background. To solve this problem, our scale factors are calculated with equations 3.2:

$$s_R = \frac{C_{std}}{R_{avg}}, \quad s_G = \frac{C_{std}}{G_{avg}}, \quad s_B = \frac{C_{std}}{B_{avg}}$$

$$R_{avg} = \frac{\sum_i^m r_i}{n}, \quad G_{avg} = \frac{\sum_i^m g_i}{n}, \quad B_{avg} = \frac{\sum_i^m b_i}{n}$$

$$C_{std} = \frac{\sum_i^m (\max(r_i, g_i, b_i) + \min(r_i, g_i, b_i))}{2n} \tag{3.2}$$

Here m stands for the number of pixels in the image and n stands for the number of non-black pixels in the image to avoid over compensation in images with dark background. Figure 3.3 shows an example of employing the adaptive light compensation on hand washing videos.

Skin probability computation

With lighting compensated images, the skin probabilities are then computed using a histogram-based non-parametric skin model with Bayes classifier [161]. The skin probability of each pixel is estimated by formula 3.3. $P(skin)$ and $P(nonskin)$ are the prior probabilities which can be estimated from the overall number of skin and non-skin pixels in a training set. Probabilities $P(rgb|skin)$ and $P(rgb|nonskin)$ can be directly computed from skin and non-skin colour histograms. The histograms are built by quantizing the colour space RGB into a number of bins $rgb \in RGB$ for both

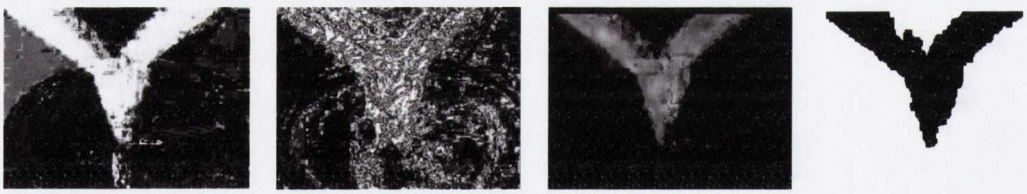
(a) (b)

Figure 3.3: *Adaptive light compensation. (a) before light compensation; (b) after light compensation*

skin and non-skin classes. After normalization, two histograms for skin and non-skin classes are obtained.

$$P(skin|rgb) = \frac{P(rgb|skin)P(skin)}{P(rgb|skin)P(skin) + P(rgb|nonskin)P(nonskin)} \tag{3.3}$$

Figure 3.4(a) shows an example of the skin detection result. It can be seen that a portion of background is detected with a considerable high skin probability. In practice, it is hard to set a universal threshold to separate the real skin area and background due to the uncontrollable illumination.



(a) (b) (c) (d)

Figure 3.4: *Skin motion mask processing.* (a) *skin probability;* (b) *motion probability;* (c) *skin motion probability;* (d) *skin motion mask*

Motion probability computation

To overcome the skin detection difficulty mention above, motion probability is computed to suppress the false skin detection. The motion probability is calculated by averaging and normalizing optical flow from five continuous frames as illustrated in figure 3.5. The optical flow is calculated between the current frame and one, two, three,

four frames before respectively. Horn-Schunck method [82] is used for the optical flow computation. Comparing to optical flow which is computed by Lucas-Kanade method [82], Horn-Schunck method can provide a smoother high density of flow vectors which is preferred in our hand motion probability estimation. This is because Horn-Schunck method is a global method which has a constraint of smoothness for solving the aperture problem of optical flow computation. The maybe inadequate flow information from the inner hand can thus be filled in from the motion boundaries estimated near the hand contours.

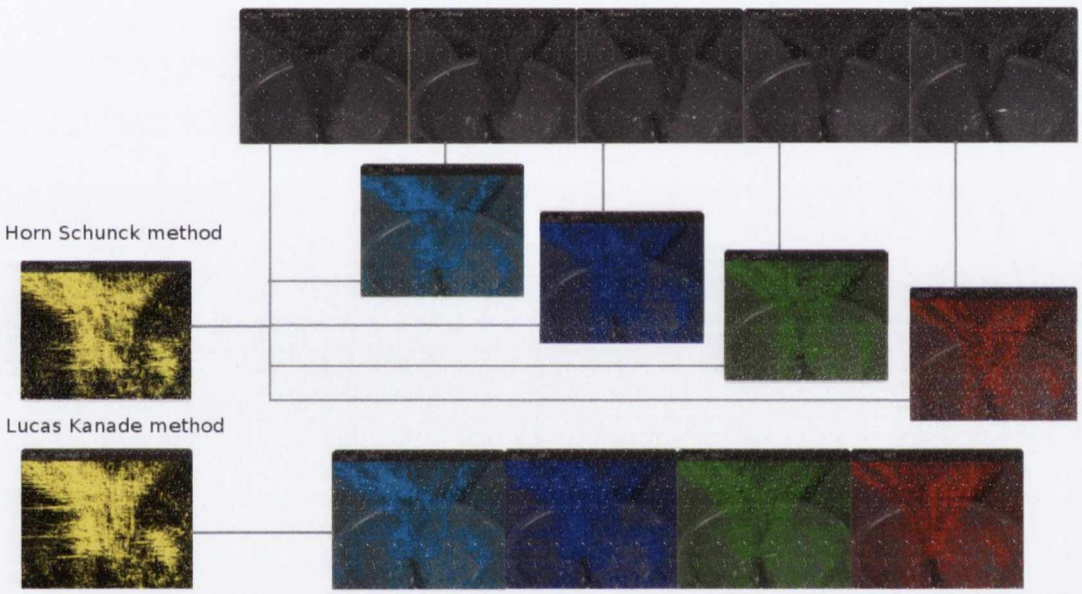


Figure 3.5: *Motion probability computation*

Skin-motion mask

A skin-motion probability of current frame is obtained by multiplying the skin probability with the motion probability as shown in figure 3.4(c). Then the hands during hand washing activity would possess a high skin-motion probability. As shown in figure 3.4(d), the hand can easily be segmented out by setting a universal threshold, and the false detected hand area from both skin measurement and motion measurement is clear out. With skin-motion probability, it is also possible to discard those frames in which hands are not moving.

Morphology operations

The aforementioned skin-motion mask may still possess some noises as small holes and spots. A sequence of morphology operations is then performed to remove those noises. These operations consist of dilation, connected component analysis and erosion. Dilation operation removes small holes with the skin-motion mask. Connected component analysis then gets rid of small regions which are mistakenly detected as hand area. Erosion finally removes small anomalies at the boundaries of the mask.

Square bounding box

Once a refined skin-motion mask is obtained, a bounding box is created as the region of interest to facilitate subsequent HOG feature extraction. The procedure of drawing a bounding box is described in diagram 3.6. As shown in figure 3.7, the vertical symmetry axis is detected first. Then a horizontal line searches down along the axis from the top until dense hand area is reached such that the upper boundary of the box is found. Similarly, the low boundary can be found by searching from the bottom. Once obtaining the upper and lower boundaries, the right and left boundaries are found by searching along the horizontal line from right and left until dense hand area. Generally the procedure will find a rectangle. We then take the long sides as references and extend the short sides such that a square bounding box is fixed.

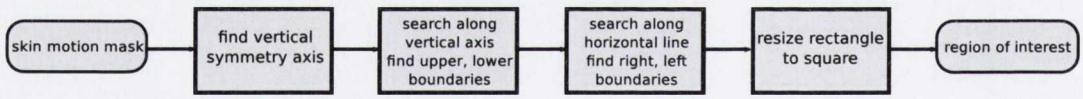


Figure 3.6: *A bounding box of ROI*

3.2.2 HOGs extraction

In order to describe the hand shape in every ROI, single frame HOG features are extracted. HOG features have given promising performance in many applications [122][160][128]. The aim of this method is to describe an image by a set of local histograms. These histograms count occurrences of gradient orientation in a local part of the image.

To compute the HOG features, the square image patch framed in the ROI is first extracted and resized into a 128 $\times$ 128 ROI image. This colour ROI image is then

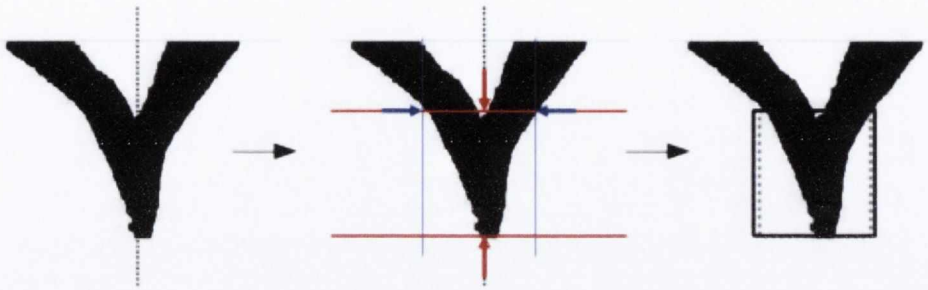


Figure 3.7: *ROI extraction*

converted to grey level and gradients are calculated. Next, the ROI image is split into square cells with a predefined size. In each each, a histogram of gradients is computed by accumulating votes into bins of orientation. Each vote is weighted by the magnitude of the gradient vector so that the histogram takes into account the importance of the gradient at a given point.

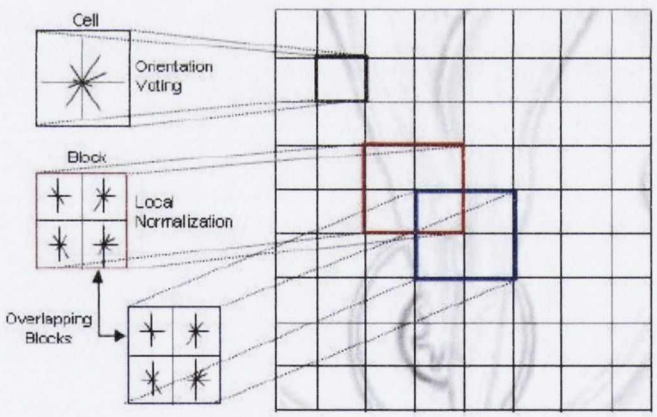


Figure 3.8: *HOGs computation*

Every 2×2 cells are group into a block as illustrated in figure 3.8. Within each block, cell histograms are locally normalised according to values of the neighbouring cell histograms. The normalization is necessary such that HOG features are insensitive to contrast variations in images. In our experiments, *L2-norm* scheme 3.4 is used,

$$v \rightarrow \frac{v}{\sqrt{\|v\|_2^2 + \varepsilon}} \tag{3.4}$$

where v is the cell histograms within a block and ε is a small regularization constant to avoid zero in denominator. Note that the blocks can be overlapped with each other and a histogram from a given cell would be involved in several block normalisations. Thus, the features are going to have some redundant information which, according to the work of Dalal et al. [122], can improve the performance.

When histograms from all cells have been computed and normalized, a HOG descriptor of the ROI image is built by concatenating all histograms into a single vector. The vector dimension can be calculated with equation 3.5.

$$\begin{aligned} HOGsDimension &= Blocks_{Row} \times Blocks_{Col} \times CellsInBlock_{Row} \\ &\quad \times CellsInBlock_{Col} \times BinsInCell \end{aligned} \tag{3.5}$$

For example, if cell of size 16×16 is used for HOG computation, there would be 8 cells and 7 blocks in every row and column of a 128×128 ROI image. The dimension of the HOG feature vector with 16 orientation bins in each cell would be 3136 according to the above equation.

3.2.3 SVM

SVM is a wildly used classification method in Computer Vision. It minimises a bound on the generalisation error based on the structural risk minimisation principle [24] and would have good performance on novel data. In our evaluation, linear SVM is applied to train the classifiers, which searches the optimal hyperplanes in the original feature space.

Our classification is a multi-class classification problem, thus a one-against-one approach [69][45] is applied in our evaluation. The approach constructs $k(k-1)/2$ binary classifiers for different pairwise combination of 10 gestures, where $k = 10$ is the number of gestures. During the testing, a voting strategy is used: a given sample x is assigned to either class i or class j by one of the binary classifiers, and the vote for class i or j is incremented by one correspondingly. After being tested by all binary classifiers, the sample x is predicted to be the class with the largest vote. This approach is also referred as the *max-wins* strategy.

3.3 Evaluation

Our evaluation of recognition using static postures method runs a 5-fold cross validation as described in section 1.1.4. The HOG features with different parameter settings are extracted from each detected ROI in every frame. The libsvm library [22] is then used to train the SVM classifiers.

HOGs settings

Various HOG parameters as listed in table 3.1 are evaluated. These parameters result in coarse to fine descriptions of hand postures. Take the first row of table 3.1 as an example. The "Sign" column indicates if the gradient orientation is measured in a range of $0^o \sim 180^o$ (unsigned) or $0^o \sim 360^o$ (signed). If "unsigned" is used, an orientation in the range of $180^o \sim 360^o$ would be converted into a range of $0^o \sim 180^o$, which gives less orientation discrimination than using the full range. The "Bins" indicates the quantization levels for the orientation. More bins give finer measurement in orientation. An unsigned 8 bins setting means that the orientation interval in each bin would be 22.5^o. "Cells" and "Blocks" are measured in pixels as described in section 3.2.2. As each our block is fixed to contain 2×2 cells, for 32×32 cells, the size of the block would be 64×64. Given a HOG parameter setting as the first row in table 3.1, according to the equation 3.5 the dimension of the HOG features would be 288.

Dimension	Bins	Cells	Blocks	Sign
288	8	32	64	unsigned
576	16	32	64	signed
1568	8	16	32	unsigned
3136	16	16	32	signed

Table 3.1: *HOGs settings for recognition using static postures*

Recognition performance

The linear SVM classifiers are trained and tested with different HOG features. The soft-margin controlling parameter C in SVM is set to 1 by libsvm default. Table 3.2 lists their testing results. These results appear to confirm that the hand washing gestures can be recognized in a way of recognizing static hand postures while ignoring the dynamic characteristic of the hand washing gestures. From the table 3.2, it can be

seen that 3136 dimension HOG give the best recognition performance. This implies that the finer details of the hand postures are important for the recognition in this method. Meanwhile, the very high dimension of HOG also makes the hyperplane searching of SVM much easier in the original feature space. On the other hand, the very high dimension does cause a very heavy computation load, which could be an issue for real-time performance. Relatively, 1568 dimension HOG give very similar recognition performance to 3136 dimension HOG but only have a half of the dimension. This implies that when fine grid is applied to the HOG calculation, the "signed" option is not very important for hand washing posture recognition.

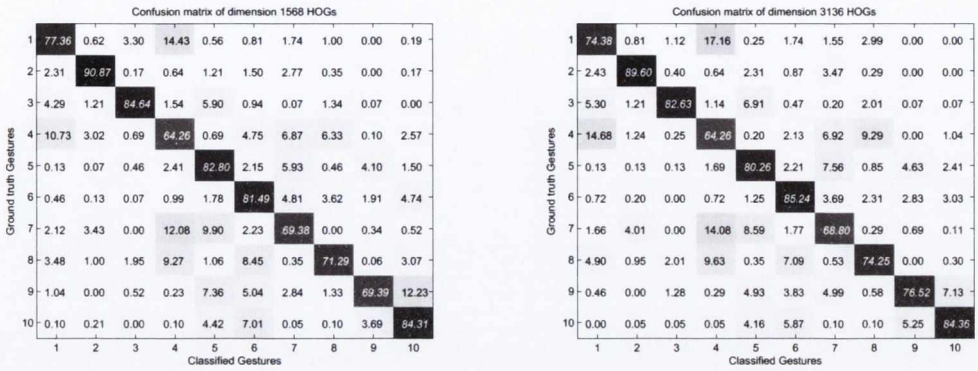
Dimension	Accuracy	Macro F	Micro F
288	$69.56 \pm 5.34\%$	0.7082	0.6956
576	$72.64 \pm 6.71\%$	0.7383	0.7264
1568	$77.57 \pm 6.39\%$	0.7859	0.7757
3136	$78.04 \pm 5.94\%$	0.7891	0.7804

Table 3.2: *Linear SVM classification results*

Figure 3.9 gives the confusion matrices of classification using 3136 dimension and 1568 dimension HOG. It can be seen that both suffer the misclassification between gesture 1 and gesture 4 which have very similar hand washing gesture appearance in nature, especially when a frame is judged independently. The gesture 4 also has difficulties with gestures 7 and 8. This may be because the clasped fingers are not much vertical to the other palm. By definition, gestures 2 and 3 look similar to each other, and so do gestures 9 and 10. However, the confusion matrices show that they can be well separated using high dimension HOG features with linear SVM classifiers.

3.4 Summary

In this chapter we elaborate and evaluate the method of recognizing hand washing gestures using static postures. The method considers each frame independently and the dynamic features from the gestures are disregarded for classification. A skin-motion ROI detection method is introduced, which can drastically decrease the false positives in hand area detection with either skin detection or motion detection alone. The static HOG features extracted from ROIs are used to train linear SVM classifiers. The classification results show that the fine-grid HOG features have better performance



(a) (b)

Figure 3.9: *Confusion matrices of* (a) *dimension 1568 HOGs,* (b) *dimension 3136 HOGs*

than the coarse-grid HOG features. This is mainly because the fine-grid features can provide detailed hand shape information for classification. As no kernel tricks is used in the SVM training, the testing phase of the method is fast. However, the fine-grid HOG could give very high dimension feature vectors and result in a heavy computation load during the feature extraction.

Chapter 4

Recognition using Sequence Labelling

4.1 Introduction

Hand washing gestures are essentially continuous hand movements, which implies that the dynamic nature of the gestures is a very important characteristic for recognition. In this chapter we analyse hand washing gestures with not only the static information extracted from every frame but also the dynamic information from the hand motion sequences. Hand washing gesture within a short period of time is modelled as a frame sequence with HCRFs model [11]. In contrast to chapter 3 where every gesture frame is predicted independently, recognition using sequence labelling method considers the whole sequence for recognition and the state of every frame in the sequence is estimated within a context from previous and following frames.

4.2 Methodology

In general, people make a judgement of the correctness of a gesture upon a short time period of hand washing. To enable the hand washing gesture recognition system operates similarly, in this chapter we model the hand movements within a short time period to be a hand washing gesture sequence. A unique gesture class is assigned to the whole sequence, and all frames within the sequence would have the same gesture class as the sequence. It is assumed that the hand washing gesture would not change abruptly from one to the other within the short time period.

To build this sequence, single frame HOG features are extracted first as described in chapter 3. These HOG features or their combinations (concatenated HOG of multiple

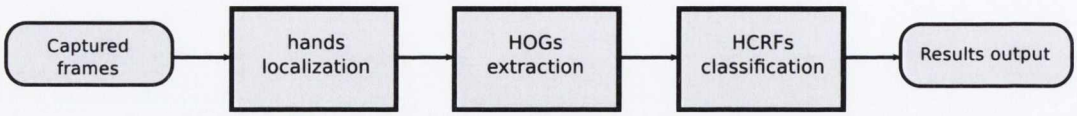


Figure 4.1: *Recognition using sequence labelling workflow*

frames) would be modelled as building blocks [60] of hand movements. Linear-chain HCRFs is applied to model the internal structure of gesture sequences. Linear-chain HCRFs is a popular sequence labelling tool in speech modelling [6] and Computer Vision [54][139]. It takes a whole sequence as an input and only outputs a single class label. With the linear-chain HCRFs model, a hand washing gesture is assumed to have a linear-chain structure, and each building block of hand movements is a node of the chain, which a hidden state will be assigned to. Figure 4.1 illustrates the whole procedure of recognition using sequence labelling method. It can be seen that the recognition using sequence labelling method also requires ROI detection for HOG feature extraction but will use the HCRFs model for classification instead of SVM. Note that the HCRFs classification stage actually contains two steps: the training step and the testing step of HCRFs.

In the training step, all gesture sequences in the training set are pooled together and fed into HCRFs for generating a multi-class HCRFs model. During the testing step, in order to report gesture class for each frame of a hand washing video, a time-shift window having current frame in the middle is applied. A short video segment from a long sequence is extracted by the time-shift window, and is classified by the trained HCRFs model. The label output from the HCRFs model is then assigned to the current frame as the classification result.

As the hand localization and HOG extraction in figure 4.1 have been described in chapter 3, in the rest of this section we will mainly focus on the stage of HCRFs classification.

4.2.1 Linear-chain HCRFs model

Linear-chain HCRFs is an extension of linear-chain CRFs with hidden variables. Given a sequence observation $X = \{x_1, \cdots, x_t, \cdots, x_T\}$, a sequence of hidden states h are inferred from X, and are used to explain the sequence class label y, as shown in figure 4.2. In our hand washing gesture recognition, X would represent the static HOG features extracted from all frames in a sequence. Each frame in the sequence can be

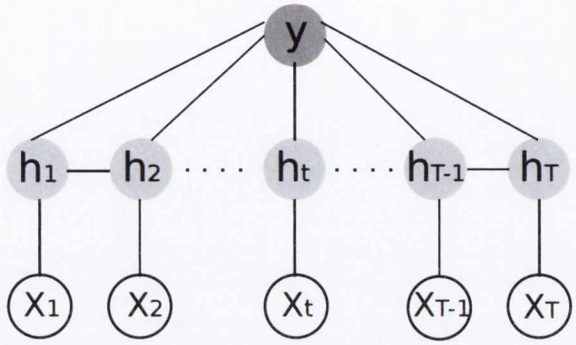


Figure 4.2: *Linear-chain HCRFs for sequence labelling*

regarded as a node in a chain, and is assigned with a hidden state h_t by the HCRFs model. The gesture class y of the sequence would then be estimated based on the inferred hidden state sequence h.

