

VIDEO-BASED FACIAL EXPRESSION ANALYSIS

BY ZHIGUO LI

A dissertation submitted to the
Graduate School—New Brunswick
Rutgers, The State University of New Jersey
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
Graduate Program in Computer Science

Written under the direction of

Dimitris N. Metaxas

and approved by

New Brunswick, New Jersey

October, 2010

ABSTRACT OF THE DISSERTATION

Video-based Facial Expression Analysis

by Zhiguo Li

Dissertation Director: Dimitris N. Metaxas

Recognizing facial expressions from facial video sequences is an important and unsolved problem. Among many factors that contribute to the challenges of this task are: non-frontal facial poses, poorly aligned face images, large variations in the temporal scale of facial expressions, and the subtle differences between different subjects for the same facial expression etc. A successful video-based facial expression analysis system should be able to handle at least the following problems: robust face tracking, or spatial alignment of the faces, video segmentation, effective feature representation and selection schemes which are robust to face mis-alignment and temporal normalization by sequential classifier. In this work we report several advances we made in building various components of a system for classifying facial expressions from video inputs. Particularly, my work focus on robust face tracking, facial feature representation and selection under different face alignment conditions, sequential modeling for facial expression recognition. We performed extensive experiments using the proposed algorithms on publicly available dataset and achieved state of the art performances.

Acknowledgements

I want to thank Professor Dimitris Metaxas for his advice, encouragement, trust, and support over my Ph.D. years. He has been an excellent advisor and always directed me toward doing fundamental research that will make a difference. He also encouraged me pursue independent work. None of the work in this thesis would have happened without him.

I would like to thank the other members of my doctoral committee: Prof. Vladimir Pavlovic, Prof. Ahmed Elgammal and Prof. Chandra Kambhamettu for their advice, help and valuable suggestions regarding this thesis. It is a privilege for me to have each of them serve in my committee.

I also want to thank Dr. Qingshan Liu for his insightful discussions with me. Special thanks to Dr. Christian Vogler. His excellent skills on system design and programming helped and inspired me a lot.

Last but not least, special thanks to my friends and colleagues from the Center for Computational Biomedicine Imaging and Modeling (CBIM). I learned a lot from discussions with them. Their friendship and help made my life at Rutgers a real pleasure.

Table of Contents

Abstract	ii
Acknowledgements	iii
List of Tables	vii
List of Figures	viii
1. Introduction	1
1.1. Motivation	2
1.1.1. Face Tracking	3
1.1.2. Facial Feature Representation and Selection	5
1.1.3. Sequential Facial Expression Modeling	7
1.2. Contributions	9
1.3. Thesis Outline	13
2. Overview of Related Work	14
2.1. Face Tracking	15
2.2. Facial Feature Representation and Selection	18
2.2.1. Facial Features for Poorly Aligned Images	18
2.2.2. Facial Features for Well Aligned Images	19
2.3. Sequential Facial Expression Modeling	23
3. Face Tracking	27
3.1. Overview of 3D Deformable Models	28

3.2.	Overview of Active Shape Model	30
3.3.	ASM and 3D Deformable Model Integration	32
3.3.1.	Overview of Combined Tracking Algorithm	33
3.3.2.	Explanation of the Tracking Algorithm	35
3.3.3.	Statistical Inference: Unsupervised Meets Supervised	36
3.4.	Experimental Results	38
3.4.1.	Automated Calibration and Model Adaptation	38
3.4.2.	Statistical Inference with 3D models and ASMs	39
4.	Facial Feature Representation and Selection	41
4.1.	With Poor Face Alignment	41
4.1.1.	Multiple-Instance based Feature Selection	42
	Motivation	42
	Approximating the Constrained K-minimum Spanning Tree	43
	Testing Procedure	46
4.1.2.	Experimental Results	46
	Testing with Well-aligned Training Data	47
	Testing with Noisy Training Data	48
4.1.3.	Conclusions and Discussions	54
4.2.	With Good Face Alignment	55
4.2.1.	Representative Features and Feature Selection Methods	56
	Haar-like Features	56
	Gabor Features	57
	LBP Features	59
4.2.2.	Feature Selection	60
	Bayesian Multinomial Logistic Regression	60

Boosting for Feature Selection	61
Feature Selection Based on Mutual Information	64
4.2.3. Comparison of Different Facial Features and Feature Selection	
Methods	64
5. Sequential Facial Expression Modeling	67
5.1. Background on Graphical Models	69
5.1.1. Inference	71
Exact Inference	71
Approximate Inference	73
5.2. Facial Feature Representation	74
5.3. Sequential Data Classification algorithms	76
5.3.1. The problem: a unified view	76
5.3.2. Discriminative Sequence Classification - HCRF	80
5.3.3. Generative Sequence Classification - HMM-C	81
5.3.4. Discriminative Sequence Tagging - CRF	81
5.3.5. Generative Sequence Tagging - HMM-T	82
5.3.6. Discriminative Frame Classification - MNLR	84
5.3.7. Generative Frame Classification - Naive Bayes	84
5.4. Experimental Results	85
5.4.1. Observations and Analysis	90
5.5. Conclusions and Discussions	94
6. Conclusions and Future Research Direction	96
References	103
Vita	112

List of Tables

4.1. Testing combinations for aligned training data	47
4.2. Results comparison	48
4.3. Testing combination for noisy training data	49
4.4. Results comparison	49
4.5. Confusion matrices for different training bases. The traces for the matrices are 300 (82.2%),305 (83.6%),284 (77.8%) and 279 (76.4%)	53
4.6. Confusion matrices for multiple instance selected noisy bag images, trace for the matrix 298 (81.6%)	54
4.7. Comparison of different features for expression recognition	65
4.8. Comparison of different features selection methods for face recognition (Haar feature)	65
5.1. Naive Bayes, HMM Tagging, HMM Classification, the traces for the matrices are 311 (85.21%),325 (89.04%),326 (89.32%)	86
5.2. MNLR, CRF and HCRF, the traces for the matrices are 320 (87.7%), 336 (92.1%), 354 (97.0%)	87
5.3. Overall recognition rate for different classifiers	87
5.4. Comparison of our results with other published results	89

List of Figures

1.1. Six prototypical emotions [55]	2
1.2. Example upper face AUs and combinations [71]	3
1.3. Example lower face AUs and combinations [71]	4
1.4. Video-based facial expression analysis framework	4
1.5. Recognition Rate Change for FisherFace w.r.t Rotation 1.5a, Scale 1.5b and Translation 1.5c Perturbations	6
3.1. Example multi-view ASM Shapes	30
3.2. Example multi-view ASM search results	31
3.3. ASM to 3D model correspondences. The thick red lines on the right show the 3D model nodes that correspond to the ASM nodes on the left.	34
3.4. 3D model-guided ASM initialization. 3.4a: tracked model position from previous frame. 3.4b: initial ASM guess in next frame, based on model position from previous frame. 3.4c: final ASM shape in next frame.	36
3.5. Automated model fitting and adaptation. Left to right: Before model fitting, ASM feature detection, Adjustment of rigid model parameters, Final fit after adjustment of nonrigid shape parameters.	38
3.6. Tracking example where ASMs encounter difficulties, and are only par- tially reliable. The ASMs shown in this figure have never been trained on this subject. Left to right: Original frame, Tracked 3D model, ASM search result, Statistical rejection of unreliable ASM features, using pro- jected regions of confidence (in red, blue points are accepted).	40

3.7. Results of sign language tracking sequence using ASM, KLT and edge tracking.	40
4.1. Bags of Face Images. From the original not precisely cropped input image, we generate more samples by adding perturbations. High density area in the subspace represents better face samples.	42
4.2. Bag Distances Map	45
4.3. Recognition Rate Change for FisherFace w.r.t single aligned training base 4.3a, aligned bag training base 4.3b	50
4.4. Single Noisy Training Base	50
4.5. Recognition Rate Change for FisherFace w.r.t Single aligned gallery, single aligned probe 4.5a, Single aligned gallery, single noisy probe 4.5b . .	51
4.6. Aligned bag gallery, noisy bag probe	52
4.7. Example Haar-like feature	57
4.8. Example Gabor filters with 4 orientations and 4 different scale	58
4.9. Example facial expression image before 4.9a and after 4.9b convolution with Gabor filters	59
4.10. Image processing with LBP operator	60
4.11. Example selected Haar-like features by AdaBoost	63
4.12. ROC curves for major AUs on NSF dataset	66
5.1. Frame-wise Classification	70
5.2. sequence tagging	71
5.3. Regular grid on facial images	75
5.4. Histogram of mean flow for different expressions	75
5.5. Graphical model for sequence classification	76
5.6. Convergence Rate for HMM, CRF and HCRF	87

5.7. Error Rates vs. PCA Dimension Reduction	88
5.8. Error Rates with Dynamic Feature vs. Number of Frames Used from a Sequence	88
5.9. Error Rates with Multiple Instance Feature vs. Number of Frames Used from a Sequence	89

Chapter 1

Introduction

Human face is a major communication channel between different subjects. This system conveys information via four general classes of signals [34]: (1) static facial signals represent relatively permanent features of the face, such as the bony structure and soft tissues masses, that contribute to an individual's appearance; (2) slow facial signals represent changes in the appearance of the face that occur gradually over time, such as the development of permanent wrinkles and changes in skin texture; (3) artificial signals represent exogenously determined features of the face, such as eyeglasses; and (4) rapid facial signals that lead to visually detectable changes in facial appearance. In this work, we mainly focus on the rapid facial signals for expression analysis. These signals typically are brief, mostly lasting $250\text{ ms} \sim 5\text{ s}$ [37]. With a typical video camera capturing facial images at 30fps, this translates into $7 \sim 150$ frames.