Hidden variables

There is a large number of variance in hand washing gestures. Learning different gesture categories directly with observed hand washing data may not well capture this variance. In contrast to the CRFs model, HCRFs model use hidden variables to represent some "shared" variance among observations. With HCRFs, observed hand washing data can be represented by a number of unobserved intermediate states which greatly decrease the degree of variance. In our linear-chain HCRFs, a hidden state h_t represents an unobservable hand shape configuration within the continuous hand movement. This state can be inferred with or without information from neighbouring frames. All these states are linked by a first-order Markov chain where the current frame state h_t is only based on the previous frame state h_{t-1} according to the Markov property [112].

Context window

HCRFs model does not require independence assumption on the observed data X, which is a prerequisite assumption in HMM model. This relaxation of observation independence can greatly help the recognition as the contextual information consisted of rich overlapping features can be used. In our HCRFs gesture modelling, a temporal context window is constructed as shown in figure 4.3. Given a context window size parameter w, current frame HOG features are concatenated with HOG features from w

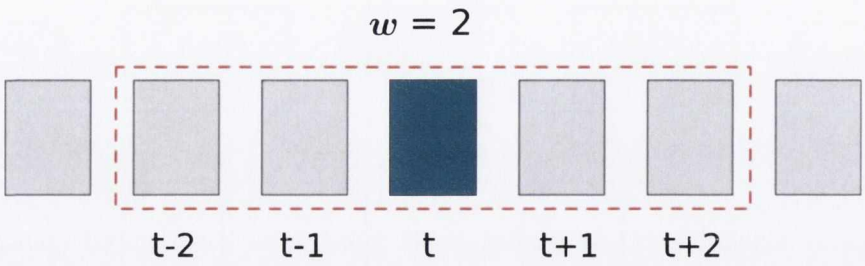


Figure 4.3: *An example of context window used in recognition via sequence labelling. The green image indicates current frame and the context window size parameter is set as $w = 2$.*

frames before current frame and w frames after current frame. In other words, features from $2w + 1$ frames are used as the observation of current frame. As shown in figure 4.3, if $w = 2$, a context window with 5 frames is constructed. The static HOG features extracted from these 5 frames may not be independent with each other, but with the help of HCRFs model it is possible to concatenate them together as a long description vector of current frame. Therefore, each frame in HCRFs is no longer considered independently but within a context from neighbouring frames. In our HCRFs model, if there is not enough frames can fill in a context window, which usually happens at the beginning or the end of a sequence, the zero-padding technique is applied to form the long concatenated vector.

The model

With the hidden states and context window described above, we are able to introduce a specific form of the potential function used in our linear-chain HCRFs model. The potential function $\Psi(y, h, X; \lambda, w)$ is defined as formulae 4.1 and 4.2,

$$\Psi(y, h, X; \lambda, w) = \exp\left(\Phi(y, h, X; \lambda, w)\right) = \exp\left(\lambda \cdot f(y, h, X; w)\right) \tag{4.1}$$

$$\lambda \cdot f(y, h, X; w) = \sum_{t=1}^{T} \lambda_1 \cdot f_1(h_t, x_t; w) + \sum_{t=1}^{T} \lambda_2 \cdot \mathbb{1}(y, h_t) + \sum_{t=2}^{T} \lambda_3 \cdot \mathbb{1}(y, h_{t-1}, h_t) \tag{4.2}$$

where "$\mathbb{1}$" denotes the indicator function. An indicator function gives value 1 if desired elements in the brackets present, otherwise it would give 0. For example,

$$\mathbb{1}(y, h_t) = \begin{cases} 1 & \text{if } y = 1 \text{ and } h_t = 3 \\ 0 & \text{otherwise} \end{cases}$$

Both $f_1(h_t, x_t; w)$ and $\mathbb{1}(y, h_t)$ are node features. The first type of feature $f_1(h_t, x_t; w)$ has a parameter w to indicate how much contextual information is used for current frame state inference. It is weighted by λ_1. The inner-product $\lambda_1 \cdot f_1(h_t, x_t; w)$ measures the compatibility between a hidden state h_t and the local observations x_t at frame t. The second type node feature is extracted with an indicator function $\mathbb{1}(y, h_t)$. Its weight vector λ_2 can be interpreted as a measure of the compatibility between a gesture class y and the hidden state h_t. Indicator function $\mathbb{1}(y, h_{t-1}, h_t)$ represents the third type edge features. Analogously, its weight λ_3 can be regarded as a measure of the compatibility between the gesture label y and an edge connecting two neighbouring hidden states (h_{t-1}, h_t). The parameter tying technique is used for all feature weights in HCRFs model, which means that all the weights λ are position independent.

4.2.2 Model training

The linear-chain HCRFs model can be trained with ML based methods as described in section 2.2.4. However, the training needs to compute the normalization constant $\mathcal{Z}$ and would be very slow for our large-scale high dimension hand washing data. Therefore, we adapt Wang's max-margin HCRFs method [3] where the computation of $\mathcal{Z}$ is not required. Wang's method is a EM based max-margin training approach. It was proposed combining large-scale global template features and part-based local features in a principled way for action recognition. On the contrary, our HCRFs model does not have any global features about the sequences, which directly model the relationship between the observation X and the gesture class y.

Following the description in section 2.2.4, the max-margin training aims to find the parameter λ by optimizing a quadratic form 4.3,

$$\begin{aligned} & \min_{\lambda,\xi} \frac{1}{2}\|\lambda\|^2 + \frac{C}{n}\sum_{l=1}^{n}\xi_l, \\ s.t. \quad & \Phi(y^{(l)}, h^{(l)}, X^{(l)}; \lambda, \omega) - \Phi(y', h', X^{(l)}; \lambda, \omega) \geq 1 - \xi_l \\ & y' \neq y^{(l)}, \quad \xi_l \geq 0, \quad \forall l \end{aligned} \tag{4.3}$$

where ω is the pre-defined window size; l denotes each training sample and ξ_l is corresponding slack variable; $\Phi(y, h, X; \lambda, w)$ represents the inner-product $\lambda \cdot f(y, h, X; w)$. The training recursively runs following two steps until a maximum iteration number is reached:

1. Holding current λ, ξ values fixed, for each training sample the hidden structure h for every gesture class y, including the true class $y^{(l)}$, is inferred as

$$h = \arg\max_{h'} \Phi(y, h', X^{(l)}; \lambda, \omega)$$

2. With the inferred h, we search the optimal λ, ξ in optimization 4.3 with any quadratic programming (QP) solver. The output values of λ and ξ would then be used in the next iteration.

In our training, we actually optimize the dual form of the above optimization such that there is no slack variable ξ involved in training and the whole optimization can be decomposed into a series of smaller QPs, which is preferred when a large data set is used for training.

For the dual optimization, the quadratic form 4.3 can be firstly rewritten as optimization 4.4

$$\begin{aligned} &\min_{\lambda,\xi} \frac{1}{2}\|\lambda\|^2 + \frac{C}{n}\sum_{l=1}^{n}\xi_l, \\ &s.t. \quad \lambda \cdot \varphi(y, X^{(l)}) + \xi^{(l)} - \delta(y, y^{(l)}) \geq 0 \quad \forall y, \forall l \end{aligned} \tag{4.4}$$

where $\varphi(y, X^{(l)})$ denotes the vectors of feature difference, which are calculated with equation 4.5 for each possible gesture label y,

$$\varphi(y, X^{(l)}) = f(y^{(l)}, h^{(l)}, X^{(l)}; \lambda, \omega) - f(y, h, X^{(l)}; \lambda, \omega) \tag{4.5}$$

The vector $\delta(y^{(l)}, y)$ represents the binary model loss used in our model, which is formed as 4.6.

$$\delta(y^{(l)}, y) = \begin{cases} 1 & \text{if } y \neq y^{(l)} \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

Next, Lagrange multipliers $\boldsymbol{\alpha}$ are allocated to obtain the prime objective of the optimization $\mathcal{L}_P$ 4.7,

$$\mathcal{L}_P = \frac{1}{2}\|\lambda\|^2 + C\sum_{l=1}^{n}\xi_l - \sum_{l,y}\alpha_{l,y}\left[\lambda \cdot \varphi(y, X^l) + \xi_l - \delta(y^{(l)}, y)\right] \tag{4.7}$$

where all dual variables have non-negative values $\alpha_{n,y} \geq 0$. Setting the derivatives of $\mathcal{L}_P$ with respect to λ and ξ_l to zeros, we can derive equations 4.8 and 4.9.

$$\lambda = \sum_{l=1}^{n} \sum_{y} \alpha_{l,y} \varphi(y, X^{(l)}) \tag{4.8}$$

$$\sum_{y} \alpha_{l,y} = C \tag{4.9}$$

Substituting 4.8 and 4.9 back to $\mathcal{L}_P$, we obtain the dual form of the optimization as 4.10,

$$\begin{aligned} &\max_{\alpha} \sum_{l=1}^{n} \sum_{y} \alpha_{l,y} \delta(y^{(l)}, y) - \frac{1}{2} \| \sum_{l=1}^{n} \sum_{y} \alpha_{l,y} \varphi(y, X^{(l)}) \|^2 \\ s.t. \quad &\sum_{y} \alpha_{l,y} = C, \quad \alpha_{l,y} \geq 0, \quad \forall y \end{aligned} \tag{4.10}$$

In the dual optimization 4.10, The size of dual variables $\boldsymbol{\alpha}$ would be $n \times |y|$. When a large training set is used, the optimization can become infeasible for any generic QP solver. To overcome this problem, the quadratic decomposition technique [135][127] is applied such that the optimization is decomposed into a series of smaller QPs. In our case, a QP solver would only involve dual variables $\{\alpha_{l,y}\}$ from a particular training sample while all the other variables $\{\alpha_{k,y} : \forall k : k \neq l\}$ are fixed. Thus, in each QP solver, only $|y|$ dual variables need to be solved.

With the quadratic decomposition, we get the dual objective of the optimization $\mathcal{L}_D$ as 4.11,

$$\begin{aligned} \mathcal{L}_D = &\sum_{y} \alpha_{l,y} \delta(y^{(l)}, y) - \frac{1}{2} \Bigg[\| \sum_{y} \alpha_{l,y} \varphi(y, X^{(l)}) \|^2 \\ &+ 2 \sum_{y} \alpha_{l,y} \varphi(y, X^{(l)}) \left(\sum_{k:k \neq l} \sum_{y} \alpha_{k,y} \varphi(y, X^{(k)}) \right) \Bigg] \\ &+ \text{other terms not involving } \{\alpha_{l,y}\} \end{aligned} \tag{4.11}$$

where the summation $\sum_{k:k\neq l} \sum_{y} \alpha_{k,y} \varphi(y, X^{(k)})$ can be calculated through equation 4.12.

$$\sum_{k:k \neq l} \sum_{y} \alpha_{k,y} \varphi(y, X^{(k)}) = \lambda - \sum_{y} \alpha_{l,y} \varphi(y, X^{(l)}) \tag{4.12}$$

The dual variables $\alpha_{l,y}$ therefore can be computed by optimizing 4.13 for each training sample,

$$\max_{\alpha_{l,y}} \mathcal{L}_D \quad s.t. \quad \sum_y \alpha_{l,y} = C, \quad \alpha_{l,y} \geq 0 \tag{4.13}$$

Consequently, the parameter λ can finally be obtained with equation 4.8.

The iterative learning procedure described above continuously updates the parameter λ at each training sample, and the iteration terminates when a predefined maximum iteration number is reached. Note that in each iteration, the hidden structure of current training sample needs to be inferred. In our training algorithm, the Viterbi algorithm [105] is applied for the hidden structure inference.

4.2.3 Model testing

HCRFs is a sequence based classification method. It outputs a single label as the classification result for the whole testing sequence. In this chapter, most evaluation is measured in terms of sequence. However, the sequence measurement may be not practical for continuous video stream in real situation. Moreover, our evaluation also needs frame-based classification output for cross-comparison between different recognition methods. Therefore, a time-shift window is applied in order to give frame-based classification results. A time-shift window around current frame segments a short sequence from the long sequence for testing. The HCRFs testing result of this sequence segment is regarded as the gesture label of current frame. The time-shift window operates very similarly to the context window. Differently, the information extracted in time-shift window is used to model a short sequence while in context window, the information is only used to model a node in a chain.

4.3 Evaluation

In the evaluation of recognition using sequence labelling method, we implement the algorithm described above in Matlab. We apply the 1568 dimension HOG as the hand shape descriptor for every ROI. Based on our evaluation in recognition using static posture method, it can be seen that the recognition with 1568 dimension HOG is comparable to the recognition with 3136 dimension HOG in accuracy. However, although it is still a high dimension descriptor, 1568 dimension HOG have only half

dimension of the 3136 HOG, which make it much easier to be operated in HCRFs when context window is applied.

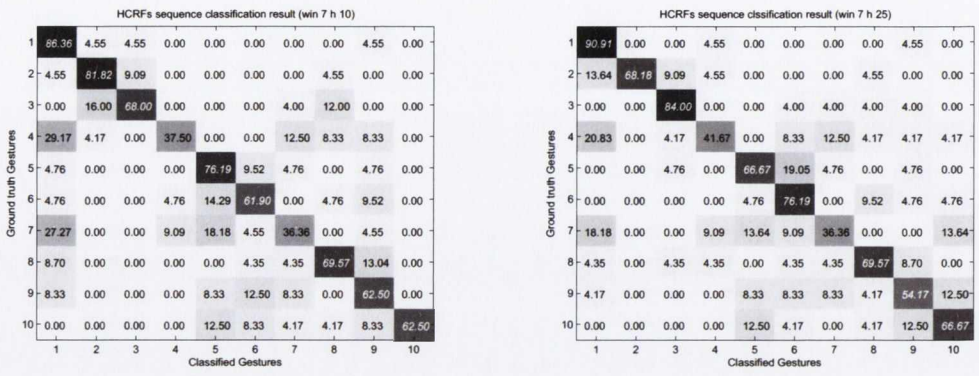
4.3.1 Context window size

In the evaluation, the context window in HCRFs is experimented first. We fixed the number of hidden variables at 10, and report the classification results in terms of sequences. The context window is smoothly enlarged in the evaluation. The parameter w is tested from 0 to 5, which gives the context window size $1, 3, 5, 7, 9$ and 11. A context window with size 1 means that only features from current frame is used for state inference. A context window with size 3 means that features from one frame before and one frame after is also used for current frame state inference. The meaning of other context window sizes can be deduced by analogy. For the frames near the start and end of the videos, zero padding is applied to fill in the context window when no video frames are available. Training samples from all classes are polled together to train a single multi-class HCRFs model. Table 4.1 lists the sequence recognition results with different context window sizes.

Size	Accuracy	Macro F	Micro F
1	$62.78 \pm 11.45\%$	0.6662	0.6278
3	$63.23 \pm 8.81\%$	0.6505	0.6323
5	$63.52 \pm 9.19\%$	0.6696	0.6325
7	$64.10 \pm 4.71\%$	0.6723	0.6410
9	$59.65 \pm 15.05\%$	0.6288	0.5965
11	$56.24 \pm 10.61\%$	0.5920	0.5624

Table 4.1: *Classification results with different context window size in HCRFs*

From the above table, it can be seen that the recognition performance is gradually improved when the context window size is increased until size 7. Figure 4.4(a) shows the confusion matrix of the classification when 7 frames are used in a context window. The table result tallies with Wang's finding [139] that applying contextual information in the model can improve the recognition performance. On the other hand, in the results our improvement with long context window is slight. This is probably because that the information for discriminating gestures has been mostly included in the well-defined single frame static HOG features. The context window does not introduce much more information for classification. In other words, the information extracted



(a) (b)

Figure 4.4: *Confusion matrices of HCRFs with 10 and 25 hidden variables:* (a) *7 frames in a context window, 10 hidden variables;* (b) *7 frames in a context window, 25 hidden variables*

from current frame has contributed the most for determining the decision boundaries in the model. The additional information introduced by the context window may mostly be tolerated during the learning and does not adjust the decision boundaries much.

The performance starts decreasing after 9 frames are used in a context window. This is probably because the overfitting of the model since very high dimension (142220 dimensions for the context window of size 9) feature vectors are employed. Therefore, as a complement to Wang's finding, we claim that the recognition improvement from contextual information depends on the features of primitives. Better features description would require smaller context window size to improve the recognition performance without a risk of model overfitting.

4.3.2 Number of hidden variables

Next, we evaluate the recognition using sequence labelling method with different number of hidden variables. We fix the context window size at 7 as it shows the best accuracy in our previous experiments. HCRFs with $5, 10, 15, 20, 25$ and 30 hidden variables are evaluated. Table 4.2 lists the classification results, and figure 4.4(b) shows the confusion matrix when 25 hidden variables are used in HCRFs.

It can be seen that except the model with 5 hidden variables, the other models have close recognition performance. In order to verify this, we perform a Student's t-test on these models with following null hypothesis:

No. of states	Accuracy	Macro F	Micro F
5	$51.64 \pm 12.01\%$	0.5189	0.5164
10	$64.10 \pm 4.71\%$	0.6723	0.6410
15	$58.87 \pm 15.26\%$	0.6076	0.5887
20	$58.78 \pm 9.52\%$	0.6149	0.5878
25	$65.28 \pm 10.43\%$	0.6630	0.6528
30	$64.30 \pm 9.71\%$	0.6532	0.6430

Table 4.2: *Classification results with different number of hidden variables in HCRFs*

there is no statistical significant difference between the performance with 25 hidden variables and performance with other settings.

T-test	5 against 25	10 against 25	15 against 25	20 against 25	30 against 25
P-value	0.0004	0.5947	0.3125	0.3756	0.2255

Table 4.3: *T-test results about the number of hidden variables in HCRFs.* $5, 10, 15, 20, 25$ *and* 30 *represent the t-test samples with designated number of hidden variables*

In this t-test the model with 25 hidden variables, which gives the highest accuracy in the result table, is used as a performance reference. Each t-test sample is built with a group classification results of a model. As described in section 1.1.4, we run a model classification by randomly picking 20% videos as the testing set and applying the rest for training. This procedure repeats 15 times such that 15 classification results would be generated for a model. These 15 results then form a sample in our Student's t-test. We apply the two samples two-tailed t-test for the hypothesis testing. One sample is from the model with 25 hidden variables, and another is from one of the other models. The p-values from all t-tests are listed in table 4.3 and plotted in figure 4.5. "5 against 25" means that a paired t-test is performed with a sample from a model using 5 hidden variables and a sample from a model using 25 hidden variables. Setting the alpha level at 0.1, it can be seen that except the "5 against 25" t-test, all the others fail to reject our null hypothesis. This means that as long as enough hidden variables are supplied in the HCRFs model, the size of the hidden variable pool does not play a critical role in the classification. The model may be dominated by some of the hidden variables even though more hidden variables are available. In fact, although all gesture classes share the same pot of hidden variables, each gesture has a unique distribution

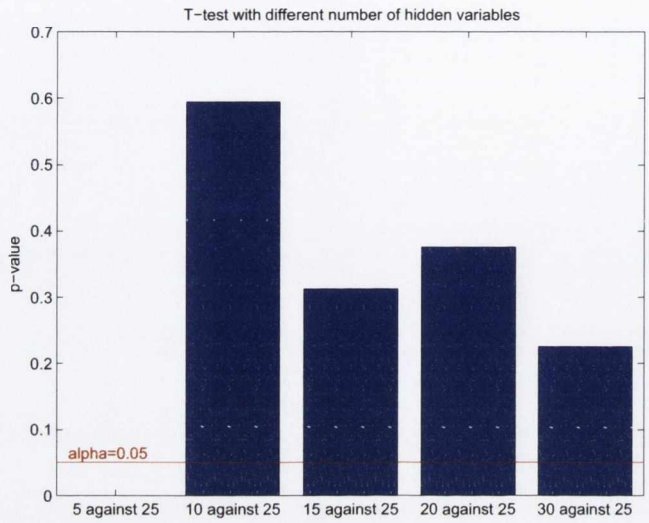
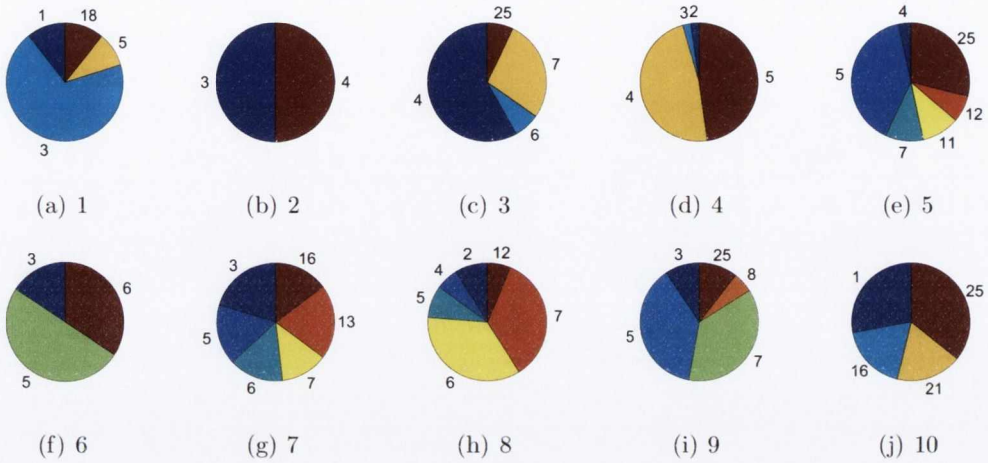


Figure 4.5: *T-test on HCRFs performance with different number of hidden variables*

of hidden states as shown in figure 4.6. It can be counted that in figure 4.6 only 15 hidden variables are used while the other 10 hidden variables do not contribute to the classification.

4.3.3 Temporal resolution

Since the temporal characteristic of hand washing gestures is modelled by the linear-chain sequence, it would be interesting to study the effect of different sequence temporal resolution on the recognition performance. To accomplish this study, we choose the test videos only from the white board subset. The white board hand washing video set is recorded at a rate of 30 frames per second. For each test video, we downsample it to be videos with three different frame rates: 25 fps, 15 fps and 10 fps. These videos are tested by previously trained HCRFs models with 25 hidden variables and 7 frames in a context window. The test results are listed in table 4.4. It can be seen that in general there is no much performance difference among sequences with different frame rates. This conveys us that the frame rate is not a critical factor for the HCRFs model in hand washing recognition. However, a poor frame rate such as 10 fps does decrease the performance slightly.



(a) 1 (b) 2 (c) 3 (d) 4 (e) 5

(f) 6 (g) 7 (h) 8 (i) 9 (j) 10

Figure 4.6: *Distribution of* 25 *hidden variables for each gesture. The numbers in each pie represent the hidden state, and the area enclosed represents the proportion*

Frame rate	Accuracy	Macro F	Micro F
30	$62.77 \pm 20.32\%$	0.5654	0.6277
25	$62.77 \pm 20.32\%$	0.5673	0.6277
15	$62.77 \pm 20.32\%$	0.5654	0.6277
10	$59.99 \pm 20.97\%$	0.5406	0.5999

Table 4.4: *HCRFs classification results of sequences with different frame rates*

4.3.4 Frame classification

The above evaluations are all measured sequence by sequence. In order to give a frame-by-frame prediction, a time-shift window is applied in the HCRFs model. The time-shift window segments a short sequence around current frame and the testing result of this short sequence is regarded as the gesture label of current frame. Taking the HCRFs model with 25 hidden variables and 7 frames in a context window, various sizes of the time-shift window are evaluated as listed in table 4.5. These results are also plotted in figure 4.7. From figure 4.7 it can be seen that the over all recognition performance measured in frames is better than the performance measured in sequences. This is probably because: (a) the prediction error is diluted by a large number of true positive frames in the frame-based classification; (b) long test sequences may encounter error propagation problem while the time-shift window technique sets up a new testing sequence for every segment.

Time-shift window size	Accuracy	Macro F	Micro F
5	$70.56 \pm 9.82\%$	0.7259	0.7056
9	$76.98 \pm 9.84\%$	0.7813	0.7698
13	$80.22 \pm 9.73\%$	0.8130	0.8022
17	$82.33 \pm 9.52\%$	0.8346	0.8233
21	$83.89 \pm 9.76\%$	0.8495	0.8389
25	$84.45 \pm 10.06\%$	0.8544	0.8445

Table 4.5: *Classification results with different length of test sequence in HCRFs*

It can also be seen that the classification accuracy improves significantly along with the size increase of the time-shift window. This is probably because that the longer the test sequence is, the more sequence history is encoded in the model for prediction. This phenomena coincides with how human beings observe a gesture. The longer gesture we watch, the higher confidence we have about the class of that gesture. The confusion matrices in figure 4.8(a) and 4.8(b) further demonstrate this phenomena. However, large time-shift window does suffer high latency in frame prediction, which is a drawback for real-time applications.

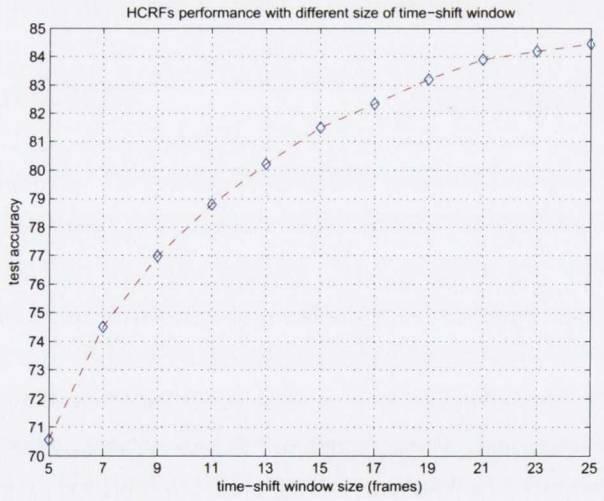
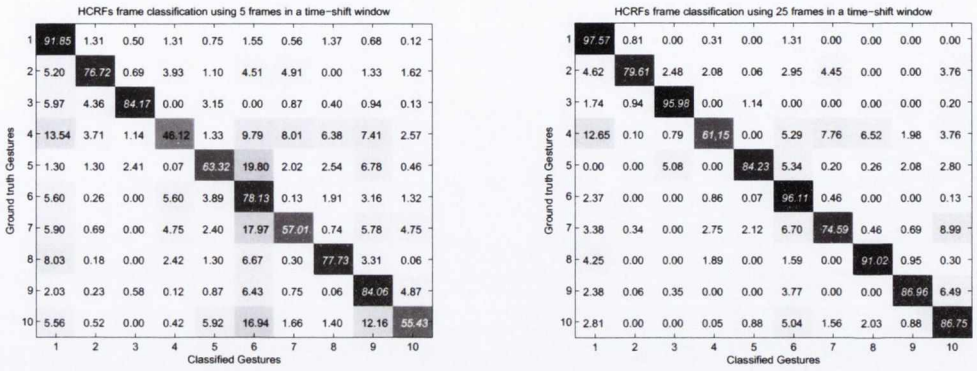


Figure 4.7: *HCRFs frame prediction accuracy with different length of test sequences*



(a) (b)

Figure 4.8: *Frame classification confusion matrices using 7 frames in a context window, 25 hidden variables and* (a) *5 frames in a time-shift window;* (b) *25 frames in a time-shift window*

4.4 Summary

In this chapter, we model the hand washing gestures as linear chains with sequence labelling tool HCRFs. The static information of the gestures are captured by the "2D" dense HOG features while the dynamic information are modelled with the first-order Markov process. Since the parameter estimation via maximising the log-likelihood function is very time consuming for our large training set, we perform the model training by adapting Wang's max-margin training method. Several key parameters, such as the context window size, hidden variable number and the test sequence length for frame-based output, are evaluated in terms of the classification results.

We first evaluate the effect of the context window size on the recognition performance. Same as Wang's finding [139], adding contextual information to each frame can improve the recognition performance. However, we also notice that the overall improvement by using long context window is not substantial. We believe that this is because the information for discriminating gestures has been mostly included in the single frame static HOG features. Simply increasing the context window size in HCRFs would not improve performance significantly if primitive features from each node in a sequence are adequately discriminative.

We then evaluate different number of hidden variables in the HCRFs model. Our experiments show that once the number of the hidden variables meets the minimum

requirement, the extra hidden variables do not increase the performance significantly. We suspect that this is because the HCRFs model is dominated by some of the hidden variables even though more hidden variables are available.

Our HCRFs model is also tested with videos in different frame rates. Experiment results show that the HCRFs model plays equally well on our test videos which have different temporal resolution.

The HCRFs model naturally reports the classification results in terms of sequences. In order to give the frame by frame classification, we apply a time-shift window during the testing. Our experiment results show that the recognition performance improves significantly along with the increase of the shift-window size. This phenomena coincides with humans' behaviour when we observe the gestures. The confidence about the gesture class is built up when we continuously watch the gestures.

It can be seen that the recognition using sequence labelling tool has better performance than the recognition using static postures when the accuracy is measured in terms of frames. This is reasonable since both the dynamic information and static information are used for classification in recognition using sequence labelling method.

Chapter 5

Recognition using Space-Time Interest Points

5.1 Introduction

Both chapter 3 and chapter 4 recognize hand washing gestures based on the global description of hand shapes. The hand shape configuration in each frame is well encoded in the dense grid HOG features. However, this dense grid encoding has limited capacity of handling the big spatial variety in hand washing gestures, and strongly depends on the ROI detection results. Thus, it would be interesting to evaluate the sparse representation approaches for the hand washing gesture recognition. In this chapter, we therefore attempt to recognize the hand washing gestures via space-time interest points. The motivation of the method is that the hand washing gestures can be discerned with a rich set of local features, regardless of global appearance and motion.

Recognition using space-time interest points does not require ROI detection in the image preprocessing, thus the lighting condition would have little effect on the recognition performance. Different with the recognition using sequence labelling method which encodes the dynamic information of gestures as a sequence of intermediate states, recognition using space-time interest points method encapsulates the dynamic information into local interest point descriptors.

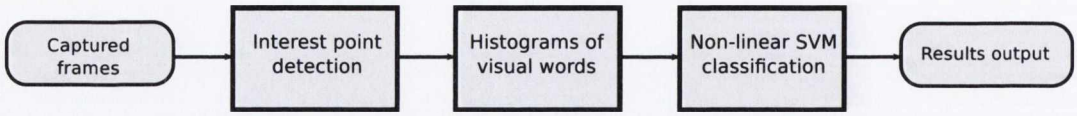


Figure 5.1: *Workflow of recognition using space-time interest points*

5.2 Methodology

Figure 5.1 illustrates the workflow of recognition using space-time interest points method. The first step is the space-time interest point detection. In our evaluation, the interest points are detected by Dollar's cuboid detector [46], and are described by the spatial-temporal "3D" HOG features [7]. In the second step, an extension of the BOFs approach [57][176], ERC Forests [52], is employed to build the descriptors of hand washing gestures. The ERC-Forests method provides a rapid and highly discriminative alternative of k-means for building the visual vocabulary and occurrence histograms of visual words. For classification, the non-linear SVM with a χ^2 kernel is applied, which has been widely used in object recognition and action recognition [109][65][178].

5.2.1 Interest point detection

Hand washing gestures are continuous hand movements, which present both static and dynamic characteristics. We therefore perform our interest point detection in the spatial-temporal domain to accentuate the hand washing gestures in the videos. Every detected point is a short, local video patch, and is described by spatial-temporal "3D" HOG which wrap the local shape and motion information into a vector.

Cuboid detector

A large number of sparse representation recognition approaches has been proposed to detect and leverage the use of spare, informative feature points. However, in some domains, some methods such as Harris3D [109] may give very sparse detection or non-informative detected points [46]. On the contrary, our interest point detection is based on Dollar's cuboid detector [46] which is designed to detect too many points rather than too few, and suppress irrelevant or misleading points generated from scene clutter.

Cuboid detector operates on a stack of frames. These frames are convoluted with spatial filters and temporal filters. Space-time interest points are detected by searching

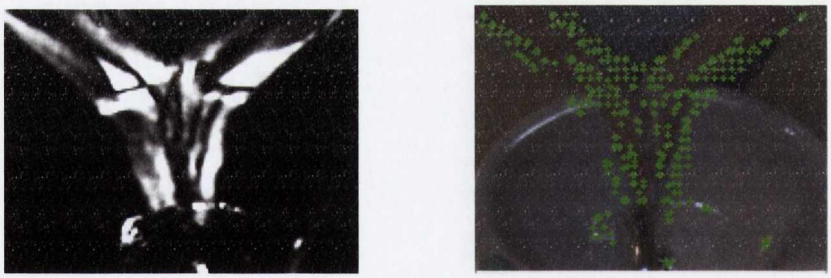
the local maxima of filter response. The response function R is defined as equation 5.1,

$$R = (I(x, y, t) * g(x, y; \sigma) * h_{ev}(t; \tau, \omega))^2 + (I(x, y, t) * g(x, y; \sigma) * h_{od}(t; \tau, \omega))^2 \quad (5.1)$$

where $I(x, y, t)$ represents the input image stack, $g(x, y; \sigma)$ denotes the spatial 2D Gaussian smoothing kernel, and h_{ev} and h_{od} are a quadrature pair of 1D Gabor filters which are defined as formulae 5.2.

$$h_{ev}(t; \tau, \omega) = -cos(2\pi t\omega)e^{-t^2/\tau^2} \qquad h_{od}(t; \tau, \omega) = -sin(2\pi t\omega)e^{-t^2/\tau^2} \quad (5.2)$$

The parameter ω is set at $4/\tau$ as suggested by Dollar [46]. This effectively gives the response function R two parameters σ and τ which correspond roughly to the spatial and temporal scale of the detector. The Gabor filters only operate temporally, thus the output of the Gabor filters would represent the motion energy of hand washing gestures. Regions with spatially distinguishing characteristics undergoing a complex motion would induce high energy output, but areas without spatially distinguishing features cannot induce a response. Moreover, pure translation motion will in general not induce a response from R as well [46]. Figure 5.2 shows an example of the detection result from cuboid detector.



(a) (b)

Figure 5.2: *Spatio-temporal interest points*

Spatial-temporal HOGs

All detected space-time interest points are described by the spatial-temporal "3D" HOG. "3D" HOG descriptor is a generalization of Dalal's 2D HOG. It encodes the shape and motion information at the same time. Given a detected interest point $s = (x_s, y_s, t_s, \sigma_s, \tau_s)$ where σ_s is the spatial scale and τ_s is the temporal scale, a small

cube with the detected point in the centre is sampled. The width w_s, height h_s and length l_s of the cube are given as formula 5.3,

$$w_s = h_s = \sigma_0 \sigma_s, \qquad l_s = \tau_0 \tau_s \tag{5.3}$$

where parameters σ_0 and τ_0 give additional control of the sampled region size around point s. Next, the sampled cube is divided into a set of $M \times M \times N$ cells which are aligned next to each other. Each cell is subsequently divided into $S \times S \times S$ subblocks as illustrated in figure 5.3 (b) [7]. In every subblock, each gradient orientation is quantized with regular polyhedrons. The regular polyhedrons can be tetrahedron (4-sided), cube (6-sided), octahedron (8-sided), dodecahedron (12-sided), and icosahedron (20-sided). The quantization of a 3D gradient vector is operated by projecting it to the axes running through the gravity centre of the polyhedron and the centre positions of all faces. The quantized gradient vector is then normalized to obtain a quantized mean gradient q_{b_i}. A cell histogram is calculated by summing the quantized mean gradients q_{b_i} of all subblocks b_i as formula 5.4,

$$h_c = \sum_{i=1}^{S^3} q_{b_i} \tag{5.4}$$

All cell histograms are then concatenated and $L2$ normalized to form the final "3D" HOG feature vectors. Figure 5.3 [7] illustrates the entire encoding procedure described above. Note that the gradients in subblocks are computed based on integral videos which enables the memory-efficient feature computation at arbitrary spatial and temporal scales [7].

5.2.2 Histograms of visual words

All "3D" HOG descriptors of detected interest points are pooled together and clustered to build a visual vocabulary. The visual vocabulary is conventionally built by the K-means clustering [65][79]. However, the K-means clustering methods are generally very computationally expensive for large training sets. Thus, in our method, the ERC-Forests method is employed. The ERC-Forests method is ensembles of randomly created clustering trees. Each tree is trained as classifiers but used as descriptor-space quantization rules [52]. The ERC-Forests method provides a rapid and highly discriminative approach for building a visual vocabulary, and has good resistance to background clutter.

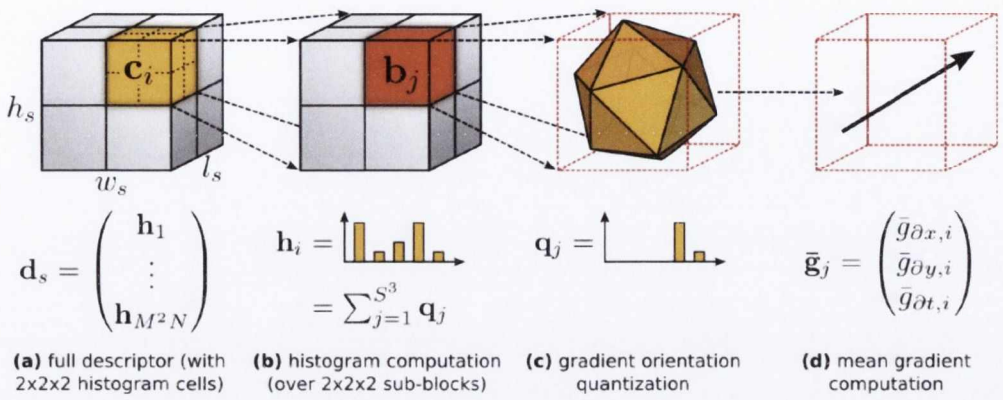


Figure 5.3: *Overview of the descriptor computation: (a) Sampled cube is divided into a grid of cells; (b) each cell is computed over a grid of subblocks; (c) each gradient orientation is quantized using regular polyhedrons; (d) each mean gradient is computed using integral videos [7]*

During a query, for each "3D" HOG descriptor, every tree in the forests is traversed from the root down to a leaf through a bunch of nodes as shown in figure 5.4. At each node, the descriptor is tested by a random function $\mathcal{T}$ to decide the descendant. The random function is learned during the training with a small random subset $I' \subseteq I$ of the training set I. An elementary feature f_i indexed by i is randomly picked from the feature vector. The subset I' will be split into left and right subsets I_l and I_r by comparing the selected elementary feature f_i with a random threshold t sampled from a uniform distribution.

$$\begin{aligned} I_l &= \{i \in I' \mid f(i) < t\}, \\ I_r &= I' \setminus I_l \end{aligned} \tag{5.5}$$

The random splitting procedure described in formula 5.5 is repeated many times until a fixed maximum number T_{max} is reached or the expected gain in information is higher than a fixed threshold S_{min}. The expected gain is calculated by equation 5.6,

$$\Delta E = -\frac{|I_l|}{|I'|}E(I_l) - \frac{|I_r|}{|I'|}E(I_r) \tag{5.6}$$

where $E(I)$ is the Shannon entropy [59] of the classes in the set of examples I. The random function with the highest expected gain is retained. All trees in the forests is recursively built by learning the random function for each node as described above until a predefine maximum depth D of the tree is reached.

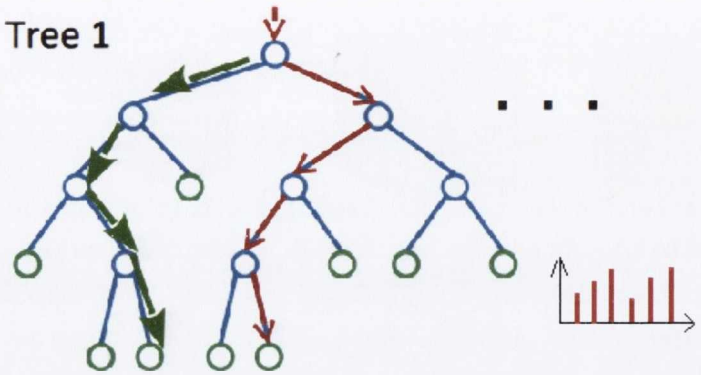


Figure 5.4: *Traversing a tree in ERC-Forests*

By traversing a tree, a unique leaf index is returned as a distinct label. Therefore, as shown in figure 5.5 [52], the leaves can be used as visual words, and ensembles of trees present the visual vocabulary. An occurrence histogram of visual words is thus built by transforming each HOG descriptor into a set of leaf node indices from all trees. Votes for each index are accumulated into a global histogram for a set of "3D" HOG descriptors. In our evaluation, this set of HOG are extracted from a group of spatial-temporal interest points as described in section 5.2.1.

The dimension of the histogram is controlled by the number of trees employed in the forests, which also determines the discriminative power of the visual vocabulary. Some work has shown that the discrimination of the visual words would be saturated when 5 trees are used in ERC [52][147]. Therefore, we explore the categorization ability of the visual vocabulary by controlling the maximum depth D of each tree in our evaluation. We believe that the discriminative power of the visual words can be further enhanced by using trees with deep depth.