Since the early 1970s, Paul Ekman et al. [35] have performed extensive studies of human facial expressions and found evidence to support universality in facial expressions. They proposed six prototypical emotions: *happiness*, *sadness*, *fear*, *disgust*, *surprise* and *anger*. A group of the six expressions is shown in figure 1.1. Paul Ekman et al. [36] further developed the Facial Action Coding System (FACS). The FACS was developed by determining how the contraction of each facial muscle (singly and in combination with other muscles) changes the appearance of the face. Measurement with FACS is done in terms of Action Units (AU). There are 46 AUs which account for changes in facial expression, and 12 AUs which more grossly describe changes in gaze

direction and head orientation. Figures 1.2 and figure 1.3 show some example AUs on the upper face region and the lower face region respectively. In this work, we focus on the six universal expressions and also report some results on the several important AUs.

Facial expression analysis as a research topic are started by behavioral scientists and psychologists [29]. Since the early attempt to build an automatic system [115] for expression analysis, facial analysis have been an active research topic in computer vision research and much progress have been made toward building an automatic, robust and fast system for recognizing facial expressions [13, 39, 23, 33, 71, 140, 121, 137, 19]. Facial expression analysis has a wide area of application domains, including multi-modal human computer interaction, lie detection, paralinguistic communication and psychiatry.



Figure 1.1: Six prototypical emotions [55]

1.1 Motivation

A successful video-based facial expression analysis system should be able to handle at least the following problems: robust face tracking, or spatial alignment of the faces, video segmentation, effective feature representation and selection schemes under different face alignment conditions and temporal normalization by sequential classifier. In this work our objective is to recognize facial expressions from video sequences. My work focus on robust face tracking, facial feature representation and selection under different
















<i>NEUTRAL</i>	AU 1	AU 2	AU 4	AU 5
				
Eyes, brow, and cheek are relaxed.	Inner portion of the brows is raised.	Outer portion of the brows is raised.	Brows lowered and drawn together	Upper eyelids are raised.
AU 6	AU 7	AU 1+2	AU 1+4	AU 4+5
				
Cheeks are raised.	Lower eyelids are raised.	Inner and outer portions of the brows are raised.	Medial portion of the brows is raised and pulled together.	Brows lowered and drawn together and upper eyelids are raised.
AU 1+2+4	AU 1+2+5	AU 1+6	AU 6+7	AU 1+2+5+6+7
				
Brows are pulled together and upward.	Brows and upper eyelids are raised.	Inner portion of brows and cheeks are raised.	Lower eyelids and cheeks are raised.	Brows, eyelids, and cheeks are raised.

Figure 1.2: Example upper face AUs and combinations [71]

face alignment conditions, sequential modeling for facial expression recognition. The relation between the different modules of the system is shown in figure 1.4

1.1.1 Face Tracking

Reliable face detection and face tracking is the first component of the video-based facial expression system. 3D deformable model proves to be an effective approach for face tracking [31, 32, 30, 47, 48, 129]. Parametrized 3D deformable models rely on the correct estimation of image features that are then used to obtain good estimates for each of the model's parameters. Most existing methods for image feature extraction are based on deterministic feature tracking (e.g KLT) or tracking feature distributions [52, 47, 133]. However, these methods are not based on learning and rely on local computations for the tracking of each feature. If the assumptions that underlie the feature extraction methods are invalid, because of factors such as changes in illumination, noise or occlusions, the








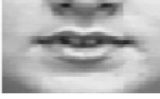


<i>NEUTRAL</i>	AU 9	AU 10	AU 12	AU 20
 Lips relaxed and closed.	 The infraorbital triangle and center of the upper lip are pulled upwards. Nasal root wrinkling is present.	 The infraorbital triangle is pushed upwards. Upper lip is raised. Causes angular bend in shape of upper lip. Nasal root wrinkle is absent.	 Lip corners are pulled obliquely.	 The lips and the lower portion of the nasolabial furrow are pulled pulled back laterally. The mouth is elongated.
AU15	AU 17	AU 25	AU 26	AU 27
 The corners of the lips are pulled down.	 The chin boss is pushed upwards.	 Lips are relaxed and parted.	 Lips are relaxed and parted; mandible is lowered.	 Mouth stretched open and the mandible pulled downwards.

Figure 1.3: Example lower face AUs and combinations [71]

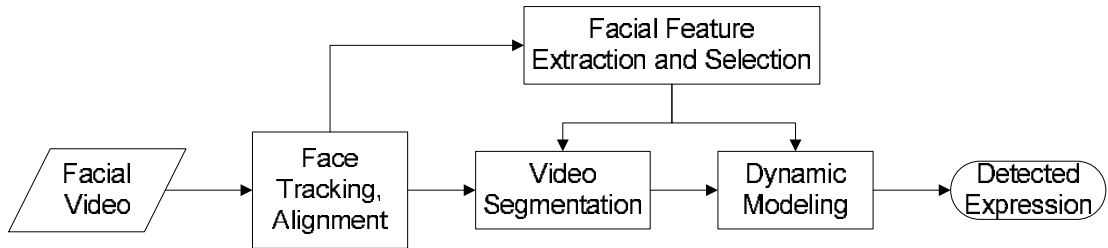


Figure 1.4: Video-based facial expression analysis framework

feature correspondences from frame to frame are not computed correctly. Such errors in turn cause the 3D model to drift over time, and results in incorrect tracking of the video.

Active Shape Models (ASMs), on the other hand, is a good combination of both discriminative and generative approach for face alignment, and it could act as a better cue to drive the 3D deformable model. It's a generative approach because it's based on learning a point distribution model for normalized facial shapes, and tend to give more reliable results than feature trackers for poses that match closely the ones on which

they have been trained. They also are less vulnerable to drifting, because the method is also discriminative, in the sense that it locates each landmark points on the facial shape using cues directly in the image. This latter ability is invaluable in recovering from occlusions or other situations that would cause a non-learning based system to lose track. Conversely, ASMs can fail spectacularly when they encounter a situation for which they were not trained for or get stuck to local minimum when the initialization is too far from target location. ASMs are also linear models in principle, while face shape variations lie in non-linear space. However, multiple local linear ASMs can approximate this non-linear space.

Parametrized 3D deformable models [83, 31, 32, 30] allow us to perform unsupervised techniques on the data driving the tracking procedure, such as parameter-space outlier rejection [129], occlusion handling [46], and statistical inference to feed a Bayesian Filter such as the Kalman Filter [48]. Additionally, the information provided by a 3D deformable model is a relatively low-dimensional parameter vector, where information describing motion and deformation is already parsed in some meaningful way — making recognition and analysis considerably easier.

We propose a hypothesis that the tight integration of the model based cue, such as the ASM, to the deformable model framework would improve 3D deformable model face tracking greatly.

1.1.2 Facial Feature Representation and Selection

Even under very good lighting condition, the tracking results will still tend to be prone to errors due to large pose change and occlusions etc, as shown in the face tracking section. For most face recognition and facial expression analysis tasks, the tracking results will not be accurate enough to crop and align the face image for further analysis.

Accurate face alignment is critical to the performance of both appearance-based

and geometric feature-based facial analysis. However, current feature extraction techniques are still not reliable or accurate enough. It is unrealistic to expect localization algorithms to always get very accurate results under very different lighting, pose and expression conditions. To get better recognition rate, we need to improve the robustness of existing recognition algorithms.

To illustrate the effect of the face alignment error on face recognition performance, we use the FERET face database [100] with ground truth alignment information available. We intentionally add some perturbations to the ground truth. Perturbations are added by moving the left center and right eye center ground truth with some random pixels.

Figure 1.5 shows that the rotation perturbation affects the recognition performance most, and the translation perturbation has the smallest effects. Overall, we can see that even small perturbations could reduce the recognition rate significantly.

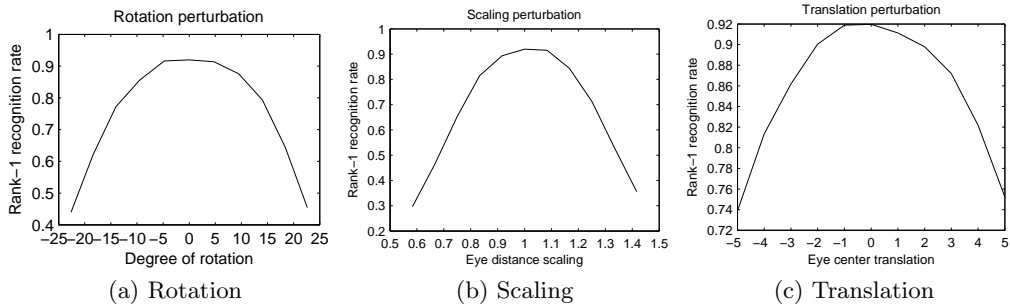


Figure 1.5: Recognition Rate Change for FisherFace w.r.t Rotation 1.5a, Scale 1.5b and Translation 1.5c Perturbations

One intuitive way to make classifiers robust to image alignment errors is to augment the training sets by adding random perturbations to the training images. By adding noisy but identifiable versions of given examples, we can expand our training data and improve the robustness of the feature extraction against a small amount of noise in the input. The augmented training set can model the small image alignment errors. The other way is to add perturbations to the probe images during the testing stage. Adding

perturbations to the training set requires that we know the ground truth before hand.

In multiple-instance learning algorithms, the task is to learn a classifier given positive and negative bags of instances. Each bag may contain many instances. A bag is labeled positive if at least one of the instance in it is positive. A bag is labeled negative only if all instances in it are negative. To get better feature representations from poorly aligned face images, we formulate the process as a multiple-instance learning problem: we take the whole image as a bag, and all possible sub-windows within it as instances. If an image contains a face, then we label this image as a positive bag, since we know that there is at least a sub-window containing the face, but we don't know where exactly that sub-window is.

We propose a multiple instance based facial feature representation and extraction algorithm using poorly aligned facial images, for example, cropped images from face tracking procedure. We hypothesize that methods used previously to augment training and probe set by blindly adding random perturbed images will not be effective if the perturbed images are too far away from a good pose to be identifiable. We propose a method which can select from the augmented set only those relatively well aligned and identifiable facial images, and thus improving performance over previous methods.