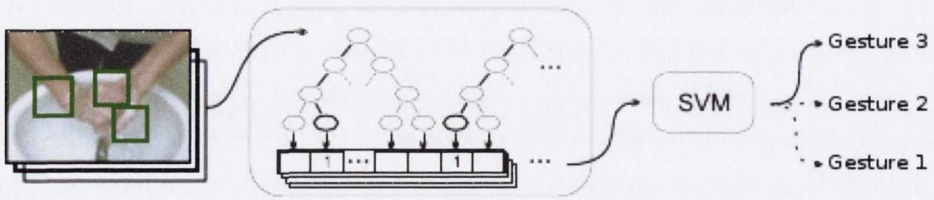


Figure 5.5: *Building visual vocabulary with ERC-Forests*

5.2.3 Non-linear SVM classification

Although binary trees are trained in the previous section, they are only used for clustering rather than classification. In recognition using space-time interest points method, the classification is a separated stage. A multi-class SVM model is built for classification with occurrence histograms of visual words generated from ERC-Forests. In our evaluation, we apply the non-linear SVM with a χ^2 kernel for classification. The χ^2 kernel is best suited for histogram style feature vectors and has shown good performance in object recognition and action recognition [109][65][178]. In our evaluation, the χ^2 kernel is defined as formula 5.7 [7],

$$K(H_i, H_j) = \exp\left(-\frac{1}{A}\sum_{n=1}^{V}\frac{(h_{in} - h_{jn})^2}{h_{in} + h_{jn}}\right) \tag{5.7}$$

where $H_i = \{h_{in}\}$ and $H_j = \{h_{jn}\}$ are the histograms of visual words and V is the histogram dimension. A is a scaling parameter and we set it as $A = V$ empirically.

5.3 Evaluation

Recognition using space-time interest points method is based on the sparse representation of hand washing gestures. Thus, it is not necessary to have a ROI detection in the process. In our evaluation, the interest point detection is operated over the whole image area. However, if the interest point detection is performed for every video frame, there would be a huge number of interest points that the computer can not handle during the training. Therefore, in our evaluation the space-time interest point are detected in every 3 frames during the training in order to obtain a manageable training set, but during the testing phase, the interest points are detected frame by frame.

5.3.1 Interest points detection

We implement Dollar's cuboid detector to detect the space-time interest points in the videos. The detector has two major parameters: the spatial scale σ and the temporal scale τ. The parameter σ is used in the spatial Gaussian filters. According to the scale-space theory [154], it controls the spatial details captured in a video patch. In our evaluation, σ is set at 4, which removes the salt and pepper noise but keeps most of the shape information. τ is the temporal scale of the 1D Gabor filters. It controls the

temporal resolution to be considered. Our experimenting on τ shows that the value of τ is not exclusive to have good detection function response. The value of τ can be a good choice as long as the Gabor filter responses contain positive and negative signs and the response directions are symmetrical. However, larger τ would result in smaller ω value, and need more video frames to give good detection function responses. In our evaluation, we set $\tau = 16$ which requires 5 video frames to compute the detection function response. Figure 5.2 shows an example of the detection result with the given parameter settings. It can be seen that many interest points are detected in the regions around the motion boundaries which usually have strong detection function response.

5.3.2 Video patch descriptor

All detected space-time interest points are described by the "3D" HOG descriptors. Klaser's 3D descriptor computing tool [7] is applied to extract the "3D" HOG features. As described above, each interest point has scales $\sigma = 4$ and $\tau = 16$. By setting the patch size controlling parameters $\sigma_0 = 8$ and $\tau_0 = 0.3125$, we obtain a video patch with size $32 \times 32 \times 5$ according to the equation 5.3. This video patch is further divided into $2 \times 2 \times 2$ cells, and each cell has $4 \times 4 \times 4$ subblocks by default. Each subblock is described by an icosahedron with half sphere quantization. This parameter settings finally give us a $\{(20 \div 2) \times 2 \times 2 \times 2 = 80\}$ dimension "3D" HOG descriptor.

5.3.3 Building visual vocabulary with ERC-Forests

We developed our ERC-Forests based on Yu's random forest library [175]. All the video patch descriptors from the training set are fed into the ERC-Forests to build a visual vocabulary. The vocabulary size usually has an impact on the classification performance. There has been methods attempt to search an optimal vocabulary size by adding decision trees to the forests [52][147]. In these methods, the discriminative power of the vocabulary is almost saturated when 5 trees are employed. In our evaluation, we believe that the depth of trees has more control on the discriminative power of the vocabulary. We employ 5 trees in the forests and evaluate different tree depths for recognition. Table 5.1 lists the size of vocabularies generated by ERC-Forests using different tree depths.

Tree depth	5	6	7	8	9	10
Dimension	160	320	640	1280	2560	5113

Table 5.1: *Vocabulary sizes using different tree depths*

Tree depth	Accuracy	Macro F	Micro F
5	$52.17 \pm 6.50\%$	0.5198	0.5217
6	$57.39 \pm 7.16\%$	0.5763	0.5739
7	$61.09 \pm 7.75\%$	0.6172	0.6109
8	$62.61 \pm 8.27\%$	0.6317	0.6261
9	$64.53 \pm 8.11\%$	0.6535	0.6453
10	$65.83 \pm 7.33\%$	0.6689	0.6583

Table 5.2: *Classification results with different tree depths*

5.3.4 Classification results

The classification is performed by the non-linear SVM with a χ^2 kernel as described in equation 5.7. Our implementation adapted the libsvm [22] by adding a χ^2 kernel inside the program. Table 5.2 lists the classification results with different visual vocabularies generated by ERC-Forests using different tree depths. The results are also plotted in figure 5.6

It can be seen that the classification accuracy improves slowly along the growth of the tree depth in ERC-Forests. This verifies our assumption that the tree depths in ERC-Forests can be used to control the discriminative power of a visual vocabulary. The trees with deeper depth may generate finer visual words which can capture the details in gestures for discrimination. Figure 5.7(a) and figure 5.7(b) also show the classification confusion matrices when the trees in ERC-Forests have depth 5 and 10 respectively. We can see that gesture 1 and 4, and gesture 9 and 10 are mostly misclassified to each other. Moreover, there is also certain amount of misclassification between gesture 4 and 8. This is probably because that recognition via space-time interest points method does not preserve any global spatial configuration of the gestures. Any partial correlation between two sets of interest points can fire a misclassification regardless of these interest point positions. For example, the left spread out hand in gesture 8 may be confused with the left hand in the gesture 4.

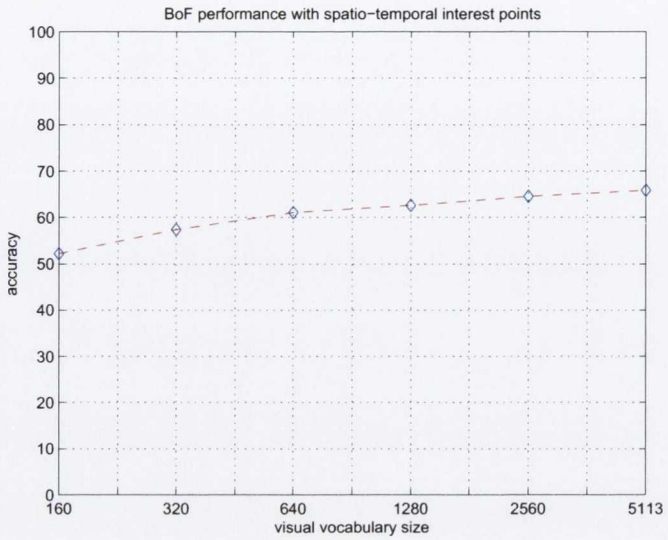
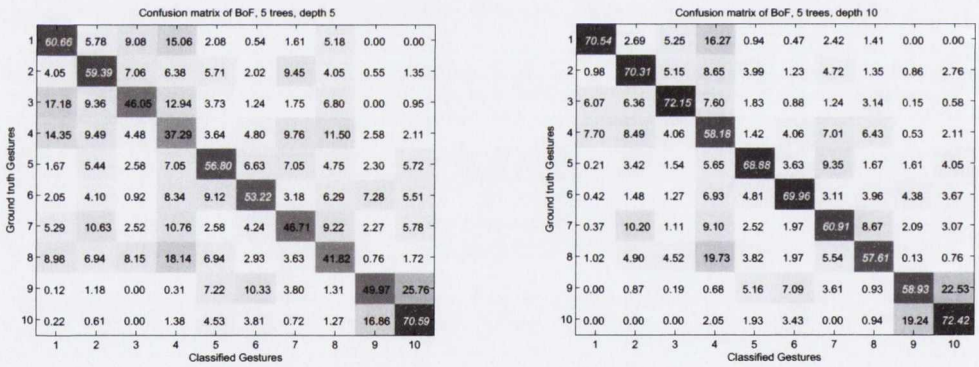


Figure 5.6: *Classification curve with different vocabulary size*



(a) (b)

Figure 5.7: *Confusion matrix of classification:*(a) *tree depth is 5;* (b) *tree depth is 10*

5.4 Summary

In this chapter, we evaluate the method of recognizing hand washing gestures using space-time interest points. The interest points are detected as maxima of spatial-temporal energy map. The detection method is designed to detect too many rather than too few interest points. Our experiments show that many interest points are detected in the regions around the motion boundaries which usually have strong detection function

response.

The detected space-time interest points are described by the "3D" HOG features, and are used to build the visual vocabulary. Our visual vocabulary is built by the ERC-Forests, a fast efficient alternative of K-means clustering. The discriminative power of the visual vocabulary is generally controlled by the number of trees employed in the forests, and usually saturates when 5 trees are used. In our evaluation, we believe that the depth of trees can have better control on the discriminative power of the vocabulary. Our experiments show that the classification accuracy can improve slowly along the growth of the tree depth in ERC-Forest even 5 trees have been used.

Our overall evaluation shows that the hand washing gestures can be recognized with a rich set of local features regardless of the global appearance and motion. Accordingly, ROI detection is not necessary in the recognition. However, because the method does not preserve any global spatial configuration of the gestures, any partial correlation between two sets of interest points can confuse the classifiers regardless of their relative geometry positions in frames.

Chapter 6

Recognition using TGC-HCRFs

6.1 Introduction

In this chapter, we propose TGC-HCRFs method for hand washing gesture recognition. The TGC-HCRFs method unifies texton analysis and HCRFs method in a same framework such that the recognition can be beneficial from both HCRFs and texton analysis. The basic idea of the method is that the gesture textons can be modelled by the hidden states in HCRFs. Our experiments show that the HCRFs model can automatically determine the visual vocabulary size for recognition.

In TGC-HCRFs, the local spatial-temporal gesture information is captured by gesture textons via the "3D" HOG descriptors as described in chapter 5. In chapter 5, the "3D" HOG descriptors are simply clustered to visual words. No global spatial configuration of the gestures is conserved. Instead, in TGC-HCRFs the global spatial structure is modelled by the HCRFs with a grid graph. Comparing to the linear-chain HCRFs which has been studied in chapter 4, the grid graph structure used in TGC-HCRFs method offers a more flexible way to include the contextual and neighbouring information for recognition. Moreover, the local features used in TGC-HCRFs can be extracted simultaneously such that the TGC-HCRFs method can easily be parallelised, which makes the method a good candidate for real-time applications.

6.2 Methodology

Figure 6.1 illustrates the workflow of the TGC-HCRFs method. Given captured images, hand regions are detected first as described in chapter 3 to eliminate noisy background,

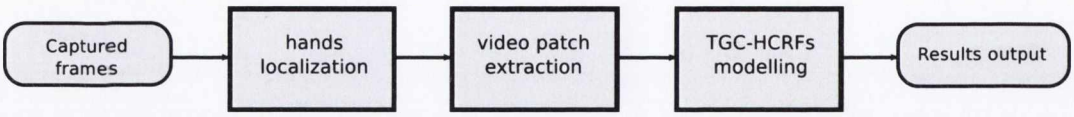


Figure 6.1: *Recognition using TGC-HCRFs workflow*

however drawing bounding boxes of hands is an option in TGC-HCRFs method as general graph structures are acceptable in TGC-HCRFs. Next, the detected hand regions are split in grid, and we call a small spatial-temporal cube a video patch as illustrated in figure 6.2. These patches are regarded as nodes on the grid graph and are modelled as textons by the hidden states in HCRFs.

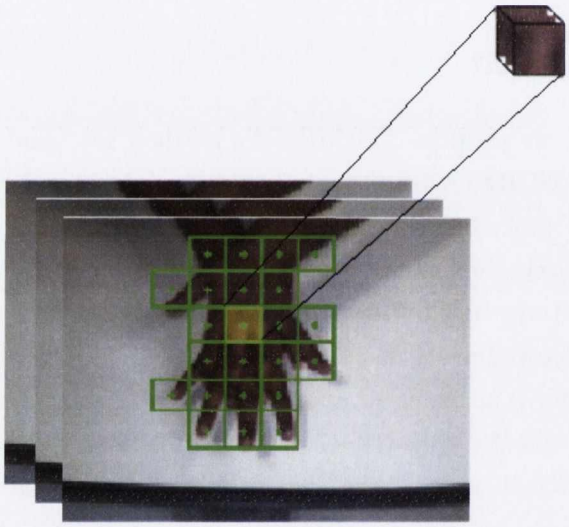


Figure 6.2: *An example of video patch in TGC-HCRFs method*

Although HCRFs are applied in both TGC-HCRFs method and the sequence labelling method in chapter 4, they are used in very different ways. In recognition using sequence labelling method, the HCRFs is modelled as a linear-chain, and each node in the chain is a video frame. The output of the linear-chain HCRFs model is a class label for the whole video sequence. In TGC-HCRFs, a grid graph structure is used to model the spatial configuration of hands as illustrated in figure 6.3. Each node of the graph, which is a small video patch extracted in the spatial-temporal space, is assigned with a hidden state. The ensemble of all node states then represent the graph class. Since the grid graph is constructed for every video frame, the TGC-HCRFs

method can report the classification result frame by frame naturally. In the rest of this section, we are going to elaborate the details of the TGC-HCRFs model.

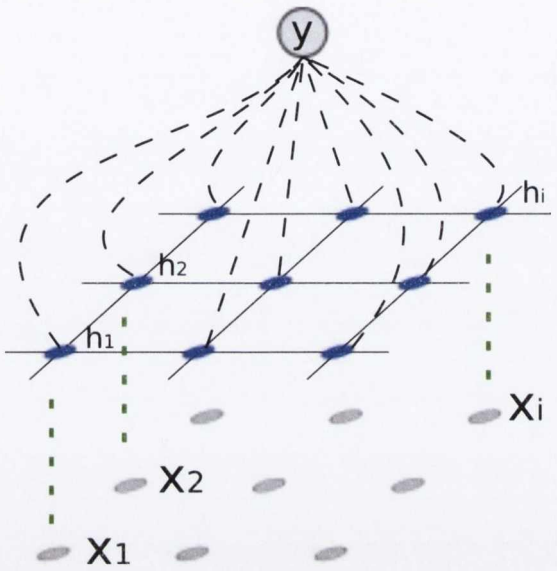


Figure 6.3: *The structure of the TGC-HCRFs model*

6.2.1 Video patch sampling

In TGC-HCRFs, a video patch is a small spatial-temporal cube extracted from a stack of frames as shown in figure 6.4, and each video patch is described by a spatial-temporal "3D" HOG descriptor. However, in TGC-HCRFs the video patches are not detected by searching local maxima of a spatial-temporal filter response as chapter 5. Instead, they are extracted regularly in grid within a hand region or a ROI.

As shown in figure 6.5(a), given a predefined grid size, the video patches are sampled over the ROI in grid. The green points in figure 6.5(a) indicate the centres of sampled video patches, and the spacing between adjacent points defines the grid size. If the hand region is used in the video patch sampling as shown in figure 6.5(b), the video patches fall outsize the hand region will be dropped. Consequently, the graph structure used in TGC-HCRFs would then be a general grid graph.

The extracted video patches have a predefined size $W_p \times H_p \times L_p$ where W_p denotes the width of the patch and H_p denotes the height. L_p is the length of the patch, which indicates how many video frames are used for the feature extraction. Each patch cube

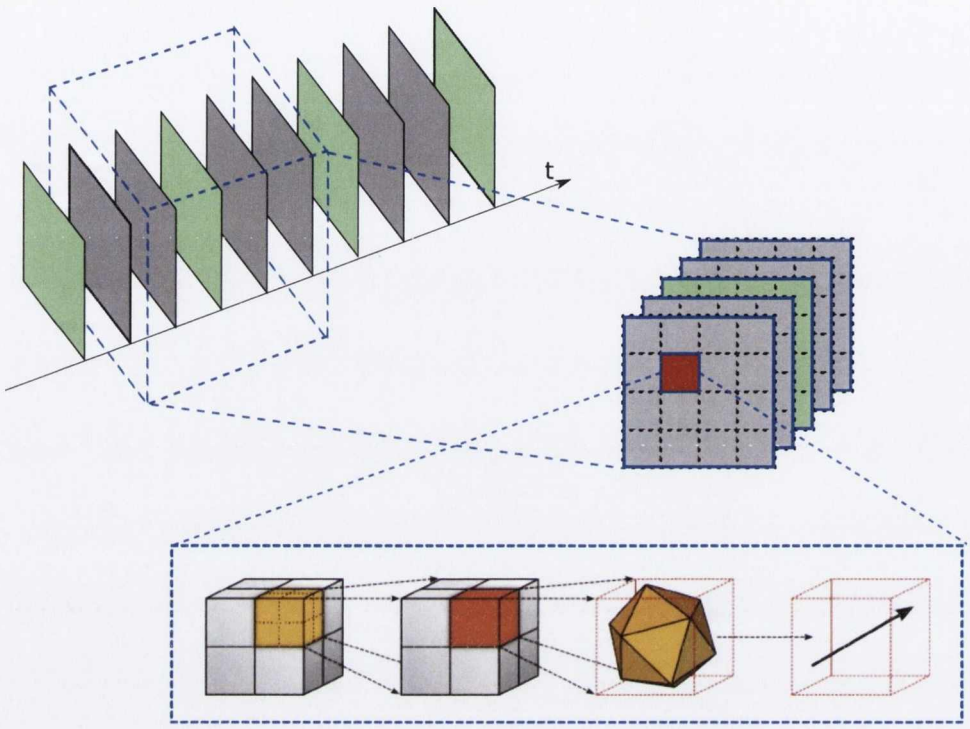


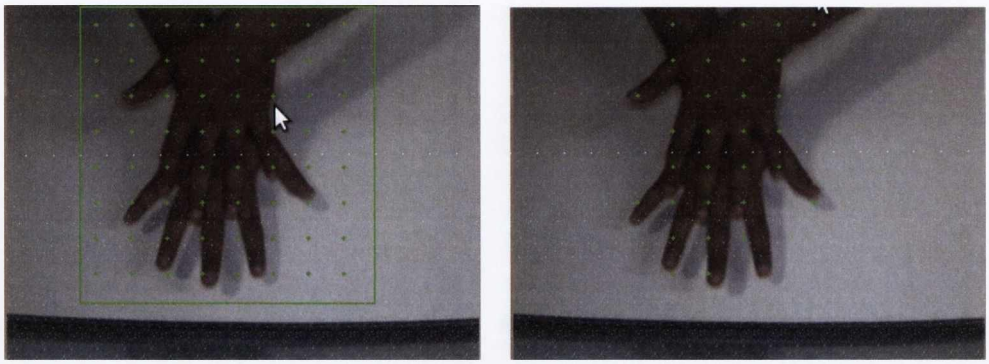
Figure 6.4: *A video patch generation in TGC-HCRFs*

is described by the "3D" HOG features. Moreover, different patch sizes can also be used in the feature extraction. This would result in a multiple scale description of a patch.

The grid video patch sampling operates similarly to the cells used in HOGs calculation, however in TGC-HCRFs, there is no block normalization and the local features extracted from video patches are directly applied in the TGC-HCRFs model.

6.2.2 TGC-HCRFs model

Our TGC-HCRFs model is defined similarly to Kumar's Discriminative Random Fields model [99]. Let $X = \{x_i\}, i \in \mathcal{S}$ be the extracted video patches at current frame and $\{x_j\}, j \in \mathcal{N}_i$ be the neighbours of video patch x_i, where $\mathcal{S}$ denotes the patch set and $\mathcal{N}_i$ denotes the neighbour set. We define the potential function used in TGC-HCRFs



(a) (b)

Figure 6.5: *Video patch sampling in TGC-HCRFs.* (a) *sampling in a ROI;* (b) *sampling in hand region*

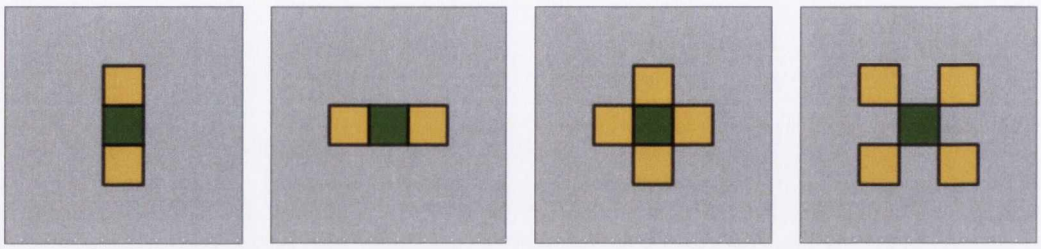
model as formulae 6.1 and 6.2.

$$\Psi(y, h, X; \lambda, w) = \exp\left(\Phi(y, h, X; \lambda, w)\right) = \exp\left(\lambda \cdot f(y, h, X; w)\right) \tag{6.1}$$

$$\lambda \cdot f(y, h, X; w) = \sum_{i \in \mathcal{S}} \lambda_1 \cdot f_1(h_i, x_i; w) + \sum_{i \in \mathcal{S}} \lambda_2 \cdot \mathbb{1}(y, h_i) + \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}_i} \lambda_3 \cdot \mathbb{1}(y, h_i, h_j) \tag{6.2}$$

As shown in equation 6.2, there are three types of features used in TGC-HCRFs. The unary node feature $f_1(h_i, x_i; w)$ with its weight λ_1 models the compatibility between a video patch x_i and a hidden state h_i. In our evaluation, the "3D" HOG features are used for this type of feature. Analogously, another unary feature $\mathbb{1}(y, h_i)$ with its weight λ_2 measures the compatibility between a class label y and a hidden state h_i. The inner-product $\lambda_2 \cdot \mathbb{1}(y, h_i)$ tells how likely current frame with a class label y contains a video patch with a hidden state h_i. The pairwise edge feature $\mathbb{1}(y, h_i, h_j)$ with its weight λ_3 models the compatibility between a class label y and a pair of hidden states (h_i, h_j). This tells that how likely a pair of hidden states (h_i, h_j) would present when current frame belongs to class y.

From the potential function definition, it can be seen that the context window w and neighbours $\mathcal{N}_i$ are utilized in the TGC-HCRFs model. Since a grid graph is used, in TGC-HCRFs the contextual information and neighbours would be included differently to the sequence labelling method in chapter 4. In the following sections we would roll out the details.



(a) (b) (c) (d)

Figure 6.6: *Spatial neighbouring system in TGC-HCRFs:* (a) *2-way neighbours on the vertical;* (b) *2-way neighbours on the horizontal;* (c) *4-way neighbours on the cross;* (d) *4-way neighbours on the diagonal*

6.2.3 Neighbours

The linear-chain HCRFs only consider the forward and backward relationship between two adjacent nodes in a chain. On the contrary, The grid graph structure in TGC-HCRFs enables a more flexible way to model the relationship between neighbouring nodes. For a video patch, the neighbouring system $\mathcal{N}_i$ can have 2 nodes, 4 nodes or even 8 nodes as its neighbourhoods, which correspondingly forms a 2-way, 4-way or 8-way neighbouring system. A 2-way neighbouring system takes 2 adjacent nodes as neighbourhoods: the top node and the bottom node, as illustrated in figure 6.6(a), or the left node and the right node as illustrated in figure 6.6(b). The 4-way neighbouring system includes all 4 nodes as neighbourhoods: the top, bottom, left and right nodes as shown in figure 6.6(c), or the top-left, top-right, bottom-left and bottom-right nodes as shown in figure 6.6(d). For a 8-way neighbouring system, all 8 neighbourhoods mentioned in the 4-way neighbouring system are used. The neighbouring system can even have a 10-way neighbours as shown in figure 6.7(a), two nodes along the time axis are added, which are the one node in the previous frame and one node in the after frame. Moreover, it is also possible to have a skip-neighbouring system as shown in figures 6.7(b) and 6.7(c), where an irrelevant node lies between a neighbourhood and the centre node. Since we mainly concern the spatial configuration of hands and the temporal dynamic information has been encapsulated in the "3D" HOG descriptors, in our evaluation we only focus on the spatial neighbouring systems for recognition. No temporal neighbours are considered.

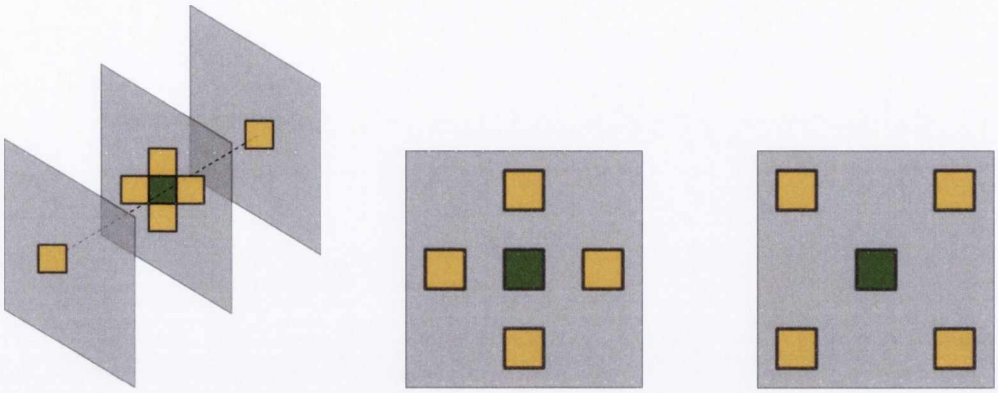


Figure 6.7: *Other possible neighbouring systems in TGC-HCRFs:* (a) *spatial-temporal neighbours;* (b) *4-way skip neighbours on the cross;* (c) *4-way skip neighbours on the diagonal*

6.2.4 Context window

From chapter 4, it has been known that contextual information extracted from a context window can help the state inference. In TGC-HCRFs the context window w is defined very similar to the neighbouring system. What is different is that all the features extracted from surrounding nodes are concatenated with current node feature to form a long feature vector as the descriptor of current node. It has to be aware that the dimension of the descriptor can increased drastically when many nodes from different directions are included in a context window.

6.2.5 Hidden states

In TGC-HCRFs each video patch is modelled by a hidden variable, and each hidden variable is regarded as a gesture texton. The hidden variable set then becomes the visual vocabulary. Generally, choosing a proper vocabulary size is not a easy job. Small size vocabulary may lack of discrimination while a large vocabulary may result in overfitting. Our experiments show that by combining the graph cut and HCRFs method, the TGC-HCRFs model has the ability of automatically determining the size of a visual vocabulary for recognition during the training.

6.2.6 Model training

The training of the TGC-HCRFs is very similar to the training algorithm described in chapter 4. However, the learning procedure of the sequence labelling method in chapter 4 can be regarded as an online training algorithm where recent training samples have more effect on the model performance. In sparse representation based approaches, the local descriptors per se generally are not as discriminative as dense global descriptors for recognition. Thus, the online training may not be a good choice for our TGC-HCRFs model. We develop our training algorithm of TGC-HCRFs model based on the cutting-plane training algorithm of structural SVM [83] which conserves all the training constraints during the learning procedure.

The training algorithm of TGC-HCRFs is also an EM-based training method within the max-margin training framework. There are two iterative steps in the training. In the first step, the hidden structure is inferred with fixed model parameters. In the second step, the hidden structure is fixed. The model parameters are optimized by a max-margin based training algorithm. In chapter 4, for a linear-chain HCRFs model, the hidden structure of a sample can be inferred by the viterbi algorithm [105]. However, for a general graph structure, exact inference would become computationally intractable and an approximate inference has to be applied. In our training algorithm, we use the graph cuts method for this approximation since empirically graph cut usually runs faster than the message passing methods.

Joachims' cutting-plane training method was proposed as a fast and tractable learning method of structural SVM on large datasets. It contains two training algorithms: margin-rescaling training and slack-rescaling training. Neither of them has hidden variables involved. In TGC-HCRFs training, we adapt the margin-rescaling training algorithm for the parameter learning with hidden variables.

Same as chapter 4, the training of TGC-HCRFs is performed in the dual form of the optimization, which tries to find optimal dual variables that maximize the dual objective $\mathcal{L}_D$ 6.3:

$$\begin{aligned} &\max_{\alpha} \sum_{l=1}^{n} \sum_{y} \alpha_{l,y} \delta(y^{(l)}, y) - \frac{1}{2} \| \sum_{l=1}^{n} \sum_{y} \alpha_{l,y} \varphi(y, X^{(l)}) \|^2 \\ &s.t. \quad \sum_{y} \alpha_{l,y} = C, \quad \alpha_{l,y} \geq 0, \quad \forall y, \ \forall l \end{aligned} \tag{6.3}$$

where $\alpha_{l,y} \in \boldsymbol{\alpha}$ are the dual variables and vector δ represent the binary loss of the model. The feature difference map $\varphi(y, X^{(l)})$ is calculated with the following equation

6.4 for all possible labels y:

$$\varphi(y, X^{(l)}) = f(y^{(l)}, h^{(l)}, X^{(l)}; \lambda, \omega) - f(y, h, X^{(l)}; \lambda, \omega) \tag{6.4}$$

Since the same dual objective function needs to be optimized, TGC-HCRFs shares most training steps described in chapter 4. Figure 6.8 outlines the learning procedure of TGC-HCRFs.

I. Input:

   a) give training samples $\{y^{(l)}, X^{(l)}\}$, context window size w, and hidden variable number $|h|$

   b) initialize: $\lambda \leftarrow 0$, $\mathcal{R} \leftarrow \emptyset$, where $\mathcal{R}$ is the working set of constraints

II. Repeat until a maximum iteration number is reached

   a) $\varphi_{sum} \leftarrow 0, \quad \Delta_{sum} \leftarrow 0$

   b) Loop over all training samples

      1. holding the parameter λ fixed, estimate the hidden structure h for the current training sample using graph cuts
      2. calculate $\varphi(y, X^{(l)})$ for all possible labels y
      3. $\hat{y} \leftarrow \arg\max_y \{\Delta(y^{(l)}, y) + \lambda \cdot \varphi(y, X^{(l)})\}$
      4. $\varphi_{sum} = \varphi_{sum} + \varphi(\hat{y}, X^{(l)}), \quad \Delta_{sum} = \Delta_{sum} + \delta(y^{(l)}, \hat{y})$

   c) $\mathcal{R} \leftarrow \mathcal{R} \bigcup \{\varphi_{sum}/n,\ \Delta_{sum}/n\}$

   d) $\boldsymbol{\alpha} = \arg\max_{\alpha} \mathcal{L}_D \quad s.t. \quad \sum \boldsymbol{\alpha} = C$

   e) $\lambda = \boldsymbol{\alpha}^{\intercal} \boldsymbol{\varphi}_{sum}$, where $\boldsymbol{\varphi}_{sum}$ denotes all the φ_{sum} included in the working set $\mathcal{R}$

Figure 6.8: *Outline of TGC-HCRFs training algorithm*

The training of TGC-HCRFs starts with setting the parameter λ at 0 and setting the working set of constraints $\mathcal{R}$ empty. It is optional to initialize each training sample with random hidden states. Our experiments show that this does not make much difference in performance. In each training iteration, all the training samples are processed through an EM training procedure, and a cutting plane constraint is computed and added into the working set of constraints $\mathcal{R}$. This cutting plane constraint is calculated with the 1-slack margin-rescaling cutting plane algorithm which assumes that all training samples share only one single slack variable.

The 1-slack margin-rescaling cutting plane algorithm firstly holds the model parameter λ fixed and infers the hidden structure of each training sample for every possible class label. This will generate a feature difference map $\varphi(y, X^{(l)})$ for every training sample. With $\varphi(y, X^{(l)})$, the most violating constraint for each training sample can then be found. During the training, a margin is enforced between the true label and any other labels. However, due to the unoptimized parameters, this margin requirement is usually violated by incorrect expected labels. Thus, the one breaks the margin requirement most would be identified as the most violating constraint as illustrated in above algorithm II.b.3. If this violating constraint is indeed not the true label, a binary loss with value 1 will be inducted. In other words, if it turns out the margin requirement of current training sample is satisfied, the loss would be zero.

Since a violating constraint is identified for each training sample, the cutting plane algorithm would end with n constraints after one iteration, and the working set $\mathcal{R}$ would easily grow to be very heavy for a large training set after several iterations. Joachims shows that it is possible to combine the n constraints to be one constraint for training [83]. This single one constraint can equally form a cutting plane that cuts off the current solution from the feasible set.

After adding that new constraint into the working set of constraints $\mathcal{R}$, the corresponding dual variables are computed by optimizing the quadratic dual objective function $\mathcal{L}_D$ 6.3. In our program, this is done by an efficient QP solver CPLEX [1] . Consequently, the model parameter λ can be updated via the step II.e in the outlined training algorithm, and one training iteration finishes. The whole training algorithm iteratively processes the training set as described above until a prefixed iteration number is reached.

Graph cuts inference

During the TGC-HCRFs training described above, it is required to infer the hidden structure h for each training sample. Since the TGC-HCRFs model has general graph structure involved, exact inference such as the viterbi algorithm would become computationally intractable. Therefore, we apply the graph cuts method for the approximate inference. Specifically, we use the QPBO method [95] to infer the hidden structures.

Graph cuts has been widely used in computer vision for image segmentation, image restoration, stereo matching, etc. It formulates the problems in terms of energy

[1]The IBM ILOG CPLEX Optimizer

minimization, and under most formulations, the minimum energy solution corresponds to the MAP estimate of a solution [111]. Recall our potential function description in section 2.2.1. If the exponentials are used, the potential functions can be rewritten as

$$\Psi_C(\mathcal{V}_C) = \exp(-E(\mathcal{V}_C))$$

where $E(\mathcal{V}_C)$ is the energy function. In TGC-HCRFs model, this energy function would be $E(y, h, X; \lambda)$. A MAP estimate of the hidden structure for a given class label y is found as $\hat{h} = \arg\max_h P(h|y, X, \lambda)$, and the posteriori is computed as $P(h|y, X, \lambda) = \frac{1}{\mathcal{Z}} \exp(-E(y, h, X; \lambda))$ where $\mathcal{Z}$ is a normalization constant. Therefore, the hidden structure can also be found by solving an energy minimization problem as $\hat{h} = \arg\min_h E(y, h, X; \lambda)$. In TGC-HCRFs, we calculate the model energy as $E(y, h, X; \lambda) = -\Phi(y, h, X; \lambda)$ which simply takes a negative result from the inner expression of the potential function, then apply graph cuts on this energy to obtain the hidden state estimates.

The graph cuts method is applied as an intermediate step during our model parameter estimation procedure. There has been a number of methods also applying graph cuts for parameter estimations [19][118][97]. However, these methods have to work with submodular energy functions such that the functions can be minimized by graph cuts in polynomial time [93]. The Submodular functions are discrete analogues of convex functions [14], and have to hold the condition 6.5 for all labels [96][95],

$$E_{pq}(0, 0) + E_{pq}(1, 1) \leq E_{pq}(0, 1) + E_{pq}(1, 0) \tag{6.5}$$

where $E_{pq}(\cdot, \cdot)$ is the pairwise energy term in the potential functions.

In some situations the functions can not be submodular, for example, the energy function with parameters learned from training data. In these situations, the truncation technique can be applied, which ignores the non-submodular terms during the parameter learning. However, this technique may not be appropriate when the number of non-submodular terms is very high [95]. From our experience, when the parameter dimension is high, the updates of the parameters are small in each learning iteration, while the truncation technique can easily miss these updates and make the training hard to converge.

In TGC-HCRFs training we apply the QPBO graph cuts method which explicitly takes into account the non-submodular terms. The QPBO method can cope with the non-submodular terms by constructing a graph with double the number of vertices, and

"flip" a subset of vertices such that all terms become submodular [95]. Since no terms are truncated, the QPBO method would give better parameter estimation results.

Bootstrapping

Since hidden variables are used in TGC-HCRFs model, TGC-HCRFs can not guarantee the training converges to a global maxima. In fact, the EM training method usually gets stuck in a local maxima which may be still far away from the global maxima. In order to alleviate this problem, we apply a bootstrapping strategy during TGC-HCRFs training. Bootstrapping is a resampling method with replacement, which is usually used for estimating the sampling distribution of an estimator. It assumes that the distribution of the samples is a close approximation to the population distribution. The technique is usually employed with the EM training to help avoid getting trapped in local minima.

In TGC-HCRFs training, we randomly downsample the training set with replacement to build a subset for generating a cutting plane in each iteration. In our evaluation, the subset size is chosen to be one twentieth of the original training set. Since a very large training set is employed in our evaluation, the hidden structure inference was a bottleneck of the total training time. However, as a side-effect of the bootstrapping strategy, the training is drastically sped up since in each training iteration only a portion of the original data set needs the hidden structure inference.

Shrinking

After a number of iterations, the working set of the constraints would grow big and slow down the optimization. However, since the feasible training region is iteratively cut, a lot of dual variables would become zeros after a number of optimization, and the working set $\mathcal{R}$ becomes very sparse. In order to save the training time, it is better to remove these constraints with α value 0. On the other hand, we do not want to erase recent constraints as these constraints may temporarily have α 0 but may become non-zeros after the parameters being updated. Therefore, we set a buffer to the dual variables. In our evaluation, this buffer size is 10, which means we only remove the constraints with α 0 which have been optimized at least 10 times.

To sum up the techniques described above, we update the step II in TGC-HCRFs training algorithm in figure 6.9:

II. Repeat until a maximum iteration number is reached

a) $\varphi_{sum} \leftarrow 0, \quad \Delta_{sum} \leftarrow 0$

b) build a subset from 1/20 of the original training set by randomly picking

c) Loop over the subset

1. infer $\hat{h}$ for all possible labels y

$$\hat{h} = \arg\min_{h} E(y, h, X^{(l)}; \lambda)$$

2. calculate $\varphi(y, X^{(l)}) = f(y^{(l)}, h^{(l)}, X^{(l)}) - f(y, \hat{h}, X^{(l)})$
3. $\hat{y} \leftarrow \arg\max_y \{\Delta(y^{(l)}, y) + \lambda \cdot \varphi(y, X^{(l)})\}$
4. $\varphi_{sum} = \varphi_{sum} + \varphi(\hat{y}, X^{(l)}) \quad \Delta_{sum} = \Delta_{sum} + \delta(y^{(l)}, \hat{y})$

d) push $\{\varphi_{sum}/n, \ \Delta_{sum}/n\}$ in a buffer $\mathcal{B}$

e) $\mathcal{R} \leftarrow \mathcal{R} \bigcup \{\varphi_{sum}/n, \ \Delta_{sum}/n\}$

f) $\boldsymbol{\alpha} = \arg\max_{\alpha} \mathcal{L}_D fg : four_m ethods \quad s.t. \quad \sum \boldsymbol{\alpha} = C$

g) $\lambda = \boldsymbol{\alpha}^{\intercal} \boldsymbol{\varphi}_{sum}$

h) remove $\{\varphi_{sum}/n, \Delta_{sum}/n\}$ from $\mathcal{R}$ for any $\alpha < 0.0001$ and $\{\varphi_{sum}/n, \Delta_{sum}/n\} \notin \mathcal{B}$

Figure 6.9: *Updated training steps in TGC-HCRFs*

6.3 Evaluation

We implement our TGC-HCRFs algorithm in Matlab. The raw features for training the TGC-HCRFs model are extracted with the help of Klaser's "3D" HOG computing tool [7]. In our evaluation, all the video patches share the same "3D" HOG parameter setting except for the video patch size. For each video patch, it is divided into $2 \times 2 \times 2$ cells, and each cell has $4 \times 4 \times 4$ subblocks by default. Every subblock is measured by the dodecahedron with half sphere orientation. These settings together give the final "3D" HOG descriptor a dimension of 48 ($2 \times 2 \times 2 \times 6$). This dimension is fixed for the raw "3D" HOG features through all the evaluation of TGC-HCRFs, and is irrelevant to different video patch sizes applied in the experiments.

Extracting video patches from all video frames in the training set would result in a very big data set which is hard to be handled by the computer. In order to get a manageable data set for training, we extract the video patches in every 2 frames

(a) (b) (c)

Figure 6.10: *Patch sampling on hand region with grids of size:* (a) *16;* (b) *24;* (c) *32*

during the training, but during the testing the video patches are still extracted frame by frame.

6.3.1 Sampling patterns

We fist evaluate our TGC-HCRFs method with different sampling patterns. In this evaluation, the number of hidden variables is set at 30. The video patch size is fixed at $32 \times 32 \times 5$. 4-way cross neighbouring system is applied in the model, and 4-way cross context window is used.

The evaluation runs for both ROI sampling and hand region sampling with three different grid sizes: 16 pixels, 24 pixels and 32 pixels. We finally obtain five sampling patterns in the evaluation as listed in the first column of table 6.1 [2] . Figure 6.10 illustrates the patterns of hand region sampling with three different grid sizes. From (a) to (c), they are in turn sampling with grids of size 16, 24 and 32 pixels. We can see that grid 16 sampling is much denser than the grid 32 sampling.

Table 6.1 lists the recognition performance of TGC-HCRFs with different sampling patterns, where "ROI 16" means that the sampling is performed on the ROI with grid size 16 pixels, and "Hand 32" means that the sampling is performed on the hand region with grid size 32. As we can see, the hand region sampling with grid size 16 gives the best recognition performance. Moreover, we can notice that the recognition performance with hand region sampling is better than the performance with ROI sampling. This is probably because that the hand region sampling can remove non-informative video patches from the background such that the gesture information carried by the gesture textons will not be covered by noise from the background. We

[2]the performance of ROI sampling with 16 pixel grid is not available due to too long training time

also can see that when hand region sampling is used, the denser the sampling is, the better performance the model has. This implies that the recognition prefers detailed information for distinguishing gestures.