As in all pattern recognition tasks, good representation and selection of features are important to acquire better recognition rates. We also empirically studied various feature representations and feature selection methods for expression recognition in this work.

1.1.3 Sequential Facial Expression Modeling

While posed exaggerate expressions could be easily recognized from a single snapshot of a face, the ability to discriminate more subtle expressions normally requires a comparison over time as the face changes shape and appearance. Even for exaggerated

expressions, the use of dynamic models could improve performance greatly. Typical facial actions normally have an onset, one or more peaks, and offsets, and the temporal organization of these events is critical to facial expression understanding and perception. Psychological studies [93] also suggest that temporal information are useful for expression analysis. The temporal pattern of changes in facial expression observed through motion analysis may carry important information. Similar problems are found in speech processing. It's well known that speaking rate variation causes nonlinear fluctuation in a speech pattern time axis [104]. Elimination of this fluctuation, or time-normalization, has been one of the central problems in spoken word recognition research. It is likely that previously proposed speech recognition methods such as hidden Markov models, discrete Kalman filters, recurrent neural networks, and dynamic time warping (DTW) will prove useful in the facial domain as well.

An intuitive way to classify facial expression sequences is to borrow techniques from time series study. Time series analysis is a major research topic in statistics and other research fields. For example, in linear dynamical system, the Kalman filtering scheme for prediction. Different from most time series studies, here we focus on sequential classification problem. Given a set of facial expression sequences (usually with different lengths), our goal is to learn a function that maps each input sequence to a label, i.e., one of the facial expression categories, and to predict the labels of new sequences using the learned function. Our focus here is to study effective feature representations and empirically study different sequential classification algorithms in dynamic expression modeling.

We hypothesize that the use of dynamic features such optical flow, and the local histogram of such dynamic features will make the classifier robust to small face alignment error while maintaining relative spatial structures among different facial regions.

Contrast to previous bag of words model, we also argue that elimination of temporal ordering information by histogramming in temporal domain will cause performance degradation. We hypothesize the coupling of temporally ordered dynamic features and sequential modeling algorithms will greatly improve recognition performance.

1.2 Contributions

Our objective in this work is to recognize facial expressions from video sequences. The problem comprises the following main subproblems: locating and tracking faces; facial feature representation and selection; video segmentation; sequential modeling for facial expression recognition. Specifically, a complete video-based facial expression recognition system should contain at least the following major components:

- Robust face detection and face tracking under various conditions, which is first module of the system. In a sense, the most important part and also the bottle neck of the video-based facial expression recognition system is the face tracking part. For example, if we can get accurate 3D deformable model face tracking results, then the other parts of the system could be very simple or even ignorable.
- Facial feature representation and selection. It's important to use features which are robust to face alignment errors due to the fact that current automated face tracking system cannot get perfect tracking results. As in all pattern recognition tasks, the extraction and selection of efficient features to feed in the classifier is critical for overall recognition rate. Large amount of irrelevant features could decrease classification rate.
- Video segmentation. Video segmentation has different meanings under different contexts. One explanation is spatial segmentation within each frame, which contrast to image segmentation in a single image. Another explanation for video

segmentation is temporal segmentation [61], where we divide raw video sequence into video subsegment based on different criteria. We mean the second explanation in our work. A closely related topic for temporal video segmentation is shot boundary detection [73]. Video segmentation could be done through classification [24, 73], however this renders the problem a chicken or the egg situation. Also note that facial expression video sequence segmentation is harder than boundary detection since the transition between different segments is gradual and more subtle.

- Sequence classifier which can model large temporal variations and incorporate temporal correlations in the sequence to improve performances. Ideally, different speed for the same facial action should not affect the recognition performance. Also as a baseline comparison, the sequence classifier should perform better than a majority voting scheme for the sequence.

Among the above mentioned components, this thesis focuses on the following modules: robust face tracking, facial feature representation and selection under different face alignment conditions, sequential modeling for facial expression recognition. The relations between the different modules of the system is shown in figure 1.4. The scope of the work is within these basic assumptions: we consider the facial expression recognition from video sequence a static environment without concept drift [124], the concept of which will be introduced later. We also assume that the videos have been segmented.

Based on the motivations and hypotheses stated above, this thesis makes the following contributions:

- We propose an algorithm to bring together the best of both worlds: the 3D deformable model and the 2D active shape model. We develop a framework for

robust 3D face tracking that combines the strengths of both 3D deformable models and ASMs. In particular we use ASMs to track reliably 2D image features in a video sequence. In order to track large rotations we train multiple ASMs that correspond to frontal and several side head poses. The 2D features from the ASMs are then used for parameter estimation of the 3D deformable model. After the 3D parameter-space outlier rejection and occlusion handling, the updated 3D model points are projected to 2D image to initialize the 2D ASM search, which becomes more robust due to the better model initialization. In addition we couple stochastically 3D deformable models and ASM by using the 3D pose of the deformable model to switch among the different ASMs, as well as to deal with occlusion. This integration allows the robust tracking of faces and the estimation of both their rigid and nonrigid motions. We demonstrate the strength of the framework in experiments that include automated 3D model fitting and facial expression tracking for a variety of applications including American Sign Language.

- We systematically investigate the effect of mis-aligned face images on face recognition systems. To make classifiers robust to the unavoidable face registration error, we formulate the facial feature representation and selection with poorly aligned face images as a multiple-instance learning task. We propose a novel multiple-instance based subspace learning scheme for facial feature representation and feature dimension reduction. In this algorithm, noisy training image bags are modeled as the mixture of Gaussians, and we introduce a method to iteratively select better subspace learning samples. Compared with previous methods, our algorithm does not require accurately aligned *training and testing* images, and can achieve the same or better performance as manually aligned faces for face recognition and facial expression recognition tasks. In this thesis, we used the

term *noisy images* to denote poorly aligned images. We also empirically studied various other feature representations and feature selection methods for face and expression recognition tasks. We also performed large scale facial action units recognition on large facial expression dataset with spontaneous facial expression and large head pose change.

- We are interested in the relationships between generative and discriminative models in both static and dynamic settings and especially their performances on the facial expression analysis problem. We propose to use a local histogram of optical flow feature representation which keeps both the spatial and temporal information; we show a unified view of different sequential data learning algorithms, both generative and discriminative, static and dynamic, and investigates their differences from various aspects such as their feature functions, objective functions and optimization strategies; we couple the proposed part-based feature representation with these algorithms for the expression recognition task. Our goal is not to find optimal learning structure and features representations for solving a specific object recognition task, but rather to fix on a particular feature set and use this as the basis to compare alternative learning methodologies and make observations. Even though we used relatively simple dynamic features and temporal models, we report some of the best expression recognition performances up to date on a publicly available data set. The observations we made from the large amount of experiments are one of the main contributions for this work. From our experimental results, sequential classification methods normally outperform sequential tagging methods and majority voting methods due to the flexibility of model structure, with the cost of longer learning time and the risk of overfitting. We also observed that either dynamic features alone or dynamic models alone are not enough for good performance. However, the coupling of simple dynamic feature

and temporal modeling improves performance a lot.

1.3 Thesis Outline

Chapter 1 gives an overview of our video-based facial expression recognition system, points out the motivations for this research, and lists thesis focus and main contributions.

Chapter 2 gives an overview of related work, and discusses the characteristics of various methods, their pros and cons.

Chapter 3 describes the face tracking module of our framework, where we combines the best of both models: ASMs and 3D deformable models, for robust face tracking.

Chapter 4 presents a multiple-instance based learning scheme for feature representation and selection with poorly aligned face images, and used the face recognition and facial expression recognition as examples to show that our proposed algorithm can improve recognition performance even under bad alignment condition. Chapter 4 also compares different feature representations and feature selection schemes for well aligned facial images.

Chapter 5 systematically examine the performances of static, dynamic, generative and discriminative methods for recognizing facial expressions.

At the end of this thesis, in chapter 6, we give conclusions and possible future research directions.

Chapter 2

Overview of Related Work

Facial expressions are one of the most powerful means for human beings to communicate and express emotions and intentions. Facial expression analysis as a research topic are started by behavioral scientists and psychologists [29]. Since the early attempt to build an automatic system [115] for expression analysis, facial analysis have been an active research topic in computer vision research and much progress have been made toward building an automatic, robust and fast system for recognizing facial expressions [13, 39, 23, 33, 71, 140, 121, 137, 19]. Research advancements in the related areas, such as face detection and face tracking have fostered the research interests in expression analysis. There are several reviews [33] [94] [41] that provide a comprehensive overview on the research work in this area. Pantic et al. [94] conducted comparative study on facial expression analysis from many different aspects, including the use of single or multiple images, different feature extraction schemes, classification methods, etc. Donato et al. [33] compared various techniques for automatically recognizing facial actions in sequences of images, mainly from the feature extraction and selection perspective. They classified existing features for recognition into three categories: optical flow, global and local spatial features. Fasel [41] reviewed various methods on face acquisition, face feature extraction and facial expression classification. Facial expression analysis has a wide area of application domains, including multi-modal human computer interaction, lie detection, paralinguistic communication and psychiatry. In the following sections, we list relative research work on each individual component of

our video-based facial expression recognition system. Being a popular research topic in computer vision, there are easily hundreds even thousands of papers on facial analysis. In this thesis we will include the most related work in the literature.

2.1 Face Tracking

Deformable models [118, 57, 83, 82] are one of the most popular method for image segmentation and shape manipulation. Particularly, deformable models have been applied for face tracking and facial animation [119, 69, 31, 32, 15, 30, 47].

Due to the fact that the information provided by a 3D deformable shape model is a relatively low-dimensional parameter vector and there have been established research on parameter fitting for those models, they are very popular for face analysis purposes. Parametrized 3D deformable models also allow us to perform parameter-space outlier rejection [129], occlusion handling [46], and statistical inference to feed a Bayesian Filter such as the Kalman Filter [48]. The basic procedure to update model parameters needs correct estimation of image features. Most existing methods for image feature extraction are based on deterministic feature tracking (e.g KLT) or tracking feature distributions [52, 47, 133]. However, these methods are not based on learning and rely on local computations for the tracking of each feature. If the assumptions that underlie the feature extraction methods are invalid, because of factors such as changes in illumination, noise or occlusions, the feature correspondences from frame to frame are not computed correctly. Such errors in turn cause the 3D model to drift over time, and results in incorrect tracking of the video.