Sampling pattern	Accuracy	Macro F	Micro F
ROI 24	$27.02 \pm 3.84\%$	0.2471	0.2702
ROI 32	$27.70 \pm 3.02\%$	0.2724	0.2770
Hand 16	$40.54 \pm 4.98\%$	0.3898	0.4054
Hand 24	$38.28 \pm 2.53\%$	0.3728	0.3928
Hand 32	$37.40 \pm 4.22\%$	0.3599	0.3740

Table 6.1: *TGC-HCRFs performance with different sampling patterns*

6.3.2 Video patch size

The hands in each video frame may present in different scale. Thus, it would be necessary to consider the scale issue in the recognition. In this section, we evaluate three different video patch sizes in the TGC-HCRFs model: size 16×16, 32×32 and 48×48. The sampling pattern is set as hand region sampling with grid size 16 which shows the best performance in the last section. The other model parameters are kept as before. Table 6.2 lists the evaluation results, where the "combined" means features from patches with three different sizes are concatenated to form a long multi-scale descriptor for recognition. It can be seen that this multi-scale descriptor has the best performance. This makes us think that building a more complex description of the video patch, such as rotating the patch with different angles, may further improve the recognition performance.

Patch size	Accuracy	Macro F	Micro F
16	$33.44 \pm 5.03\%$	0.3080	0.3344
32	$40.54 \pm 4.98\%$	0.3898	0.4054
48	$43.52 \pm 4.75\%$	0.4246	0.4352
Combined	$48.02 \pm 6.14\%$	0.4657	0.4802

Table 6.2: *TGC-HCRFs performance with different patch size*

No. of hidden variables	Accuracy	Macro F	Micro F
30	$48.02 \pm 6.14\%$	0.4657	0.4802
50	$48.98 \pm 4.53\%$	0.4817	0.4898
100	$47.97 \pm 7.77\%$	0.4599	0.4797

Table 6.3: *TGC-HCRFs performance with different number of hidden variables*

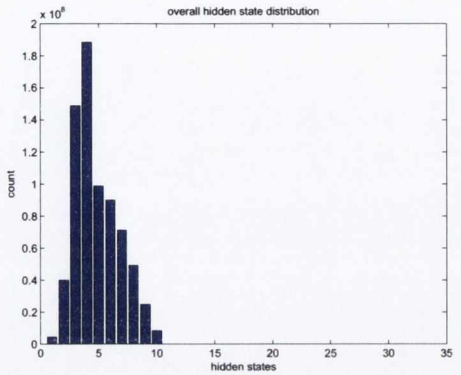
6.3.3 Hidden variables

With multi-scale patch description, we evaluate different number of hidden variables for the TGC-HCRFs model. As shown in table 6.3, the TGC-HCRFs models with 30, 50, and 100 hidden variables have very close recognition performance. Figure 6.11 counts the overall hidden variables used during the training of the TGC-HCRFs models.

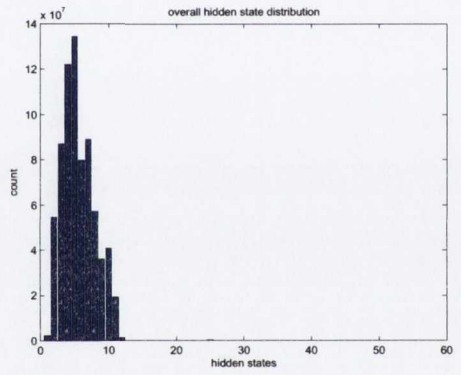
It can be seen that although different numbers of total hidden variables are set in the models, all TGC-HCRFs models tend to mainly use around 10 hidden states for the recognition. Although it is not clearly shown in the figures, the TGC-HCRFs models did explore other possible hidden variables during the training, but the accumulated counts of those hidden variables from all iterations are very small. It seems like the models trial the training with some hidden variables but find out that the first 10 hidden variables have the best explanation to the gestures, thus in the rest of the training the hidden variables that fail to interpret the gestures are ignored. This result implies that the TGC-HCRFs model can automatically determine the proper number of hidden variables for recognition, as long as the initial number of hidden variables is big enough. In other words, TGC-HCRFs has the ability of automatically determine the size of the visual vocabulary.

6.3.4 Neighbouring system

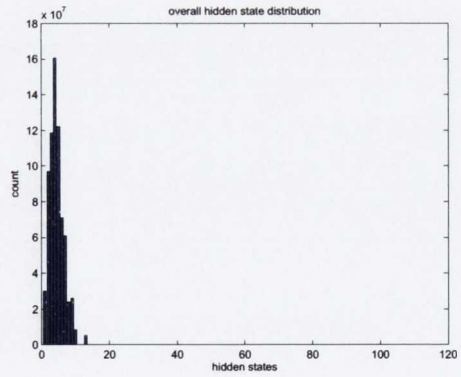
Different neighbouring systems are also evaluated in our evaluation. We experiment the neighbouring system up to 4-way neighbours are listed in table 6.4, where "2-way RL" means a 2-way neighbouring system with right and left nodes as neighbourhoods. Analogously, the "2-way UD" means a 2-way neighbouring system with up and down nodes as neighbourhoods. The "4-way cross" neighbouring system is depicted in figure 6.6(c), and the "4-way diagonal" neighbouring system is illustrated in figure 6.6(d). The other parameters in the experiment are set as 16 pixel grid sampling, multi-scale video patch description, 30 hidden variables and 4-way cross context window.



(a)



(b)



(c)

Figure 6.11: *The overall hidden variables used in TGC-HCRFs models with* (a) *30,* (b) *50,* (c) *100 hidden variables*

Neighbours	Accuracy	Macro F	Micro F
None	$18.30 \pm 9.21\%$	0.1270	0.1830
2-way RL	$39.74 \pm 11.08\%$	0.3763	0.3974
2-way UD	$46.09 \pm 4.74\%$	0.4602	0.4609
4-way cross	$48.02 \pm 6.14\%$	0.4657	0.4802
4-way diagonal	$49.98 \pm 8.22\%$	0.4905	0.4998

Table 6.4: *TGC-HCRFs performance with different neighbouring system*

From the result table 6.4, it can be seen that a model with 4-way diagonal neighbouring system gives the best performance. However, a more interesting finding is that the "2-way UD" neighbouring system performs much better than the "2-way RL" neighbouring system. This means that the vertical neighbourhoods bring in more useful information for discerning gestures.

6.3.5 Context window

Lastly, we evaluate the context window size of TGC-HCRFs model. As the neighbouring system, the context window size is also evaluated up to the 4-way context window as listed in the first column of table 6.5. The evaluation results listed in table 6.5 confirm that increasing the context window can improve the recognition performance.

Neighbours	Accuracy	Macro F	Micro F
None	$34.73 \pm 3.42\%$	0.3328	0.3473
2-way RL	$45.31 \pm 3.39\%$	0.4473	0.4531
2-way UD	$44.98 \pm 6.58\%$	0.4359	0.4498
4-way cross	$48.02 \pm 6.14\%$	0.4657	0.4802
4-way diagonal	$48.69 \pm 4.81\%$	0.4774	0.4869

Table 6.5: *TGC-HCRFs performance with different context window*

6.3.6 Hidden state initialization

In all previous experiments, the bootstrapping strategy is used to alleviate the local maxima problem. The local maxima problem may also be alleviated by initializing the hidden structure of each training example with random states. Therefore, in this experiment, we substitute the bootstrapping strategy with random state initialization in the training algorithm. At the start of the training algorithm, the states of the

Group	1	2	3	4	5	Mean
Random states	35.78%	30.64%	47.79%	51.85%	42.71%	$41.79 \pm 8.60\%$
Bootstrapping	43.17%	55.94%	52.47%	47.11%	41.40%	$48.02 \pm 6.14\%$

Table 6.6: *TGC-HCRFs with random state initialization and bootstrapping*

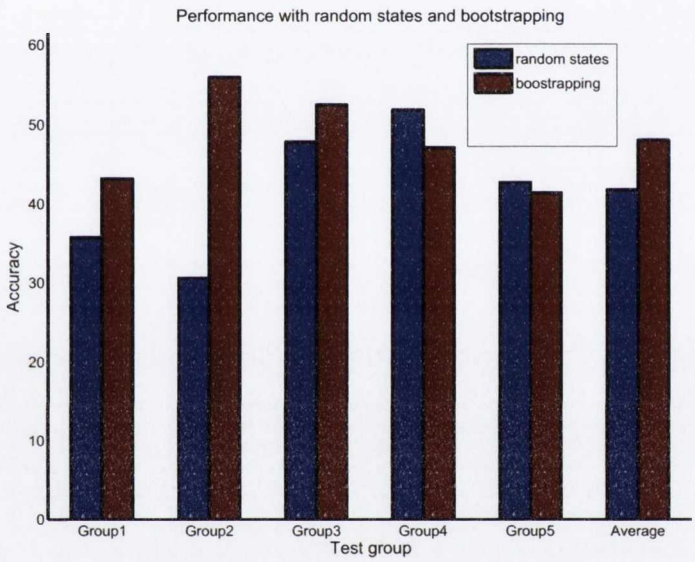


Figure 6.12: *A plot of group accuracies of models with bootstrapping and random state initialization*

hidden structure for each training sample are initialized with random integers between 1 and the number of hidden variables.

Table 6.6 lists the group accuracies of both models with bootstrapping and random state initialization. As shown in figure 6.12, the bootstrapping strategy gives a slightly better performance than random state initialization. Another benefit from applying bootstrapping strategy in the training is that the training time can be greatly reduced, since only a portion of training set is selected by the bootstrapping in each training iteration.

6.4 Summary

In this chapter, we develop a TGC-HCRFs model for hand washing gesture recognition. The model unifies texton analysis and HCRFs within the same framework such that the gesture textons can be modelled as the hidden variables in HCRFs. Our TGC-HCRFs method models the gesture spatial layout as grid graph, and each node of the graph is a small spatial-temporal video patch which is described by "3D" HOG features. The grid graph structure provides a more flexible way than the linear chain HCRFs to include contextual and neighbouring information. It also provides an efficient way to preserve the spatial configuration of gestures which can not be conserved by the traditional BOFs methods.

Our TGC-HCRFs model is trained following Joachims' cutting-plane algorithm, however, as a semi-convex model, the training suffers from the local maxima problem, therefore we propose a bootstrapping strategy in the training to alleviate the local maxima problem. Our evaluation of the TGC-HCRFs model also provides a practical study of applying HCRFs with grid graph structure for gesture recognition. Several key parameters of the model are evaluated, such as the sampling patterns, hidden variable number, neighbourhoods definition, etc. The evaluation result shows that the fine-grid sampling of the gestures can generally give better recognition performance. The evaluation also shows that the multi-scale description of the video patches can significantly improve the recognition accuracies. This make us think of supplying more complex patch descriptors to TGC-HCRFs in the future work. A distinct finding from the evaluation is that the TGC-HCRFs method has the ability of automatically determining the size of a visual vocabulary for a given recognition task. Since the local features required in TGC-HCRFs model can be computed simultaneously and efficiently, TGC-HCRFs model is also a good candidate for the real-time applications.

Chapter 7

Discussion and Conclusions

In this thesis, we research into the hand washing gesture recognition as a new subject in gesture recognition domain. Ten hand washing gestures are defined and analysed from different perspectives using computer vision methods. Our research shows that hand washing gestures can be assessed by vision-based methods effectively and efficiently.

7.1 Methods Review

Four computer vision methods are developed and evaluated for hand washing gesture recognition in this thesis.

7.1.1 Recognition using static postures

The recognition using static postures method builds a multi-class hand washing gesture model based on densely computed 2D HOG features from image ROIs. It classifies the hand washing gestures frame by frame with only static information. The method requires localizing the hand area in a preprocessing step. It encodes the hand silhouettes of each gesture in grid for recognition, and can cope with small amount of shape deformation, rotation and position shift. The evaluation result shows that the recognition with fine-grid HOG has better performance than the recognition with coarse-grid HOG.

7.1.2 Recognition using sequence labelling

Recognition using sequence labelling method treats the hand washing gestures as sequences and learns the internal structure of each sequence with the linear-chain HCRFs. The method captures the static gesture information with the 2D HOG features while the dynamic gesture movement is modelled by the first-order Markov process. A distinct property of the method is that the model can incorporate neighbouring frames as contextual information for the recognition. However, since the high dimension 2D HOG features used in our evaluation is very descriptive, adding contextual information from neighbouring frames does not significantly improve the recognition performance. This implies that the enhancement from contextual information depends on the primal features used in the model. From the evaluation, we also notice that a very large hidden variable pool does not increase recognition performance significantly as well. This is suspected that the model is dominated by some hidden variables even though more hidden variables are available.

The recognition using sequence labelling method is trained within a max-margin training framework which can handle large training set, and naturally reports the classification results in terms of sequences. In order to give frame-by-frame predication, a time-shift window is applied. A short sequence segment is extracted by the time-shift window and is classified by the HCRFs model. The output class label is then assigned to the middle frame within the time-shift window. In our evaluation, we notice that the classification performance increases along with the time-shift window grows, in which the confidence of the predication is built up. It can be seen that the recognition using sequence labelling method has better performance than recognition using static postures method for frame-by-frame classification. This is probably mainly because the sequence labelling method utilizes information from both spatial and temporal domains for recognition.

7.1.3 Recognition using space-time interest points

Both previous methods require the ROI detection in the preprocessing step, and strongly depend on the 2D grid HOG representation. The ROI detection may meet difficulties when the background is cluttered and the lighting condition is uncontrollable. The dense grid HOG features also have limited capacity of handling big spatial variety in hand washing gestures. Thus, the recognition using space-time interest points

method is explored. The recognition using space-time interest points method does not require ROI detection. It searches interest points over the whole image area by Dollar's cuboid detector. From the evaluation, we notice that most space-time interest points are detected near the motion boundaries of the gestures.

Detected interest points are described by the "3D" HOG descriptor, and are pooled together to build a visual vocabulary. The visual vocabulary is generated with ERC-Forests, an efficient alternative to K-means clustering. An important step in BOFs methods is to choose the vocabulary size. In ERC-Forest, the vocabulary size is usually controlled through the number of trees used in the forests. In our evaluation, instead we experimentally control the vocabulary size through the depth of trees in the forests. Our evaluation shows that the recognition performance grows gradually when we increase the depth of the trees, which means that the depth of trees can also be used to control the discriminative power of the visual vocabulary.

Recognition using space-time interest points method performs the classification with a rich set of local features regardless of the global appearance and motion of gestures. However, due to no any spatial configuration of the gestures is conserved, any partial correlation between two sets of interest points can confuse the classifiers regardless of their relative positions in the video frames.

7.1.4 Recognition using TGC-HCRFs

The last evaluated method is our newly developed method TGC-HCRFs. The method unifies the texton analysis and HCRFs method in a same framework such that the gesture textons can be modelled by the hidden states in HCRFs. Comparing to recognition using space-time interest points method, TGC-HCRFs is also a sparse representation based method, but the spatial configuration of gestures can be preserved through the grid graphs in HCRFs. Comparing to recognition using sequence labelling method, the grid graph used in TGC-HCRFs provides a flexible way to include the contextual and neighbouring information for recognition. Our evaluation of TGC-HCRFs shows that multi-scale description of video patches outperforms single-scale description, and the fine-grid sampling of the gestures is preferred in TGC-HCRFs. Moreover, the TGC-HCRFs method can automatically determine the visual vocabulary size for recognition. Given a big initial hidden variable set, TGC-HCRFs can choose a proper subset of hidden variables to represent the gestures, while the rest of the hidden variables is ignored. Since the local features used in TGC-HCRFs can be extracted simultaneously,

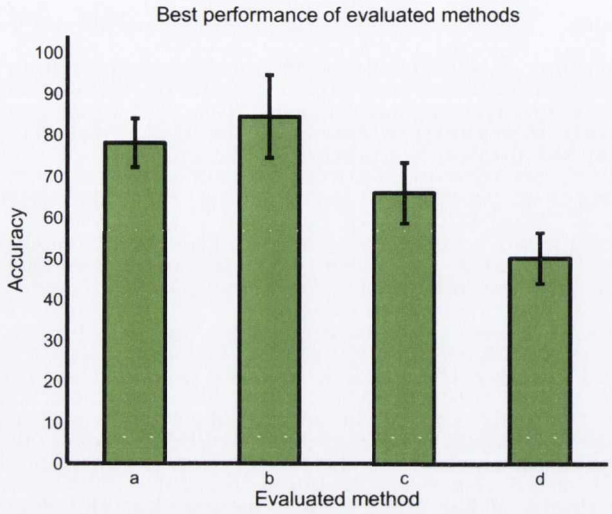


Figure 7.1: *Best performance of four evaluated methods:* (a) *recognition using static postures;* (b) *recognition using sequence labelling;* (c) *recognition using space-time interest points;* (d) *recognition using TGC-HCRFs*

TGC-HCRFs model is also a good candidate for real-time applications.

7.2 Discussion

In this section, our four evaluated methods are compared and analysed in terms of hand washing gesture recognition. Several interesting findings are drawn and discussed as below.

7.2.1 Dense and sparse representation

Figure 7.1 plots the best results from our four evaluated methods. It can be seen that the dense representation based methods are more suitable for hand washing gesture recognition than the sparse representation based methods. In figure 7.1, the first method and the second method are the dense representation based approaches, while the third and fourth methods are sparse representation based approaches. This means that the global gesture shape configuration has the critical information for the recognition. In the first and second methods, the global gesture spatial layout is hardcoded into the grid and strongly represented by the 2D dense HOG features. This grid encod-

ing may have some difficulty to represent the variety of a gesture, but the evaluation shows that it is well suited to the hand washing gesture recognition. In the fourth method, although grid structure is applied as well, the model does not capture the global spatial information.

7.2.2 Fine-scale description preference

From the first method and the fourth method, it can be seen that hand washing gesture recognition prefers detailed gesture description. In the first method, the finer-grid HOG features give the better recognition performance. The highest one is the 3136 HOG which has the finest quantization in both space and direction. The fourth method has similar phenomena. When the sampling grid is 16 and multi-scale patches are used, the recognition gets the best performance.

The reason of preferring details in hand washing gesture recognition may be because the parts of hands are relatively small and move fast. The difference between gestures are relatively small as well. The recognition algorithms need to obtain the fine scale information in order to distinguish different gestures. Although much concerning the details may overtrain the model, our evaluation in this thesis shows that, in hand washing gesture recognition, supplying more gesture details gives better recognition performance.

7.2.3 Vocabulary size

In the third method and the fourth method, a visual vocabulary needs to be built up. However, choosing a proper size of the vocabulary is not easy. small size vocabulary may lack of discrimination while a large vocabulary may result in overfitting. In the third method, we control the size of the vocabulary by the depth of trees in forests. However, if an optimal vocabulary size is required, the depth number has to be searched through the art of experiment.

On the contrary, the fourth method can automatically choose a vocabulary size for recognition. All we need to do is to supply a relatively large number of hidden variables. The method can automatically determine how many hidden variables it needs during the training. However, we also notice that the method may have a trend to give a vocabulary as small as possible. The model may lose some discriminative power in the recognition. Thus, in the future work, it would be interesting to encourage the method

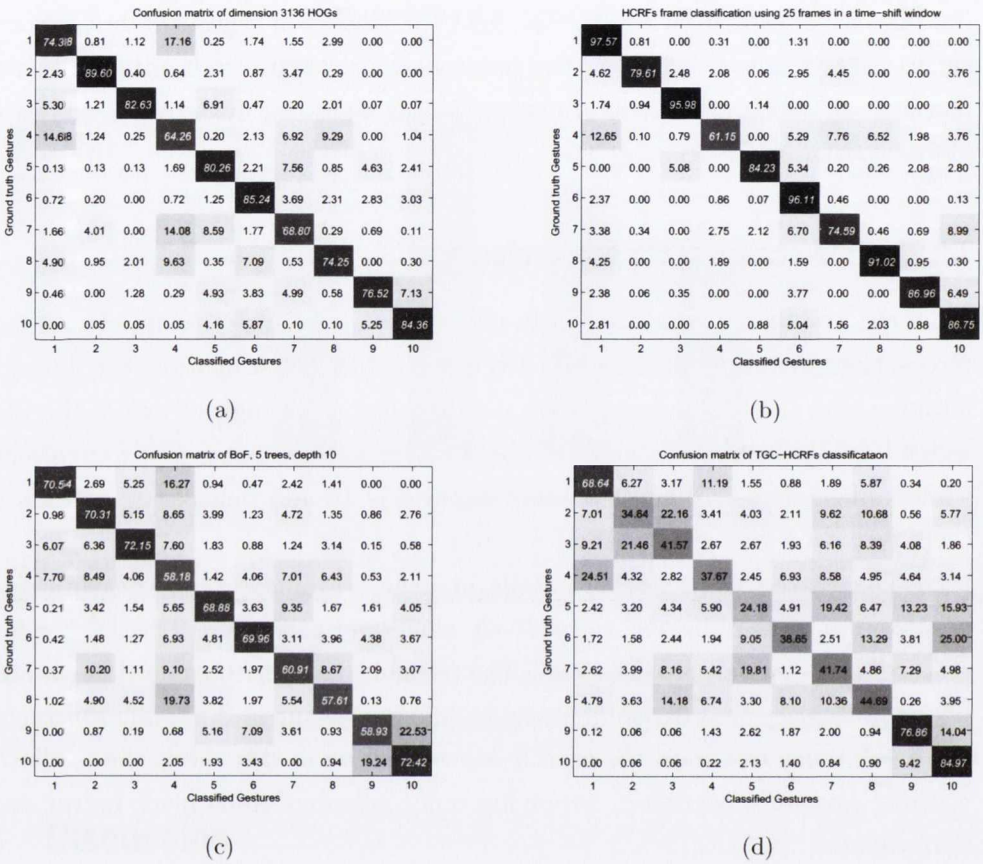


Figure 7.2: *Confusion matrices of four evaluated methods:* (a) *recognition using static postures;* (b) *recognition using sequence labelling;* (c) *recognition using space-time interest points;* (d) *recognition using TGC-HCRFs*

to have a bit bigger size vocabulary.

7.2.4 Gesture analysis

Figure 7.2 shows the confusion matrices when the four evaluated methods give their best performance. It can be seen that gesture 1 and gesture 4 are usually misclassified to each other. This is not surprising as gesture 1 and gesture 4 are very similar. The major difference is only around the region near the finger tips. A possible way to alleviate this problem is to incorporate adaboost in the model training to enforce paying much attention to the small difference between gesture 1 and 4.

Gestures 9 and 10 are classified relatively well in recognition via static postures and recognition in sequence labelling. This is due to that the hand movement in gestures 9 and 10 is small. The spatial hand configuration plays an important role in the classification. Contrarily, recognition via space-time interest points does not preserve any spatial configuration of the hands, thus its classification performance of gestures 9 and 10 is poor.

It is surprising that gesture 2 and gesture 3 can be distinguished very well by all methods except for the TGC-HCRFs method. Intuitively gesture 2 and gesture 3 are very similar as it is hard to tell which hand is on the top and which hand is on the bottom without any 3D depth information. It is suspect that features from parts around the root of thumbs play important roles for telling these two gesture apart.

Comparatively speaking, gestures 7 and 8 are classified poorly for all four approaches. The possible reason is that the variance in gestures 7 and 8 is large. The gestures could appear dynamic or static depending on the amplitude of the palm rotation. The changing of hand positions and angles to the camera can also cause large appearance difference in gestures 7 and 8.

7.2.5 Real-time processing

Our research strives to interpret and discriminate hand washing gestures effectively and efficiently with computer vision methods. Thus, a real-time processing of hand washing videos is desired in our algorithm design. In our four evaluated methods, the sequence labelling method gives the best frame by frame recognition performance, however, it is not the first choice for real-time applications. This is primarily because that the classification of sequence labelling method needs to view the whole sequence before making a predication.

On the contrary, the fourth TGC-HCRFs method has the worst recognition performance among all evaluated methods, but it could be a good candidate for real-time applications. The TGC-HCRFs method can extract local features simultaneously and make classification by simple linear operations. No global normalization is required during the processing such that the algorithm can be easily parallelized and implemented on the Graphics Processing Unit (GPU) or Field-Programmable Gate Arrays (FPGAs).

Based on current performance, the recognition with static postures would be the best choice for real-time applications, since the algorithm gives very good recognition

performance, and can be processed frame by frame very efficiently.

7.2.6 Improvement of TGC-HCRFs

Currently, the recognition performance of TGC-HCRFs method is inferior to the other three methods. However, the performance might be enhanced in recent future by the improvement of the TGC-HCRFs training algorithm. Firstly, we notice that the graph cuts inference dose not provide any posterior measurement of the inferred hidden structure. In other words, the algorithm does not possess the probability of any hidden variable assignment which might help the training. This drawback of the training algorithm could be overcome if an uncertainty measure of the graph cuts solution is provided [94], or the graph cuts approximation can give M most probable hidden structure outputs [173]. Secondly, in order to handle multi-class problems, the training of TGC-HCRFs is formulated similarly to Crammer and Singer's multi-class SVM [39], which needs to compute the feature difference 6.4 during the training. This means that the training is only guided by the negative constraints in the algorithm formulation since the feature difference from positive constraints are all zeros. Therefore, the performance of TGC-HCRFs method might be further improved if any effective positive constraints can be added to the training of TGC-HCRFs.

7.3 Conclusions

The goal of our research is to assess hand washing techniques applied in the hand washing activities robustly and effectively with vision-based methods. Aiming to this goal, 10 hand washing patterns are defined following the WHO hand washing gesture recommendation, and four computer vision methods are developed in this thesis. These four methods are evaluated with a large set of hand washing videos and a large number of experiments.

Our evaluation shows that recognizing hand washing gestures frame by frame using sequence labelling method could give the best performance among all four evaluated methods. Recognition using sequence labelling method models the gestures with first order Markov process, and applies a time-shift window during the test to give a frame by frame predication. The method can give very good accuracy when a large time-shift window is applied, which can be explained as that the large window includes

much gesture history and accumulates the confidence for classification. However, the large time-shift window comes with a price of long processing latency in the predication. Therefore, from an efficiency point of view, recognizing hand washing gestures using static postures would be the best choice for real applications. The method of recognition using static postures gives the second best performance among four evaluated methods. The method extracts 2D dense global HOG features to represent the hands in each frame, and the evaluation shows that the recognition with fine-grid HOG which provides detailed hand posture information gives better performance than the recognition with coarse-grid HOG.

Both above methods use dense representation of gestures for classification, which can be affected by the ROI detection results. Therefore, we explore the sparse representation in the recognition using space-time interest points method. It is believed that the gestures can be well represented by a rich set of local features, regardless of global appearance and motion. Recognition using space-time interest points method gives slightly inferior performance to the methods with dense representation. This implies that hand washing gesture recognition benefits from the global structure information of gestures in the dense representation methods. However, as a sparse representation method, recognition using space-time interest points method can handle the cluttered background and uncontrollable lighting condition.

Our TGC-HCRFs method innovatively unifies the texton analysis and HCRFs within the same framework. The method attempts to represent the gestures with sparse representation and preserve the spatial configuration of the gestures. The evaluation of the method also provides a practical study of applying HCRFs with grid graph structure for gesture recognition. Although the performance of TGC-HCRFs method is not as good as the other three methods, the features used in TGC-HCRFs can be extracted simultaneously, which makes the method a good candidate for real-time applications if parallel processing is applied.

Some our other findings from the evaluation of the four methods are also interesting, and they may be applicable to other types of human action recognition. For example, the discriminative power of visual vocabulary built by ERC-Forests can be controlled by the depth of trees, and TGC-HCRFs method can automatically determine the visual vocabulary size for recognition.

Hand washing activity is one of the most important ways to prevent the spread of infection and illness, however assessing the quality of the hand washing activity is

not easy. In this thesis, our research demonstrates that vision-based methods can be used to measure the hand washing techniques applied in hand washing activity robustly and effectively, which provides an economic and efficient way for hand washing activity assessment.

7.3.1 Contributions

In this thesis, we have made following contributions:

- Hand washing gesture recognition is studied as a new type of gesture recognition task. 10 hand washing gesture patterns are defined and analysed from multiple perspectives.
- Four different computer vision methods are developed and evaluated for hand washing gesture recognition.
- A new method TGC-HCRFs is proposed, which unifies the texton analysis and HCRFs within the same framework. The evaluation of the TGC-HCRFs method also provides a practical study of HCRFs with grid graph for gesture recognition, which has not been seen in literature.
- The sequence labelling tool HCRFs is applied for gesture recognition. The key parameters such as the context window size, number of hidden variables, etc. are evaluated.
- Different test sequence length is evaluated for the linear-chain HCRFs. It is found that the longer test sequence is used, the better classification results can be obtained.
- The depth of trees in ERC-Forests is evaluated for controlling the discriminative power of visual vocabulary.
- A bootstraping strategy is applied in the max-margin training of HCRFs with grid graph, which alleviates the local optimum problem and reduces the overall training time.
- The proposed TGC-HCRFs method can automatically determine the visual vocabulary size for recognition.

- It is found that gesture recognition favours fine-grid sampling and prefers rich description of video patches. Moreover, the vertical neighbours of a video patch are found to be more informative than the horizontal neighbours.

7.4 Future Work

Based on the work presented in this thesis, several issues have also been identified and can be improved in the future work.

Firstly, it can be seen that all our evaluation is based on the HOG description. It would be interesting to examine some other types of low-level features as the descriptors, such as the moments [74], the motons [174] or even simple image pixel difference [147][175].

Secondly, as mentioned previously, the training algorithm of TGC-HCRFs can be further improved. The model may give better performance if positive constraints can be added in the training and the inference can have uncertainty output of the solution. Moreover, some other strategies can also be examined for the local optimum problem, for example, simulated annealing [17] or Quantum adiabatic machine learning [124].

Finally, it would be interesting to apply all the evaluated methods to other types of human actions, and see if the same finding can be drawn.

Publications

Jiang Zhou, Vilarino, F, Li Xuchun and Gerard, L., "Statistical analysis of ground truth in human labeled data", *Machine Vision and Image Processing Conference, 2007. IMVIP 2007. International*, pp.211, 5-7 Sept. 2007

D.F. Llorca, F. Vilarino, J. Zhou and G. Lacey, "A multi-class SVM classifier ensemble for automatic hand washing quality assessment", *British Machine Vision Conference, 2007*, pp.10, September 2007

Padraig Curran, J. Buckley, B. OFlynn, X. Li, J. Zhou, G. Lacey and S. C. OMathuna, "VAMP A Vision Based Sensor Network for Health Care Hygiene", *14th Nordic-Baltic Conference on Biomedical Engineering and Medical Physics, NBC 2008*, 20:485-488, June 2008

Stefan Ameling, Johnson Li, Jiang Zhou, Anarta Ghosh and Gerard Lacey, "A vision-based system for hand washing quality assessment with real-time feedback", In. *Proceedings of the 8th IASTED International Conference on Biomedical Engineering, Biomed 2011*, pp.475-482, February 2011

Anarta Ghosh, Stefan Ameling, Jiang Zhou, Gerard Lacey, Eilish Creamer, Anthony Dolan, Orla Sherlock, Hilary Humphreys, "Pilot evaluation of a ward-based automated hand hygiene training system", *American Journal of Infection Control*, volume 41, issue 4, pp.368-370, April 2013

Bibliography

[1] J. G. Zhang and T. N. Tan. Brief Review of Invariant Texture Analysis Methods. *Pattern Recognit.*, 35(3):735–747, Mar. 2002.

[2] M. Shimosaka, T. Mori and T. Sato. Robust Action Recognition and Segmentation with Multi-Task Conditional Random Fields. In *Proceeding of 2007 IEEE International Conference on Robotics and Automation,*, pages 3780 –3786, Apr. 2007.

[3] Y. Wang and G. Mori. Max-margin hidden conditional random fields for human action recognition. In *Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* , volume 0, pages 872–879, Los Alamitos, CA, USA, June 2009. IEEE Computer Society.

[4] A. A. Efros, A. C. Berg,G. Mori and J. Malik. Recognizing Action at A Distance. In *Proceeding of IEEE International Conference on Computer Vision*, pages 726–733, California University, Berkeley, CA, USA, 2003.

[5] A. F. Bobick and J. W. Davis. The Recognition of Human Movement Using Temporal Templates. In *IEEE transactions on pattern analysis and machine intelligence*, volume 23, pages 257–267, Georgia Tech. Res. Inst., Atlanta, GA, USA, 2001.

[6] A. Gunawardana, M. Mahajan, A. Acero and J. C. Platt. Hidden conditional random fields for phone classification. In *International Conference on Speech Communication and Technology*, pages 1117–1120, September 2005.

[7] A. Kläser, M. Marszaek and C. Schmid. A Spatio-Temporal Descriptor Based on 3d-Gradients. In *BMVC08*, pages 275:1–10, Leeds, UK, September 2008.

[8] A. McCallum, D. Freitag and F. Pereira. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *ICML '00 Proceedings of the Seventeenth International Conference on Machine Learning* , pages 591–598. Morgan Kaufmann, San Francisco, CA, 2000.

[9] A. Mittal and N, Paragios. Motion-Based Background Subtraction Using Adaptive Kernel Density Estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, volume 2, pages II–302 – II–309, 2004.

[10] A. Özgür and L. Özgür and T. Güngör. Text Categorization with Class-Based and Corpus-Based Keyword Selection. In *Computer and Information Sciences - ISCIS 2005*, volume 3733 of *Lecture Notes in Computer Science*, pages 606–615. Springer Berlin Heidelberg, 2005.

[11] A. Quattoni, M. Collins and T. Darrell. Conditional Random Fields for Object Recognition. In Lawrence K. Saul, Yair Weiss, and Leon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1097–1104, Cambridge, MA, 2005. MIT Press.

[12] A. Torralba, K. P. Murphy and W. T. Freeman. Shared Features for Multiclass Object Detection. In Jean Ponce, Martial Hebert, Cordelia Schmid, and Andrew Zisserman, editors, *Toward Category-Level Object Recognition*, volume 4170 of *Lecture Notes in Computer Science*, pages 345–361. Springer Berlin Heidelberg, 2006.

[13] A. Yilmaz and M. Shah. A Differential Geometric Approach to Representing the Human Actions. *Computer Vision and Image Understanding*, 109(3):335–351, 2008.

[14] K. Alahari, P. Kohli, and P.H.S. Torr. Reduce, reuse, recycle: Efficiently solving multi-label mrfs. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, June 2008.

[15] B. Taskar, C. Guestrin and D. Koller. Max-Margin Markov Networks. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[16] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching. In *Proceedings of the 5th international joint conference on Artificial intelligence*, volume 2, pages 659–663, 1977.

[17] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st ed. 2006. corr. 2nd printing 2011 edition, Oct. 2007.

[18] John M. Boyce. Measuring healthcare worker hand hygiene activity: current practices and emerging technologies. *Infect Control Hosp Epidemiol*, 32(10):1016–28, Oct. 2011.

[19] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 648 –655, Jun. 1998.

[20] C. Achard, X. T. Qu, A. Mokhber and M. Milgram. A Novel Approach for Recognition of Human Actions with Semi-global Features. In *Proceeding of Machine Vision and Application*, volume 19, pages 27–34, 2008.

[21] C. B.Weng, Y. Li, M. M. Zhang, K. D. Guo, X. Tang and Z. G. Pan. Robust Hand Posture Recognition Integrating Multi-Cue Hand Tracking. In *Entertainment for Education. Digital Techniques and Systems*, volume 6249 of *Lecture Notes in Computer Science*, pages 497–508. Springer Berlin Heidelberg, 2010.

[22] C. C. Chang and C. J. Lin. LIBSVM: A Library for Support Vector Machines. In *Proceeding of ACM Transactions on Intelligent Systems and Technology*, volume 2, pages 1–39, 2011.

[23] C. G. Healey and J. T. Enns. Attention and Visual Memory in Visualization and Computer Graphics. In *IEEE Transactions on Visualization and Computer Graphics*, volume 99, pages 1–20, 2011.

[24] C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[25] C. J. Yang, Y. L. Guo, H. Sawhney and R. Kumar. Learning Actions Using Robust String Kernels. In *Proceedings of the 2nd conference on Human motion:*

understanding, modeling, capture and animation, pages 313–327, Berlin, Heidelberg, 2007. Springer-Verlag.

[26] C-N. J. Yu and T. Joachims. Learning Structural SVMs with Latent Variables. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1169–1176, New York, NY, USA, 2009. ACM.

[27] C. P. Papageorgiou, M. Oren and T.o Poggio. A General Framework for Object Detection. In *Proceeding of Sixth International Conference on Computer Vision, 1998.*, pages 555–562, 1998.

[28] T. Darrell C. R. Wren, Ali Azarbayejani and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:780–785, 1997.

[29] C. Rao, A. Yilmaz and M. Shah. View-Invariant Representation and Recognition of Actions. *International Journal of Computer Vision*, 50:203–226, November 2002.

[30] C. Schmid and R. Mohr. Local Grayvalue Invariants for Image Retrieval. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19, pages 530–535, Los Alamitos, CA, USA, 1997. IEEE Computer Society.

[31] C. Schuldt, I. Laptev and B. Caputo. Recognizing Human Actions: a Local SVM Approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36, Cambridge, UK., 2004.

[32] C. Sminchisescu, A. Kanaujia and D. Metaxas. Conditional Models for Contextual Human Motion Recognition. In *Proceeding of Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005.*, volume 2, pages 1808–1815, 2006.

[33] C. Thurau. Behavior Histograms for Action Recognition and Human Detection. In *Proceedings of the 2nd conference on Human motion: understanding, modeling, capture and animation*, volume 4814, pages 299–312, 2007.

[34] C. Thurau and V. Hlavac. n-Grams of Action Primitives for Recognizing Human Behaviour. In *Computer Analysis of Images and Patterns*, volume 4673, pages 93–100, August 2007.

[35] C. Wallraven, B. Caputo and A. Graf. Recognition with Local Features: the Kernel Recipe. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, volume 1 of *ICCV '03*, pages 257–264, Washington, DC, USA, 2003. IEEE Computer Society.

[36] Ch-Ch Wang and Co-Ch Wang. Hand Posture Recognition Using Adaboost with SIFT for Human Robot Interaction. In *Proceedings of the International Conference on Advanced Robotics (ICAR'07)*, Jeju, Korea, August 2007.

[37] D. Chetverikov and R. Péteri. A Brief Survey of Dynamic Texture Description and Recognition. In *Proceeding of Computer Recognition Systems (2005)*, pages 17–26, 2005.

[38] Dorin Comaniciu and Peter Meer. Mean shift: a robust approach toward feature space analysis. volume 24, pages 603–619, 2002.

[39] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res.*, 2:265–292, March 2002.

[40] E Creamer, S Dorrian, A Dolan, O Sherlock, D Fitzgerald-Hughes, T Thomas, J Walsh, A Shore, D Sullivan, P Kinnevey, and et al. When are the hands of healthcare workers positive for methicillin-resistant staphylococcus aureus? *The Journal of hospital infection*, 75(2):107–111, 2010.

[41] Val Curtis and Sandy Cairncross. Effect of washing hands with soap on diarrhoea risk in the community: a systematic review. *The Lancet Infectious Diseases*, 3(5):275–281, 2003.

[42] D. G. Lowe. Object Recognition from Local Scale-Invariant Features. In *The Proceedings of ICCV '99 Proceedings of the International Conference on Computer Vision*, volume 2 of *ICCV '99*, pages 1150–1157, Washington, DC, USA, 1999. IEEE Computer Society.

[43] D. Weinland, R. Ronfard and E. Boyer. Free Viewpoint Action Recognition Using Motion History Volumes. *Computer Vision and Image Understanding*, 104:249–257, November 2006.

[44] Jonathan J. Hull Dar-Shyang Lee and Berna Erol. A bayesian framework for gaussian mixture background modeling. *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, 3:III–973–6, 2003.

[45] Can Demirkesen and Hocine Cherifi. A comparison of multiclass svm methods for real world natural scenes. In Jacques Blanc-Talon, Salah Bourennane, Wilfried Philips, Dan Popescu, and Paul Scheunders, editors, *Advanced Concepts for Intelligent Vision Systems*, volume 5259 of *Lecture Notes in Computer Science*, pages 752–763. Springer Berlin Heidelberg, 2008.

[46] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proceedings of the 14th International Conference on Computer Communications and Networks*, pages 65–72, Washington, DC, USA, 2005. IEEE Computer Society.

[47] E. Rivlin, A. Adam, and S. Ilan. Robust Fragments-based Tracking using the Integral Histogram. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1 of *CVPR '06*, pages 798–805, Washington, DC, USA, 2006. IEEE Computer Society.

[48] E. Shechtman and M. Irani. Matching Local Self-Similarities across Images and Videos. In *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07.*, pages 1–8, Jun. 2007.

[49] E. Shechtman and M. Irani. Space-Time Behavior Based Correlation OR How to tell if two underlying motion fields are similar without computing them? In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 29, pages 2045–2056, November 2007.

[50] Ahmed M. Elgammal, David Harwood, and Larry S. Davis. Non-parametric model for background subtraction. In *Proceedings of the 6th European Conference on Computer Vision-Part II*, ECCV '00, pages 751–767, London, UK, 2000. Springer-Verlag.

[51] F. Attneave. Some Informational Aspects of Visual Perception. In *Psychological Review*, volume 61, pages 183–193, May 1954.

[52] F. Moosmann, B. Triggs and F. Jurie. Fast Discriminative Visual Codebooks using Randomized Clustering Forests. In *Advances in Neural Information Processing Systems 19*, volume 2, pages 985–992, 2006.

[53] F. Porikli. Integral Histogram: A Fast Way To Extract Histograms in Cartesian Spaces. In *Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, volume 1, pages 829–836, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

[54] F. W. Liu and Y. Jia. Human Action Recognition Using Manifold Learning and Hidden Conditional Random Fields. In *Proceedings of the 2008 The 9th International Conference for Young Computer Scientists*, pages 693–698, Washington, DC, USA, 2008. IEEE Computer Society.

[55] Lorna Fewtrell, Rachel B Kaufmann, David Kay, Wayne Enanoria, Laurence Haller, and John M Colford. Water, sanitation, and hygiene interventions to reduce diarrhoea in less developed countries: a systematic review and meta-analysis. *The Lancet infectious diseases*, 5(1):42–52, 2005.