2D deformable shape models have also been developed for face alignment and face tracking. Most popular methods include the ASM [25] and the Active Appearance Model (AAM) [27]. The ASM was originally developed for medical image segmentation

application, and it has been later extended to other area of applications. The segmentation algorithm could easily be extended to tracking scenario by combining point trackers, such as the KLT, with the ASM. The ASM are statistical models of the shape of objects which are constructed from a set of labelled training images. During search, the ASM only allows the shape to vary in ways seen similar to those in a training set. The shape of an object is represented by a set of points (controlled by the shape model). The AAM, on the other hand, can model both the shape and the texture variations seen in a training set. The main differences between the ASM and AAM is that ASM only uses textures perpendicular to the contour around each landmark points, while the AAM uses the texture of the whole image region. During search, the ASM searches along the profiles that are normal to the boundary, while AAM only samples the image texture under the current model position. As shown in [26], the ASM is faster and achieves more accurate feature point location than the AAM, but the AAM gives a better match to the texture. The ASM and AAM both use gradient based parameter updating algorithm, which render them sensitive to initial model parameters and they are very easy to get stuck to local minimum when the initialization is far from ground truth.

Matthews et al. [81] compared 2D with 3D deformable model for face modeling. The 2D and 3D model they compared are learning based models, which are acquired from either 2D images or 3D range scans. Representative 2D and 3D face models are the AAM [27] and the 3D morphable model [15] respectively. They compared the representational power, construction and real-time fitting of 2D and 3D models. From representational power perspective, although 2D model can be used to model 3D faces, the parameterization is not as compact as in 3D model, and 2D model could generate unrealistic face instances. More parameters means the fitting process could be more prone to local minimum. From the construction perspective, 3D models could be

constructed from 2D model and vice versa. However, the quality of 3D models built from 3D range scans are better than those built from 2D images. From the real-time fitting perspective, 2D real-time fitting is possible through inverse compositional algorithm, and 3D model fitting could also be more robust and converge in fewer iterations than 2D model fitting, thus confirming that 3D model is a more natural parameterization of faces. Their overall conclusion is that the 3D parameterization is more compact, more natural, 3D model fitting is more robust and requires fewer iterations to converge, and 3D occlusion reasoning is more powerful.

There have been increasing efforts on building combined 2D and 3D face tracking system. DeCarlo et al. [31] proposed a method for integrating the optical flow constraints as a cue among other cues into the framework of deformable model face tracking by using the method of Lagrange multiplier, and they showed that the integrated system can yield a great improvement in the ability to estimate shape and motion. However, optical flow is relatively unreliable for texture scarce regions, which is normally the case for face tracking problems, which in turn could cause the drifting of the 3D face model over time. It is also difficult for the algorithm to track large head pose change. Xiao et al. [134] proposed an extension to the 2D AAM by including the 3D shape model. This process introduced more parameters and it makes fitting process more difficult. Sung et al. [113] proposes to combine the AAM with the Cylinder Head Model (CHM). They argue that the combined system is more robust to pose with a higher tracking rate and wider pose coverage than 2D. Mase et al. [79] are perhaps the earliest efforts to track action units through optical flow.

2.2 Facial Feature Representation and Selection

2.2.1 Facial Features for Poorly Aligned Images

With current techniques for face tracking and face alignment, the accuracy is limited and the cropped facial images based on alignment results is far from perfect. Researchers have been trying to overcome the sensitivity of feature representations to imprecisely localized face images. For example, In Martinez [78], they pointed out that classical way of solving the face recognition problem is by using large and representative databases. However, in many applications only one sample per class is available to the system. They proposed a probabilistic method to learn the subspace that represents the error for each of training images. In addition to solving the imprecisely localized face problem, they also proposed methods for solving partially occluded and expression variant face images. Shan et al. [109] studied the curse of mis-alignment problem, and for each training image they generated several perturbed images to augment the training set and thus modeling the mis-alignment errors.

Multiple-instance learning approach, on the other hand, such as MILBoost, has been applied to face detection problems [128]. Multiple instance learning is a variant on supervised learning. Unlike in classical learning scheme, where each training instance has a label. In multiple instance learning, training data are presented in the form of training bags, and each training bag contains multiple sub-level training instances. Only the training bags are given labels, not the training instances. Viola et al. [128] formulated the face detection problem as a multiple-instance learning approach, and AnyBoost was modified to adapt to multiple-instance learning condition. Several multiple-instance learning methods have been proposed, such as diverse density [77] and MI-SVMs [7]. Diverse density algorithm tries to find the area which is both of high density positive points and of low density negative points. kNN is adopted for

multiple-instance learning by using Hausdorff distance in the work of Wang et al. [131].

The algorithm we proposed is based on the concept of multiple instance learning. We applied the algorithm for feature representation with imprecisely localized face images. Compared with the aforementioned work, our algorithm requires the ground truth for neither the *training* set nor the *testing* set.

2.2.2 Facial Features for Well Aligned Images

Most commonly used facial features expect very well aligned face images, and they can be categorized as either static feature and dynamic feature. The use of facial feature is closely connected to the classifier used for classification. Here we only introduce the features that have been used in related work, and in sequential modeling section 2.3 we introduce the various classifier.

Human faces convey information via four general classes of signals [34]: (1) static facial signals; (2) slow facial signals; (3) artificial signals; and (4) rapid facial signals. It's obvious that only rapid facial signals contribute to expression analysis. Most facial expression recognition systems use dynamic features that encode facial muscle movement information. Typical dynamic features include optical flow, parameterized optical flow, bag-of-words feature, volumetric features, Volumetric Local Binary Pattern (VLBP) feature and encoded dynamic features. Yacoob etc [135] used optical flow to track the motion of the facial features: brows, eyes, nose and mouth. In each facial feature region, the flow magnitude was thresholded, and the direction of any flow is also quantized to one of eight directions to give a mid-level representation. Then they cluster the optical flow directions of each region to form a high level representation. Another classical work is the local parameterized model (such as affine) of optical flow [14]. They assume that within local regions in space and time, parametric models not only accurately model non-rigid facial motions but also provide a concise description of the motions in terms

of a small number of parameters. Then they proceed to get mid-level and high-level representation of facial actions.

Part-based approaches are effective methods for object detection and recognition due to the fact that they can cope with partial occlusions and geometric transformations. Many approaches based on this idea have been presented, and the most common idea is to model a complex object or a scene by a collection of local interest points. An approach that has become very popular is the Bag-of-words model - originally proposed for document classification in a text corpus - where each document is represented by its word frequency. In the visual domain, an image or a frame of a video is the visual analogue of a document and it is represented by a bag of quantized invariant local descriptors (usually SIFT), called visual-words [65, 91]. The local features are usually densely sampled or detected by saliency detection algorithms in volumetric region. Two alternative approaches could be applied to detect salient features: the first one is to apply 2D Gaussian filter in spatial domain, and then apply 1D Gabor filter in temporal domain; the second approach: using extended Harris corner detector to detect spatio-temporal corners, which normally are key points corresponding to edges in 2D and which undergo complex temporal motion patterns. Most of the detections by space-time interest points method are caused by boundary interactions, which may not be informative. This observation motivates the decision to apply volumetric features to the motion vectors rather than to the raw pixels. Most of the information salient to recognizing actions can be captured from the optical flow, and that the appearance of the object is less relevant. Ke et al. [58] proposes an volumetric feature based on dense optical flow and use sliding window for event detection and action classification. Problem with this approach is that dense optical flow is slow to acquire and it loses temporal resolution. Also, a sparse set of selected volumetric features are hard to capture long term interaction between action patterns. Once salient features have been

detected, feature descriptors are used to represent the detected key points. SIFT or local SIFT are widely used to describe feature points. Codebook entries are constructed by clustering the feature descriptors. Once the codebook is ready, each video sequence can be represented as a bag of words from the codebook. The method relies on the assumption that one can reliably detect sufficient number of interest points. For space-time interest points this means that the video sequence must contain motion critical events. In practice, such events are rare and the histogram representation also loses temporal orders of action patterns.

In the paper [8], actions and events are modeled as a sequence of histograms (one for each frame) represented by a traditional bag-of-words model. An action is described by a "phrase" of variable length, depending on the clip's duration, thus providing a global description of the video content that is able to incorporate temporal relations. But the paper used static SIFT feature in each frame, and we believe dynamic feature would be more effective for temporal event analysis. Zhao et al. extend their 2D LBP and propose the VLBP [141] for expression recognition. Peng et al. use encoded dynamic feature [137] for video analysis. However, the method needs the whole image sequence available to extract dynamic features, which prevents it from being applied in real-time applications. In Lien's work [72], two kinds of features are used. The first feature is facial landmark points tracked using the pyramid method. The second feature is the PCA representation of the dense flow from a large facial region. The paper also extracts and analyzes the motion of high gradient components (furrows) in the spatio-temporal domain to exploit their transient variances associated with facial expression. Each motion vector sequence was vector quantized to a symbol sequence to provide an input to the facial expression classifier. An HMM-based classifier was designed to deal with varieties of facial expression sequences variable length. Essa et al. [38, 39] used three-dimensional parametric face models and optical flow for facial expression

analysis and synthesis. In [1], video-to-video face recognition was posed as a system identification and classification problem. Recognition is done by computing subspace angles between gallery and probe video.

Motion history has also been used for facial action detection in video [126] and it consists the following steps: 1) Face detection with adapted AdaBoost, 2) Divide the whole face into 20 ROIs. One feature point is detected within each ROI with Gabor wavelet. Each detected feature point is represented by a local patch's (13x13) gray value and the 48 Gabor responses (8 orientation and 6 frequencies). 3) Facial point tracking with particle filter. 4) Final facial features used are the distances from current frame to the first frame. 5) Feature selection with Boosting and expression classification with SVM. Problem with this approach is that every frame is independently classified, even though dynamic features are used.

Non-linear manifold is one of the popular representations for facial expression visualization and analysis. Hu et al. [51, 20] used a data-driven low-dimensional Leipschitz embedding to build a manifold of shape variation across several people and use Icondensation to simultaneously track and recognize expressions. Lee and Elgammal [67] built a non-linear one dimensional closed manifold to embed the facial expression and used the generative multi-linear model to decouple the style (different people) and content (different expressions) of facial motion. The model can identify people, recognize expression and the intensity of expression. With their learned model, they can also synthesize different dynamic facial appearances for different people and for different expressions. In practice, manifold based methods may not perform as well due to noises inherent in image analysis.

Besides dynamic features, static holistic features such as Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA) of gray level images have also been used. Features based on local filter responses, such as Gabor filter and Haar wavelet

features are also popular. Donato et al. [33] used three methods to extract information on upper facial expressions, including optical flow based feature, holistic spatial analysis (principal component analysis or PCA, linear discriminant analysis or LDA, independent component analysis or ICA, local feature analysis or LFA), and Gabor filter response. For holistic spatial analysis feature, they used the differences in images obtained by subtracting the gray values of the neutral expression (first frame) from those of the subsequent images for each image sequence. The subtraction process would be sensitive to face alignment errors, which is hard to eliminate. They showed that the best performance for recognizing AUs was achieved with the Gabor wavelet representation and the independent component analysis. Bartlett et al. [11] consider using SVMs on the Gabor wavelet coefficients of a face image, and they reported that the best results on classifying facial expressions are achieved by using SVMs with feature selection through AdaBoost. This is also in line with our finding that the AdaBoost selected features is efficient compared with other feature selection methods we compared. In addition to features based on texture, geometric features are also used for expression recognition. Kapoor et al. [56] have constructed a shape model of the upper face, and uses the ASM coefficients as features for recognition. Tian et al. [71] describe a neural network approach where facial expressions are analyzed based on a set of permanent facial features (brows, eyes, mouth) and transient facial features (furrows).

2.3 Sequential Facial Expression Modeling

Computerized facial expression recognition from video sequences includes researches on feature extraction and selection, spatio-temporal modeling, and the literature review is necessarily sparse. Previous work on facial expression analysis fall into two broad categories: static recognition and dynamic recognition. By static recognition we mean the system doesn't take into account the temporal labeling information in a image

sequence, even though it can still use dynamic features, such as optical flow. By dynamic recognition we mean the system makes explicit use of the temporal context in a video sequence. Due to the advancement of relative research work on machine learning techniques and the readily available computational power, dynamic modeling of facial expressions have been very popular. Spatial-temporal features have been introduced in section 2.2. We mainly review relatively new spatio-temporal modeling efforts in action and facial expression analysis filed.

While temporal classification may be a relatively newly explored field, studying a closely related and extensively studied topic, time series, could help the understanding of the temporal classification problem. Approaches typically tried to model time series as an underlying trend, a long term pattern, together with a seasonality, short term patterns. Most popular technique is the Autoregressive Integrated Moving Average (ARIMA) model. The field of signal processing has also explored time series. The objective of signal processing is to characterize time series in such a manner as to allow transformations and modifications of them for particular purposes. One of the oldest techniques for time series analysis is the Fourier transform [17]. The Fourier transform converts from a time series representation to a frequency representation. Fourier's theory states that any periodic time series can be represented as a sum of sinusoidal waves of different frequencies and phases. Converting from a time series representation to a frequency representation allowed certain patterns to be observed more easily. However, the limitation of periodicity is quite significant, and so relatively new work has focused on wavelet analysis [76], where any time series can be represented not as a sum of sinusoidal waves, but "wavelets", non-periodic signals that approach zero in the positive and negative limits.

Graphical models have long been applied to computer vision research [44, 87]. Recently developed efficient training and inference techniques have made graphical models

even more appealing for a broad range of vision research fields, including facial and body action analysis [60, 121, 140]. Both generative models and discriminative models have been used for expression recognition. Tong et al. [121] construct a Dynamic Bayesian network (DBN) to model the spatial and temporal relationships among the action units. Zhang et al. [140] establish a DBN model to correlate the relationships between the emotions and the action units. Such complex models need large amount of training data, which renders them impractical in a lot of situations. Alternatively, there are methods formulated to directly recognize basic emotions without referencing action units. Cohen et al. [21] consider the tree-augmented-naive Bayes (TAN) classifier to learn the dependencies between the facial emotions and the motion units. They also proposed an architecture which performs both segmentation and recognition of the facial expressions automatically. Their proposed multi-level hierarchical HMM is composed of an HMM layer and a Markov chain layer. Most previous mentioned dynamic models are generative models, in the sense that their objective is to maximize the joint distribution of class label and observation. If the designed likelihood model doesn't fit data well, generative models tend to give poor prediction accuracy. Due to the relaxed assumptions on feature independences and better prediction accuracy in modeling spatial-temporal events, discriminative models, such as Conditional Random Fields (CRF) [64] and Discriminative Random Fields (DRF) [63], hybrid generative and discriminative models [136, 66], have been a competitive alternative to traditional generative models. These models have been successfully applied to natural language processing [64], visual classification [63, 102, 111] and motion tracking [60, 112, 132, 85] field. Kim et al. [60] proposed two alternative approaches for improving generative model performances: 1) learning generative model with discriminative objective 2) developing undirected conditional model. The undirected graphical model can be seen as an extension of the conditional random fields to deal with multivariate real-valued state

variables given observation sequences. They designed an efficient convex optimization algorithm for model learning and applied the learned model on human motion tracking and robot-arm state estimation.

Static classifiers have also been used with majority voting or other scheme for sequential classification. Essa et al. [38, 39] performed expression recognition by comparing similarity to the standard training expression templates, temporal effect is ignored. Yacoob et al. [135] used a rule based method for expression recognition. By dividing each expression into three phases, beginning, apex and ending, temporal information are also considered. However, the overall rule based classifier is susceptible to human judgments. Donato et al. [33] dealt with the time warping problem by manually picking up six frames from each image sequence. They showed that the best performance for recognizing AUs was achieved with the Gabor wavelet representation and the independent component analysis.

Most of the preceding techniques [72, 11, 51, 21, 140, 121] on linking facial expression to emotions, the emotion class label is frame-wise predicted (despite the formulation may use temporal information, as in CRF tagging). These approaches may not be reasonable, as the making of a facial expression is a natural transition over time. An alternative approach is to sequence-wise classify the emotion class label, and the Hidden CRFs (HCRFs) [102, 49, 132] seems a good choice. Chang et al. [19] use partially observed HCRFs and the Gabor feature with AdaBoost feature selection for facial expression recognition. Although their model is similar to ours, the features they used are not as effective and the combination of their feature and classifier performs relatively bad. On the same data set, our CRF and HCRF models perform much better than theirs. The coupling of the effective part-based dynamic feature with the spatio-temporal model is the key for good performance.

Chapter 3

Face Tracking

Reliable face detection and face tracking is the first component of our video-based facial expression system. 3D deformable model proves to be an effective approach for face tracking and facial animation [31, 32, 15, 30, 47, 48, 129]. Parametrized 3D deformable models rely on the correct estimation of image features to fit the model’s parameters. Most existing methods for image feature extraction are based on deterministic feature tracking (e.g KLT) or tracking feature distributions [52, 47, 133, 2]. However, these methods normally rely on local computations for the tracking of each feature. If the assumptions that underlie the feature extraction methods are invalid, because of factors such as changes in illumination, noise or occlusions, the feature correspondences from frame to frame are not computed correctly. Such errors in turn cause the 3D model to drift over time, and results in incorrect tracking of the video. Learning based ASMs tend to give more reliable results than feature trackers for poses that match closely the ones on which they have been trained. They also are less vulnerable to drifting, because they locate the features directly in the image. Conversely, ASMs can fail spectacularly when they encounter a situation where the initialization is too far away from target location. Parametrized 3D deformable models, on the other hand, allow us to perform unsupervised techniques on the data driving the tracking procedure, such as parameter-space outlier rejection [129], occlusion handling [46], and statistical inference to feed a Bayesian Filter such as the Kalman Filter [48].

In this work [130], we try to bring together the best of both worlds. We develop

a framework for robust 3D face tracking that combines the strengths of both 3D deformable models and ASMs. In particular we use ASMs to track reliably 2D image features in a video sequence. In order to track large rotations we train multiple ASMs that correspond to frontal and several side head poses. The 2D features from the ASMs are then used for parameter estimation of the 3D deformable model. After the 3D parameter-space outlier rejection and occlusion handling, the updated 3D model points are projected to 2D image to initialize the 2D ASM search, which becomes more robust due to the better initialization. In addition we couple stochastically 3D deformable models and ASM by using the 3D pose of the deformable model to switch among the different ASMs, as well as to deal with occlusion. This integration allows the robust tracking of faces and the estimation of both their rigid and nonrigid motions. We demonstrate the strength of the framework in experiments that include automated 3D model fitting and facial expression tracking for a variety of applications including American Sign Language.

3.1 Overview of 3D Deformable Models

Three-dimensional deformable models belong to the general family of parametrized models, where a small numbers of parameters, denoted by the vector \vec{q} , control the shape, orientation, and position of the model [30, 101, 116, 129] . For any given point p_i on the model surface, its 3D position $\vec{p}_{3d,i}$ is determined by a function defined over the parameter vector space:

$$\vec{p}_{3d,i} = F_i(\vec{q}). \quad (3.1)$$

Given a projection function, such as the one defined by a perspective camera, the 2D position of each model point is

$$\vec{p}_{2d,i} = Proj_i(\vec{p}_{3d,i}), \quad (3.2)$$

where $Proj_i$ is the projection function for the given point p_i . Conceptually, tracking a 3D deformable model consists of adjusting \vec{q} such that the projected model points $\vec{p}_{2d,i}$ best match the image being tracked.