[56] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–407, 2000.

[57] G. Csurka and C. R. Dance and L. X. Fan and J. Willamowski and C. Bray. Visual Categorization with Bags of Keypoints. In *Proceeding of Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[58] G. R. Bradski and J. W. Davis. Motion Segmentation and Pose Recognition with Motion History Gradients. *Machine Vision and Applications*, 13(3):174–184, July 2002.

[59] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, Apr. 2006.

[60] Z. Ghahramani. Building Blocks of Movement. *Nature*, 407(6805):682–683, October 2000.

[61] G. Guerra-Filho and Y. Aloimonos. A sensory-motor language for human activity understanding. *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 69–75, Dec 2006.

[62] Rahul Gupta. Conditional random fields. Technical report, IIT Bombay, 2006.

[63] H. Bay, T. Tuytelaars and L. V. Gool. Speeded-Up Robust Features (SURF). In *Proceeding of Computer Vision and Image Understanding (2008)*, volume 110, pages 346–359, June 2008.

[64] H. Park, J. Choo, B. L. Drake and J. Kang. Linear Discriminant Analysis for Data with Subcluster Structure. In *Proceeding of 19th International Conference on Pattern Recognition, 2008. ICPR 2008.*, pages 1–4, Dec. 2008.

[65] H. Wang, M. M. Ullah, A. Kläser, I. Laptev and C. Schmid. Evaluation of Local Spatio-temporal Features for Action Recognition. In *Proceeding of BMVC 2009 - British Machine Vision Conference (2009)*, 2009.

[66] H. Z. Ning, W. Xu, Y. H. Gong and T. Huang. Latent Pose Estimator for Continuous Action Recognition. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, pages 419–433, Berlin, Heidelberg, 2008. Springer-Verlag.

[67] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Proceedings of the 4th Alvey Vision Conference*, volume 15, pages 147–151, 1988.

[68] Jesse Hoey. Tracking using flocks of features, with application to assisted handwashing. In *Proc. British Machine Vision Conference (BMVC)*, pages 367–376, Edinburgh, Scotland, September 2006.

[69] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, Mar. 2002.

[70] I. Laptev. Improving Object Detection with Boosted Histograms. *Image Vision Comput.*, 27:535–544, April 2009.

[71] I. N. Junejo, E. Dexter, I. Laptev, P. Pérez, I. Rennes and B. Atlantique. Cross-View Action Recognition from Temporal Self-similarities. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, pages 293–306, Berlin, Heidelberg, 2008. Springer-Verlag.

[72] J. C. Niebles, Wang, H. C. and F. F. Li. Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. *International Journal of Computer Vision*, 79:299–318, September 2008.

[73] J. D. Lafferty, A. Mccallum and F. C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 01)*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers.

[74] J. Flusser. Moment Invariants in Image Analysis. In *Proceeding of Engineering and Technology (2006)*, volume 11, pages 196–201, February 2006.

[75] J. Lin, Y. Wu and T. S. Huang. Modeling the Constraints of Human Hand Motion. In *Proceedings of Workshop on Human Motion (HUMO'00)*, HUMO '00, 2000.

[76] J. Liu, J. Luo and M. Shah. Recognizing realistic actions from videos "in the wild". In *Computer IEEE Conference on Vision and Pattern Recognition, 2009. CVPR 2009.*, pages 1996 –2003, Jun. 2009.

[77] J. Malik and P. Perona. Preattentive Texture Discrimination with Early Vision Mechanisms. *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, 7(5):923–932, May 1990.

[78] J. Malik, S. Belongie, T. Leung and J. B. Shi. Contour and Texture Analysis for Image Segmentation. *International Journal of Computer Vision*, 43:7–27, June 2001.

[79] J. Shotton, J. Winn, C. Rother and A. Criminisi. TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context. *International Journal of Computer Vision*, 81:2–23, January 2009.

[80] J. Sullivan and S. Carlsson. Recognizing and Tracking Human Action. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *ECCV '02 Proceedings of the 7th European Conference on Computer Vision-Part I*, volume 2350 of *Lecture Notes in Computer Science*, pages 629–644. Springer Berlin Heidelberg, 2002.

[81] J. Winn, A. Criminisi and T. Minka. Object Categorization by Learned Universal Visual Dictionary. In *Proceeding of Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005.*, volume 2, pages 1800–1807 Vol. 2, October 2005.

[82] S.S. Beauchemin J.L. Barron, D.J. Fleet. Performance of optical flow techniques. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pages 236–242, June 1992.

[83] Thorsten Joachims, Thomas Finley, and Chun-Nam Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27–59, Oct 2009.

[84] J.T. Bao, A. Song, Y. Guo and H.R. Tang. Dynamic hand gesture recognition based on surf tracking. In *Electric Information and Control Engineering (ICEICE), 2011 International Conference*, pages 338 –341, Southeast University, Nanjing, China, april 2011.

[85] B. Julesz. Textons, The Elements of Texture Perception, and Their Interactions. *Nature*, 290(5802):91–97, March 1981.

[86] K. Grauman and T. Darrell. The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features. In *Proceeding of Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005.*, pages 1458–1465, Massachusetts Inst. of Technol., Cambridge, MA, 2005.

[87] K. Levi and Y. Weiss. Learning Object Detection From A Small Number of Examples: The Importance of Good Features. In *Proceedings of the 2004 IEEE computer society conference on Computer vision and pattern recognition*, CVPR'04, pages 53–60, Washington, DC, USA, 2004. IEEE Computer Society.

[88] K. Mikolajczyk and C. Schmid. Indexing Based on Scale Invariant Interest Points. In *Proceedings of Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001.*, pages 525–531, 2001.

[89] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. Van Gool. A Comparison of Affine Region Detectors. *International Journal of Computer Vision*, 65:43–72, November 2005.

[90] K. Schindler and L. van Gool. Action Snippets: How Many Frames Does Human Action Recognition Require? pages 1–8, June 2008.

[91] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004*, volume 2, pages II–506–513, 2004.

[92] Roman Klinger and Katrin Tomanek. Classical probabilistic models and conditional random fields. Technical Report TR07-2-013, Department of Computer Science, Dortmund University of Technology, Dec. 2007.

[93] P. Kohli and P.H.S. Torr. Efficiently solving dynamic markov random fields using graph cuts. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2 of *ICCV '05*, pages 922–929 Vol. 2, Washington, DC, USA, Oct. 2005. IEEE Computer Society.

[94] Pushmeet Kohli and Philip H. S. Torr. Measuring uncertainty in graph cut solutions. *Comput. Vis. Image Underst.*, 112:30–38, October 2008.

[95] V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts - a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(7):1274 –1279, July 2007.

[96] V. Kolmogorov and R. Zabin. What Energy Functions Can Be Minimized Via Graph Cuts? In *Proceeding of IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 26, pages 147 –159, Feb. 2004.

[97] F. Korč and W. Förstner. Approximate Parameter Learning in Conditional Random Fields: An Empirical Investigation. In *Proceedings of the 30th DAGM symposium on Pattern Recognition*, pages 11–20, Berlin, Heidelberg, 2008. Springer-Verlag.

[98] J.J. Kuch and T.S. Huang. Vision based hand modeling and tracking for virtual teleconferencing and telecollaboration. In *Proceedings of the Fifth International Conference on Computer Vision*, ICCV '95, pages 666–671, Jun 1995.

[99] S. Kumar and M. Hebert. Discriminative Random Fields. *Int. J. Comput. Vision*, 68:179–201, June 2006.

[100] Mathias Klsch and Matthew Turk. Fast 2d hand tracking with flocks of features and multi-cue integration. In *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*, volume 10, page 158, June 2004.

[101] L. Gorelick, M. Blank, E. Shechtman, M. Irani and R. Basri. Actions as Space-Time Shapes. In *Proceeding of Tenth IEEE International Conference on Computer Vision (ICCV), 2005*, volume 29, pages 2247–2253, December 2007.

[102] L. Gorelick, M. Galun, E. Sharon, R. Basri and A. Brandt. Shape Representation and Classification Using the Poisson Equation. In *Proceeding of IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 28, pages 1991–2005, 2006.

[103] L. H. Chen and C. Grecos. A Fast Skin Region Detector for Colour Images. In *Proceeding of IEE International Conference on Visual Information Engineering (VIE 2005)*, Glasgow, UK, 2005.

[104] L.-P. Morency, A. Quattoni and T. Darrell. Latent-Dynamic Discriminative Models for Continuous Gesture Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07.*, pages 1–8, 2007.

[105] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE on Readings in speech recognition*, pages 267–296, 1990.

[106] L. Shi. Local Influence in Principal Components Analysis. *Biometrika*, 84(1):175–186, 1997.

[107] L. Zelnik-Manor and M. Irani. Event-based Analysis of Video. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001.*, volume 2, pages II–123–130, 2001.

[108] I. Laptev. *Local Spatio-Temporal Image Features for Motion Interpretation.* PhD thesis, Department of Numerical Analysis and Computer Science (NADA), KTH, June 2004.

[109] I. Laptev and T. Lindeberg. Space-time interest points. volume 1, pages 432–439, Nice, France, Oct. 2003.

[110] E. Larson and E. Lusk. Evaluating handwashing technique. *Journal of Advanced Nursing*, 53(1):46–53, 2006.

[111] Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu-Jie Huang. A tutorial on energy-based learning. In G. Bakir, T. Hofman, B. Schölkopf, A. Smola, and B. Taskar, editors, *Predicting Structured Data*. MIT Press, 2006.

[112] S. Z. Li. *Markov random field modeling in computer vision*. Springer-Verlag, London, UK, 1995.

[113] T. Lindeberg. Feature Detection with Automatic Scale Selection. *International Journal of Computer Vision*, 30(2):79–116, November 1998.

[114] M. Ahmad and S. W. Lee. Hmm-based human action recognition using multiview image sequences. In *Proceedings of the 18th International Conference on Pattern Recognitin (ICPR'06)*, volume 1, pages 263–266, Korea University, Seoul, 2006.

[115] M Ángeles Mendoza and N Pérez De La Blanca. HMM-Based Action Recognition Using Contour Histograms. In *Pattern Recognition and Image Analysis*, volume 4477, pages 394–401, 2007.

[116] M. Li and B. Z. Yuan. 2D-LDA: A Statistical Linear Discriminant Analysis for Image Matrix. *Pattern Recogn. Lett.*, 26:527–532, April 2005.

[117] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna and T. Poggio. Pedestrian Detection Using Wavelet Templates. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997.*, pages 193–199, 1997.

[118] M. Szummer, P. Kohli and D. Hoiem. Learning CRFs Using Graph Cuts. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 582–595, Berlin, Heidelberg, 2008. Springer-Verlag.

[119] A.M. Martinez and A.C. Kak. Pca versus lda. volume 23, pages 228 –233, Feb. 2001.

[120] David S. Moore and George P. Mccabe. Bootstrap methods and permutation tests. In *Introduction to the Practice of Statistics*, chapter 14. W. H. Freeman and Company, New York, 5 edition.

[121] N. Dalai and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.*, volume 1, pages 886–893, 2005.

[122] N. Dalal. *Finding People in Images and Videos.* PhD thesis, Institut National Polytechnique de Grenoble, July 2006.

[123] N. Dalal, B. Triggs and C. Schmid. Human Detection Using Oriented Histograms of Flow and Appearance. In A. Leonardis, H. Bischof, and A. Prinz, editors, *Proceeding of European Conference on Computer Vision (ECCV) 2006*, volume 2, pages 428–441. Springer-Verlag Berlin Heidelberg, 2006.

[124] Hartmut Neven, Vasil S. Denchev, Geordie Rose, and William G. Macready. Training a large scale classifier with the quantum adiabatic algorithm. *CoRR*, abs/0912.0779, 2009.

[125] J. C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE Computer Society, 2007.

[126] O. Boiman and M. Irani. Detecting Irregualrities in Images and in Video. In *Proceeding of Tenth IEEE International Conference on Computer Vision, 2005.*, volume 1 of *ICCV 2005*, pages 462–469, 2005.

[127] E Osuna, R Freund, and F Girosi. An improved training algorithm for support vector machines. In J. Principe, L. Gile, N. Morgan, and E. Wilson, editors, *Neural Networks for Signal Processing VII — Proceedings of the 1997 IEEE Workshop*, pages 276–285, New York, 1997. IEEE.

[128] P. Felzenszwalb, D. McAllester and D. Ramanan. A Discriminatively Trained, Multiscale, Deformable Part Model. In *Proceeding of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 2008,*, University of Chicago, Chicago, IL, June 2008.

[129] P. Garg, N. Aggarwal and S. Sofat. Vision Based Hand Gesture Recognition. In *Proceeding of World Academy of Science Engineering and Technology*, volume 49, pages 972–977, February 2009.

[130] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. In *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, volume 19, pages 711 –720, Yale University, New Haven, CT, Jul. 1997.

[131] P. Sabzmeydani and G. Mori. Detecting Pedestrians by Learning Shapelet Features. In *Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[132] P. Scovanner, S. Ali and M. Shah. A 3-Dimensional Sift Descriptor and Its Application to Action Recognition. *MULTIMEDIA '07 Proceedings of the 15th international conference on Multimedia*, pages 357–360, 2007.

[133] P. Viola, M. J. Jones and D. Snow. Detecting Pedestrians Using Patterns of Motion and Appearance. In *Proceedings of Ninth IEEE International Conference on Computer Vision, 2003.*, volume 63, pages 153–161. Kluwer Academic Publishers, July 2005.

[134] Didier Pittet, Benedetta Allegranzi, Hugo Sax, Sasi Dharan, Carmem Lúcia Pessoa-Silva, Liam Donaldson, and John M Boyce. Evidence-based model for hand transmission during patient care and the role of improved practices. *The Lancet Infectious Diseases*, 6(10):641–652, 2006.

[135] John C. Platt. Fast training of support vector machines using sequential minimal optimization. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in kernel methods*, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

[136] Q. Zhu, M-C. Yeh, K-T. Cheng, S. Avidan. Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In *Proceeding of 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1491–1498. IEEE Computer Society, 2006.

[137] S. Ali, A. Basharat and M. Shah. Chaotic Invariants for Human Action Recognition. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference*, pages 1–8, Univ. of Central Florida, Orlando, October 2007.

[138] S. B. Chen, Y. Hu, B. Luo and R. H. Wang. Heteroscedastic Discriminant Analysis with Two-Dimensional Constraints. In *Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008.*, pages 4701 –4704, China Sci. and Technol. University, Hefei, Apr. 2008.

[139] S. B. Wang; A. Quattoni, L.-P. Morency, D. Demirdjian and T. Darrell. Hidden Conditional Random Fields for Gesture Recognition. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1521–1527, Washington, DC, USA, 2006. IEEE Computer Society.

[140] S. Belongie, J. Malik and J. Puzicha. Shape Context: A New Descriptor for Shape Matching and Object Recognition. In *Proceeding of Pattern Analysis and Machine Intelligence, IEEE Transactions on*, volume 24, pages 831–837, California University, San Diego, La Jolla, CA, 2000.

[141] S. Kumar, J. August and M. Hebert. Exploiting Inference for Approximate Parameter Learning in Discriminative Fields: An Empirical Study. In *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, volume 3757, pages 1–16, 2005.

[142] S. Mitra and T. Acharya. Gesture Recognition: A Survey. In *Proceeding of IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* , volume 37, pages 311–324, 2007.

[143] Rigbe Samuel, Astier M Almedom, Giotom Hagos, Stephanie Albin, and Alice Mutungi. Promotion of handwashing as a measure of quality of care and prevention of hospital-acquired infections in eritrea: the keren study. *African Journal of Health Sciences*, Vol.5(1):4–13, 2005.

[144] K. Sato and J. K. Aggarwal. Tracking and Recognizing Two-Person Interactions in Outdoor Image Sequences. In *Proceeding of 2001 IEEE Workshop on Multi-Object Tracking*, pages 87–94, 2001.

[145] H. Sax, B. Allegranzi, I. Ukay, E. Larson, J. Boyce, and D. Pittet. My five moments for hand hygiene: a user-centred design approach to understand, train, monitor and report hand hygiene. *Journal of Hospital Infection*, 67(1):9–21, 2007.

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

[147] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[148] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470 –1477 vol.2, Oct. 2003.

[149] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 2:246–252, 1999.

[150] Moritz Strring, Hans Jrgen Andersen, and Erik Granum. Physics-based modelling of human skin colour under mixed illuminants. *Robotics and Autonomous Systems*, 35(3-4):131–142, 2001.

[151] Charles Sutton and Andrew McCallum. An Introduction to Conditional Random Fields for Relational Learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2006.

[152] T. Coogan, G. Awad, J. Han and A. Sutherland. Real Time Hand Gesture Recognition Including Hand Segmentation and Tracking. In *Proceeding of Advances in Visual Computing*, volume 1, pages 495–504, 2006.

[153] T. Leung and J. Malik. Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons. *Int. J. Comput. Vision*, 43(1):29–44, June 2001.

[154] T. Lindeberg. Scale-Space: A Framework for Handling Image Structures at Multiple Scales. In *Cern European Organization for Nuclear Researchreportscern (1996)* , pages 8–21, September 1996.

[155] T. Mauthner, P. M. Roth and H. Bischof. Instant Action Recognition. In *Proceeding of SCIA '09 Proceedings of the 16th Scandinavian Conference on Image Analysis*, pages 1–10, Berlin, Heidelberg, 2009. Springer-Verlag.

[156] A. Treisman. Features and objects in visual processing. *Scientific American*, 255:114–125, November 1986.

[157] The Tran Truyen. *On conditional random fields: Applications, feature selection, parameter estimation and hierarchical modelling.* PhD thesis, Curtin University, Australia, 2008.

[158] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, volume 36 of *ICML '04*, page 104, New York, NY, USA, 2004. ACM.

[159] M. Tuceryan and A. K. Jain. Texture analysis. In *The Handbook of Pattern Recognition and Computer Vision*, chapter 2, pages 207–248. World Scientific Publishing Co., second edition, 1998.

[160] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk and B. Girod. CHoG: Compressed Histogram of Gradients A Low Bit-rate Feature Descriptor. volume 96, pages 384–399, Hingham, MA, USA, February 2012. Kluwer Academic Publishers.

[161] V. Vezhnevets, V. Sazonov and A. Andreeva. A Survey on Pixel-Based Skin Color Detection Techniques. In *Proceedings of the Cybernetics 2003*, volume 85, pages 85–92, 2003.

[162] Douglas L. Vail, Manuela M. Veloso, and John D. Lafferty. Conditional random fields for activity recognition. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, AAMAS '07, pages 235:1–235:8, New York, NY, USA, 2007. ACM.

[163] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511–I–518, Hawaii, 2001.

[164] Christian Vogler and Dimitris Metaxas. A framework for recognizing the simultaneous aspects of american sign language. *Computer Vision and Image Understanding*, 81:358–384, 2001.

[165] W. T. Freeman and M. Roth. Orientation Histograms for Hand Gesture Recognition. In *Proceeding of International Workshop on Automatic Face and Gesture Recognition*, pages 296–301, 1994.

[166] W. Zheng and L. H. Liang. Fast Car Detection Using Image Strip Features. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2703–2710, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[167] Larry Wasserman. *All of Statistics: A Concise Course in Statistical Inference (Springer Texts in Statistics)*. Springer, December 2003.

[168] M. Welling. Fisher Linear Discriminant Analysis. Technical report, Department of Computer Science, University of Toronto, 2005.

[169] AF Widmer and Infection control team control team. Andreas f widmer for the basel infection control team. *BMC Proceedings*, 5(6):1, 2011.

[170] World Health Organization. WHO Guidelines on Hand Hygiene in Health Care. *WHO*, 2009.

[171] Y. Freund and R. Schapire. A short introduction to boosting. In *Proceeding of Japonese Society for Artificial Intelligence.*, volume 14, pages 771–780, 1999.

[172] Y. Wu and T.S. Huang. Hand modeling, analysis and recognition. In *Signal Processing Magazine, IEEE*, volume 18, pages 51–60, May 2001.

[173] Chen Yanover and Yair Weiss. Finding the m most probable configurations using loopy belief propagation. *Advances in Neural Information Processing Systems 16*, 16:289, 2004.

[174] Pei Yin, A. Criminisi, J. Winn, and I.A. Essa. Bilayer segmentation of webcam videos using tree-based classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(1):30–42, Jan. 2011.

[175] Tsz-Ho Yu, Tae-Kyun Kim, and Roberto Cipolla. Real-time action recognition by spatiotemporal semantic and structural forest. In *Proceedings of the British Machine Vision Conference*, pages 52.1–52.12. BMVA Press, 2010.

[176] Chong-Wah Ngo Yu-Gang Jiang and Jun Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. CIVR '07, pages 494–501. ACM, 2007.

[177] Z. Ghahramani. Learning Dynamic Bayesian Networks. In *Adaptive Processing of Sequences and Data Structures, International Summer School on Neural Networks, "E.R. Caianiello"-Tutorial Lectures*, pages 168–197, London, UK, 1998. Springer-Verlag.

[178] J. Zhang, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, June 2007.