Traditionally, the first step in tracking consists of finding correspondences between the tracked image and the model points via standard computer vision techniques, such as point tracking, edge detection, and optical flow. The displacements between the actual projected model points $\vec{p}_{2d,i}$ and the identified correspondences are called *image forces*, and denoted by $\vec{f}_{2d,i}$. Adjusting \vec{q} is done via gradient-descent optimization. To this end, the image forces projected into parameter space, resulting in *generalized forces* $\vec{f}_{g,i}$ that act on \vec{q} :

$$\vec{f}_{g,i} = \mathbf{B}_i^\top \vec{f}_{2d,i}, \quad (3.3)$$

where

$$\mathbf{B}_i^\top = \frac{\partial Proj_i}{\partial \vec{p}_{3d,i}} \frac{\partial F_i}{\partial \vec{q}} \quad (3.4)$$

is the projected Jacobian of the model point p_i with respect to the parameter vector.

The generalized forces themselves are summed up into a single quantity

$$\vec{f}_g = \sum_i \vec{f}_{g,i}. \quad (3.5)$$

We then use Euler integration to solve the dynamical system

$$\dot{\vec{q}} = \mathbf{K}\vec{q} + \vec{f}_g, \quad (3.6)$$

where \mathbf{K} is a stiffness matrix.

Goldenstein showed in [47] that the image forces $\vec{f}_{2d,i}$ and generalized forces \vec{f}_g in these equations can be replaced with affine forms that represent probability distributions, and furthermore that with sufficiently many image forces, the generalized force \vec{f}_g converges to a Gaussian distribution. This result allows us to apply statistical inference at both Equations 3.3 and 3.5, upon which we elaborate in Section 3.3.3. This kind of

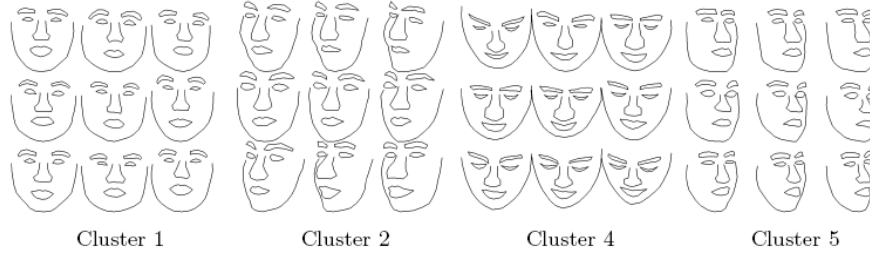


Figure 3.1: Example multi-view ASM Shapes

statistical inference would not be possible if we used ASMs alone, without 3D models, and contributes substantially to the robustness of the combined 3D model and ASM tracking.

Note that it is impossible to get the distance from the camera to the face without knowing the camera focal length and the head size, though face tracking is generally not sensitive to camera focal length [2].

3.2 Overview of Active Shape Model

ASMs [25] are landmark based model that tries to learn a statistical distribution over variations in shapes for a given class of objects. It could be viewed as an discrete version of the active contour model. Shape statistics derived from training images are used to restrict the shape range to an explicit domain. An ASM consists of two parts, a global shape model and a set of local profile models. The global shape model models shape variations. Local profiles are first derivatives of the sampled profiles perpendicular to the landmark contour. The local profile models capture the grey-levels around each landmark point and they are used for selecting the best candidate image points.

Suppose we have m aligned training face images using Procrustes Analysis, and for each training image we have n landmark points. We can represent the training images as a set of m $2n$ -dimensional vectors $\vec{x}_i \in \{\vec{x}_1, \dots, \vec{x}_m\}$. Suppose the variance covariance matrix for m training images is $\mathbf{\Lambda}$, and the eigenvectors $\mathbf{\Psi} = [\vec{\psi}_0, \vec{\psi}_1, \dots, \vec{\psi}_{2n-1}]$ of



Figure 3.2: Example multi-view ASM search results

the covariance matrix $\mathbf{\Lambda}$ span a linear subspace where valid face shapes lie in some particular region of that space.

The eigenvectors corresponding to the first few large eigenvalues represent major modes of shape variations. Any valid face shape can be represented as a linear combination of the basis vectors

$$\vec{x} = \bar{\vec{x}} + \delta\vec{x} \quad (3.7)$$

$$= \bar{\vec{x}} + \sum_{i=0}^{n-1} \left(b_{2i} \vec{\psi}_{2i} + b_{2i+1} \vec{\psi}_{2i+1} \right), \quad (3.8)$$

$$= \bar{\vec{x}} + \mathbf{\Psi} \delta\vec{b} \quad (3.9)$$

where $\delta\vec{b}$ is the coordinate difference between average shape $\bar{\vec{x}}$ and current shape \vec{x} . During ASM searching, we project the search candidate results into this space and choose the closest possible valid face shape to be the ASM search results. Specifically, given shape variation, the shape parameter can be estimated using linear least square as:

$$\delta\vec{b} = (\mathbf{\Psi}^T \mathbf{\Psi})^{-1} \mathbf{\Psi}^T \delta\vec{x} \quad (3.10)$$

which, as $\mathbf{\Psi}$ is orthonormal, simplified to:

$$\delta\vec{b} = \mathbf{\Psi}^T \delta\vec{x} \quad (3.11)$$

Active shape models in principal model only linear shape variations and they are

viewpoint dependent. For example, a single ASM is capable to cope with shape variations from a narrow range of face poses (turning and nodding of roughly 20 degree). Shape variations across different views cannot be modeled with a single linear shape model. Thus, for face tracking purpose, we need to learn multiple local linear ASMs to approximate the nonlinear shape space for different viewpoints. Figure 3.1 shows some example learned multi-view shape models. However, to smoothly switch between different local shape models is hard without using 3D head pose information. In this thesis, we used the 3D deformable model to guide the switch between different models. Given 3D poses from deformable model, the 2D ASM search works very well, as shown in figure 3.2.

3.3 ASM and 3D Deformable Model Integration

3D deformable models and 2D ASMs complement each other. Although they are both parametrized models, with \vec{q} taking on the role of the parameters in the case of 3D deformable models (Equation 3.1), and \vec{b} taking on this role in the case of 2D ASMs (Equation 3.9), they function in very different ways.

The 3D deformable face model parametrization is crafted by the designer to meet the needs of the application. Each parameter stands for a specific high-level model action, such as eyebrow movement, lip parting, and so on. Particularly, in our experiments, we have parameters corresponding to eyebrow, mouth and jaw movements. This representation also facilitates retargeting the application to different users with different face shapes, without having to change the interpretation of the parameters. Human-designed parametrization, however, are an idealization and thus an approximation to the actual deformations happening on a video. If a tracking cue, such as a point tracker, depends on being initialized with accurate positions from the model on every frame, any inaccuracies lead to drift over time as approximation errors cascade.

The parametrization of ASMs, on the other hand, is constructed from training data. Although it has no discernible high-level semantics and is not easily retargetable, the parametrization allows close correspondences between the model and images sufficiently similar to training examples. In addition, the ASM search matches shapes, instead of tracking iteratively from frame to frame, which allows them to locate features, even if they lost track in a previous frame. However, they are susceptible to getting stuck in local minima if the starting position is too far from the target, they cannot handle large off-plane rotations by the face, and tend to yield unreliable results when the tracked image does not fit the training data well.

These considerations show that 3D deformable models and ASMs can compensate for each other’s weaknesses: ASMs help with correcting drift and locating facial features, whereas the 3D model can help with a good initial position for the ASM, selecting the correct ASM depending on the face’s pose, and providing mechanisms for statistical inference.

3.3.1 Overview of Combined Tracking Algorithm

The basic premise of the tracking algorithm is that the ASM is initialized with the current shape and position of the 3D model and then is tracked to the next frame, so as to establish the image displacements $\vec{f}_{2d,i}$ for Equation 3.5. The 3D model parameters are then adjusted to match the tracked ASM as closely as possible, and the cycle starts anew.

In order to establish the displacements between the ASM points \vec{x}_i and 3D deformable model points $\vec{p}_{2d,i}$, the tracking system needs to know which ASM points correspond to which 3D model points. These correspondences need to be defined manually, once per 3D model topology, and are shown in Figure 3.3.

We now give a detailed explanation of the tracking algorithm.

Algorithm 1 Tracking algorithm for 3D models coupled with ASMs

- 1: Fit 3D model to initial frame
 - 2: **while** there are frames to track **do**
 - 3: Select ASM asm based on 3D model orientation
 - 4: $\vec{x}_{asm} \leftarrow \vec{p}_{2d}$ {Equations 3.2, 3.9: Update ASM points from 3D model}
 - 5: Compute and truncate eigen coefficients \vec{b}_{asm} and adjust \vec{x}_{asm} to return ASM to valid state. {Equations 3.9, 3.11}
 - 6: Based on ASM intensity profile model, search next frame for new \vec{b}_{asm} , \vec{x}_{asm} {Track ASM}
 - 7: $\vec{f}_{2d,i} \leftarrow \vec{p}_{2d,i} - \vec{x}_{asm,i}$ {2D displacements between 3D model and tracked ASM}
 - 8: Reject bad $\vec{f}_{2d,i}$ via statistical inference {Compensates for jumpy ASMs}
 - 9: $\vec{f}_{g,i} \leftarrow \mathbf{B}_i^\top \vec{f}_{2d,i}$ {Equations 3.3, 3.4}
 - 10: Reject bad $\vec{f}_{g,i}$ via statistical inference {Compensates for occlusions, bad correspondences}
 - 11: Integrate $\vec{q} = \mathbf{K}\vec{q} + \sum_i \vec{f}_{g,i}$ {Equations 3.5, 3.6}
 - 12: **end while**
-

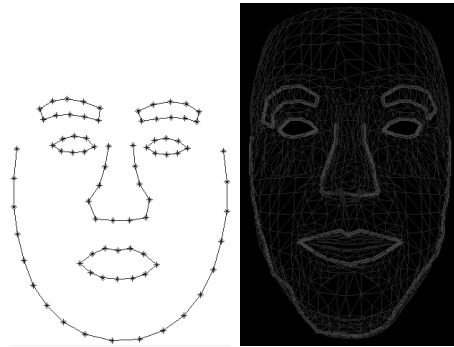


Figure 3.3: ASM to 3D model correspondences. The thick red lines on the right show the 3D model nodes that correspond to the ASM nodes on the left.

3.3.2 Explanation of the Tracking Algorithm

Step 1 matches the 3D model to the initial image. This step is based on establishing correspondences between selected 3D model points and the image. If we have a well-trained ASM for the person to whom we fit the 3D model, this step can be automated (see also the experiments in Section 3.4.1). Otherwise, the correspondences need to be defined manually. Once they are defined, we integrate Equation 3.6 via 3.3 on the model's shape, position and orientation parameters, which results in an adaptation of the 3D model to the image and person being tracked.

Step 3 selects the best ASM given the current model orientation. Recall that ASMs are a view-based method and thus sensitive to changes in the subject's orientation. We train multiple ASMs for different view angles — left, front, right, down, and up. We split and select the appropriate ASM for the views according to whether they are more than 20 degrees from a neutral frontal view.

Steps 4 and 5 provide the initial guess for the ASM search step described in Section 3.2 (Figure 3.4 left). Step 5 is necessary, because the new positions \vec{x}_i likely do not fit the constraints of the eigenspace Ψ , so they need to be projected into the eigenspace. This projection obtains the new parameter vector \vec{b} that makes the shape fit as closely as possible to the initial guess (Figure 3.4 center). The ASM search then finds the matching image features for the next frame (Figure 3.4 right).

Step 6 tracks the ASM to the new frame, based on the initial guess from steps 4 and 5. This step results in new positions for the ASM's nodes \vec{x}_i that are consistent with the eigenspace. The displacement between these ASM nodes and the projected 3D model nodes constitutes the image forces $\vec{f}_{2d,i}$ described in Equation 3.3.

Step 9 calculates the generalized forces, which describe the effect that the displacement between the current frame and the next frame has on the model parameters.

These are then integrated in step 11, which yields the model parameters for the next frame and finds the best possible match between the 3D model and the next image frame.

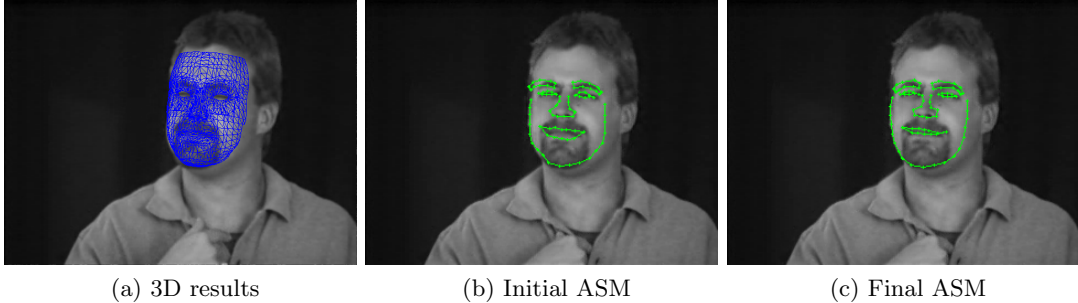


Figure 3.4: 3D model-guided ASM initialization. 3.4a: tracked model position from previous frame. 3.4b: initial ASM guess in next frame, based on model position from previous frame. 3.4c: final ASM shape in next frame.

We now turn to an explanation of the statistical inference in steps 8 and 10.

3.3.3 Statistical Inference: Unsupervised Meets Supervised

When an image exhibits patterns and structures incompatible with what the ASM has learned, the resulting shape from the ASM search step is likely to be contorted, or can even jump off track altogether. Although often these contortions are temporary, and eventually a correct ASM configuration is found again, any data over this period of time would be wildly inaccurate. We thus need to detect such contortions.

The 3D deformable model framework shines at this point. The statistical inference in step 8 of the parameters distributions give us an estimate of the expected distribution of the model points [47, 48] over time. By projecting these into 2D space, we can obtain a probability distribution of where each model point is expected to be over time in each image, as described in [129]. Any ASM point that falls too far outside these distributions is discarded. This approach is fast and detects moments where the entire ASM estimate is off-base.

The statistical inference of step 10, on the other hand, looks for more subtle problems. Sometimes, there are occlusions, or image artifacts, that compromise the ASM search on only a few parts of the image. This generally causes the ASM estimate to be subtly incorrect in a few places. In this situation, a 2D approach is not fine-grained enough. Using a robust statistical estimate of the probability distribution of the generalized forces $\vec{f}_{g,i}$ in parameter space, we instead reject outliers based on how well a single force $\vec{f}_{g,i}$ matches this distribution [129].

Both steps 8 and 10 rely on obtaining and propagating probability distributions through the 3D model equations. They couple the supervised learned behavior of the ASM component with the data-driven unsupervised outlier rejection performed by the 3D deformable model component. This coupling is the main contribution of this thesis, and achieves impressive results in sequences that could not be analyzed before by either method alone, see Section 3.4.2.

Our tracking framework can also be viewed from the linear dynamical system perspective [95, 96, 112]. Linear dynamical system is similar to HMM in the sense that both assume a hidden state variable, and the hidden states evolves with Markovian dynamics. The hidden states also emit noisy measurements. The difference is that HMM uses discrete hidden state variable with arbitrary dynamics and arbitrary measurements, while LDS has continuous state variable with linear Gaussian dynamics and measurements [84]. The forward-backward inference procedure for HMM, when specialized to linear Gaussian assumptions, leads directly to the Kalman filtering. In practice, the most popular tracking algorithm is the simulation based particle filter [52] algorithm.



Figure 3.5: Automated model fitting and adaptation. Left to right: Before model fitting, ASM feature detection, Adjustment of rigid model parameters, Final fit after adjustment of nonrigid shape parameters.

3.4 Experimental Results

We ran a variety of experiments to test and validate the integration of ASMs with 3D deformable models. These experiments include automated fitting and shape adaptation, and use a combination of individually unreliable estimates from ASMs and other cues to track 3D models under difficult circumstances.

3.4.1 Automated Calibration and Model Adaptation

The first step in tracking consists of automated calibration and model adaptation. Assuming that the first frame of the video sequence to track is a frontal face view, we first use the Viola-Jones face detector [127] to obtain a bounding box around the face to provide an initial estimate of its location. Then, ASMs can locate the frontal face feature points accurately. According to the predetermined correspondences between the 2D ASM points and 3D generic face model from Figure 3.3, we obtain two sets of coordinates (2D and 3D).

We then iteratively perform the following two steps:

1. Rigid parameters: With the 2D coordinates from ASM and the 3D coordinates of a generic face model, we use camera calibration techniques to determine the rotation

and translation of the model relative to the world coordinate system.

We have 79 2D-3D data pairs to recover the projection matrix in a robust manner. Once we get the projection matrix, it can be easily decomposed into the product of an upper triangular matrix K , which contains internal parameters, an orthogonal matrix for rotation and a translation part:

$$P_{3 \times 4} = K_{3 \times 3} R_{3 \times 3} [I_{3 \times 3} T_{C3 \times 1}]_{3 \times 4}.$$

2. Nonrigid parameters: We project the 3D model points using the recovered rotations and translations from step 1, and proceed as described in Step 1 of the tracking algorithm, explained in Section 3.3.2. Figure 3.5 shows the results of the fitting experiments.

3.4.2 Statistical Inference with 3D models and ASMs

In order to test the effect of combining ASMs with 3D model-based statistical inference, we tracked a particularly difficult sign language sequence taken from the National Center for Gesture and Sign Language Resources at Boston University ¹. The difficulties stem from the numerous occlusions and fast movements. None of the ASMs was trained on this particular subject, so we expected to see numerous “jumps” — as described in Section 3.3.3. In addition, we integrated contributions from a KLT tracker and an edge tracker, neither of which provided wholly reliable results, as well. The integration of these cues is based on a maximum likelihood estimator for the Gaussian probability distributions of the generalized forces, as mentioned at the end of Section 3.1, and is described in detail in [47].

Figure 3.6 shows how step 8 in the tracking algorithm manages to compensate for jumpy ASM behavior by excluding those points that stray too far from the most

¹Data collected at the National Center for Sign Language and Gesture Resources, Boston University, under the supervision of C. Neidle and S. Sclaroff.

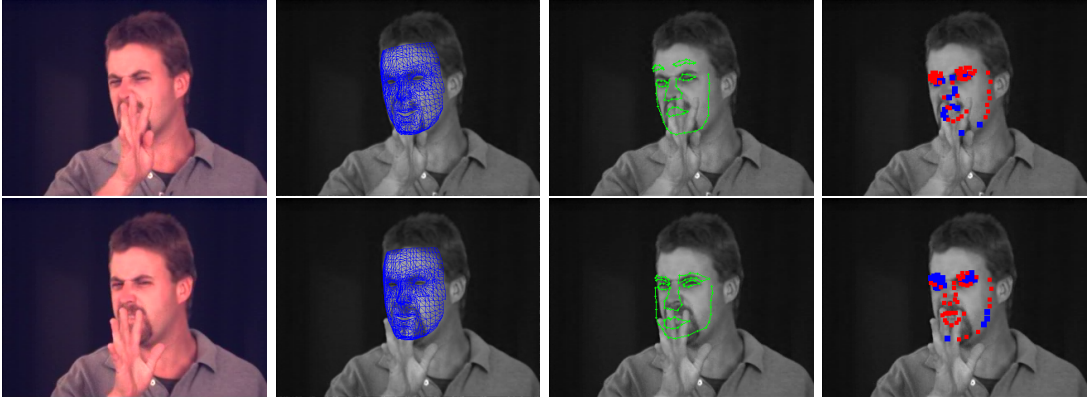


Figure 3.6: Tracking example where ASMs encounter difficulties, and are only partially reliable. The ASMs shown in this figure have never been trained on this subject. Left to right: Original frame, Tracked 3D model, ASM search result, Statistical rejection of unreliable ASM features, using projected regions of confidence (in red, blue points are accepted).



Figure 3.7: Results of sign language tracking sequence using ASM, KLT and edge tracking.

probable predicted path of the 3D model over several frames. The tracking results in Figure 3.7 are surprisingly good, considering how unreliable all the individual cues are. As the accompanying video shows, in several instances the ASM component pulls the mask back on track. It also reinforces the claim in [46] that the statistical inference in step 10 of the tracking algorithm compensates for occlusions.

Chapter 4

Facial Feature Representation and Selection

As in all pattern recognition tasks, a good representation of the input data, is as important as a suitable classifier for getting good recognition rates. Good features should be succinct and discriminative for classification tasks. Parsimonious features often lead to better generalization of classification algorithms. In this chapter, we discuss facial feature representation and selection schemes under different face alignment conditions. We propose an effective method for selecting good facial features under poor face alignment conditions, and we also empirically compare common feature extraction and selection methods with well aligned face images.

4.1 With Poor Face Alignment

In this work [70], we systematically investigate the effect of mis-aligned face images on face recognition systems. To make classifiers more robust imprecisely localized face images, we formulate the facial feature representation and selection with poorly aligned face images as a multiple-instance learning task. We propose a novel multiple-instance based subspace learning scheme for facial feature representation and dimensionality reduction. In this algorithm, we augment our training data with randomly perturbed images, and the augmented data are modeled as the mixture of Gaussians. We introduce a method to iteratively select better subspace learning samples from this augmented dataset. Compared with previous methods, our algorithm does not require accurately aligned *training and testing* images, and can achieve the same or better performance

as manually aligned faces for face recognition and facial expression recognition tasks.

In this thesis, we used the term "*noisy images*" to denote poorly aligned images.

4.1.1 Multiple-Instance based Feature Selection

Motivation

Given a limited set of noisy training images, we augment the training set by perturbing the training images. The augmented larger training set will normally cover more variations for each subject and thus model the alignment error, however, it could also introduce some very poorly registered faces into the training set, which will have negative effect for the learning process.

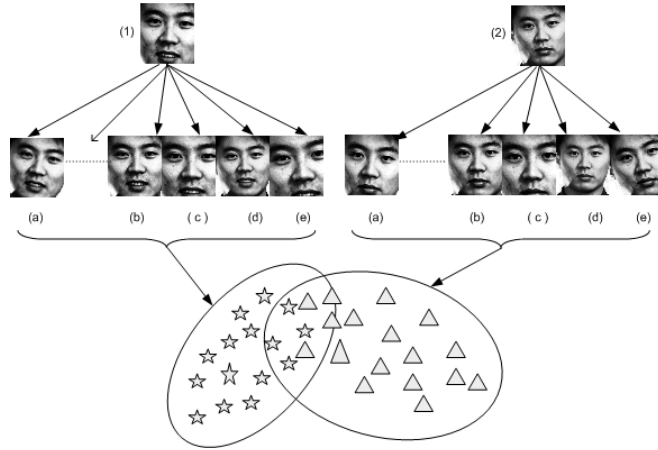


Figure 4.1: Bags of Face Images. From the original not precisely cropped input image, we generate more samples by adding perturbations. High density area in the subspace represents better face samples.

Figure 4.1 shows two noisy training images (1) and (2). From each of the noisy image, we generate two bags each with multiple instances, denoted by (a), (b), ... (e) in the figure. While image (b) and (d) will certainly benefit the training process, image (e) will most likely cause confusion for the classifier, since it could be more similar to other subject. As will be shown later, those very poorly registered images will indeed increase the recognition error. Thus given noisy training images, we must build algorithm that

can automatically select those "good" perturbed images from training bags, and exclude those very poorly registered images from being selected.

Approximating the Constrained K-minimum Spanning Tree

Excluding very poorly registered images from the noisy bags can be formulated within the multiple-instance learning framework. One assumption is that the good perturbed images from the same subject tend to be near to each other. The high density areas correspond to the good perturbed images, while the low density areas correspond to poorly perturbed images, and those are the bad images we want to exclude from the training set. As shown in figure 4.1, the good perturbed images will lie in the intersection area of the two bags. The idea is very similar to the diverse density approach used by Maron [77] for multiple-instance learning. Since the perturbed noisy images have irregular distribution, we use non-parametric method to find out the high density area. Our non-parametric method is based on k -minimum spanning tree [16]: given an edge-weighted graph $G = (V, E)$, it consists of finding a tree in G with exactly $k < |V| - 1$ edges, such that the sum of the weight is minimal. In our face recognition application, the nodes will be the face image instances, and the edges represent the Euclidean distance between face image instances. The problem is known as NP-complete problem. Fortunately, exact solution is not needed for better performance. We used heuristic method to find out the approximate k -minimum spanning tree. Firstly, for each instance, we build its k -nearest neighbor graph. Among all the instances, we find the one with minimum k -nearest neighbor graph. Since the size of the neighbors is fixed by k , the one with minimum sum of k -nearest neighbor graph will have the highest density, and thus corresponds to the good perturbed subspace area. Although in this high density area, there still exists some noisy images, those noisy images are identifiable and useful to our learning algorithm.

We also need to add the constraint to include at least one instance from each bag during the base selection phase. The idea is similar to that of MI-SVM. In MI-SVM, for every positive bag, we initialize it with the average of the bag, and compute the QP solution. With this solution, we compute the responses for all the examples within each positive bag and take the instance with maximum response as the selected training example in that bag. In our k -nearest neighbor graph algorithm, if some bag is far from other bags, using only the k -nearest neighbor graph to select training images may not include any instance from this isolated bag. We force the algorithm to accept at least one instance from every bag. If all the instances in a bag fall outside the most compact k -nearest neighbor graph, we select the instance with the minimum distance to the k -nearest neighbor graph.

The iterative multiple-instance based FisherFace [12] [40] learning procedure is shown in the following algorithm.

Algorithm 2 Multiple-Instance Subspace Learning Algorithm

Input:

S : number of subjects

$N_s, s = 1 \dots S$: number of noisy image for subject s

R : number of instances per bag

K : target number of nearest neighbors

- 1: Initialize $x_b^{(0)} = 1/R \sum_{i=1}^R x_{bi}^{(0)}$;
 - 2: **while** Base is still changing **do**
 - 3: Compute Sufficient Statistics

$$x_b^{(t)} = \frac{1}{Z} \sum_{g \in \mathcal{G}_s^{(t)}} x_{bg}^{(t)};$$
 - 4: Compute Multiple-Instance Eigenbase

$$m_s^{(t)} = \frac{1}{N_s} \sum_{b=1}^{N_s} x_b^{(t)}$$

$$m^{(t)} = \frac{1}{\sum_{s=1}^S N_s} \sum_{s=1}^S N_s m_s^{(t)}$$

$$S_W^{(t)} = \sum_{s=1}^S \sum_{b=1}^{N_s} \sum_{g \in \mathcal{G}_s^{(t)}} (x_{bg}^{(t)} - m_s^{(t)})(x_{bg}^{(t)} - m_s^{(t)})^T$$

$$S_B^{(t)} = \sum_{i=1}^S N_s (m_i^{(t)} - m^{(t)})(m_i^{(t)} - m^{(t)})^T$$

$$W^* = \arg \max_W \frac{W^T S_B^{(t)} W}{W^T S_W^{(t)} W}$$
 - 5: Base Selection

$$y = W^{*T} * x$$

Select good perturbed training samples \mathcal{G}_s for each subject by finding the most compact k -nearest neighbor graph from projected subspace y .
 - 6: **end while**
-

The learning procedure normally takes 2-3 iterations to converge. In our experiments, we use bag size of 25, i.e., each original training image is perturbed to generate 25 images. Each subject has 1-4 training images, and we take k as 60% of the total number of perturbed noisy images. The algorithm is based on LDA feature, and it can easily be modified for the case of PCA feature.

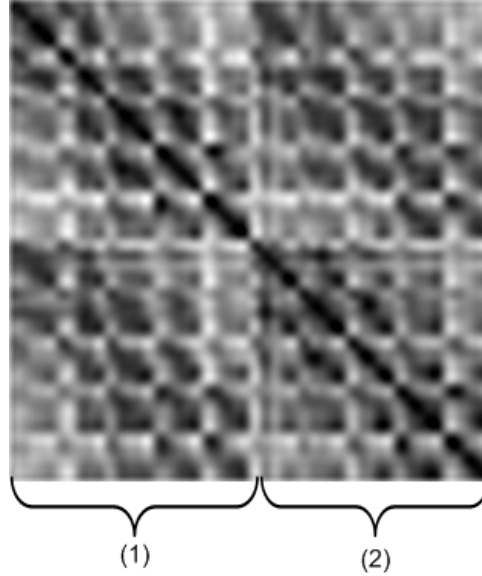


Figure 4.2: Bag Distances Map

To show that good perturbed images are similar to each other, figure 4.2 shows an example distance map for two bags (1) and (2). Each bag has 25 instances, which are generated by adding 25 random perturbations to a well-aligned image. The instances around the middle of the two bags have smaller perturbations, i.e., they are good perturbed images. In the distance map, the darker the color, the similar the two instances. From the graph we can see that an instance from bag (1) is not necessarily always nearer to instances in bag (1) than in bag (2), which means that two aligned different face images from one subject could be more similar than the same image to itself perturbed by noises. Also we can see that instances around the middle of bag (1) are more similar to those instances around the middle of bag (2), which means good

perturbed images from the same subject are similar to each other, and thus confirmed our assumption.

Testing Procedure

The aforementioned feature selection scheme produces low dimensional features which can then be fed into most commonly used classifiers. To simply illustrate that our selected features indeed are effective with poorly aligned face images, we use simple classifiers such as the Nearest Neighbor (NN) for face recognition and multinomial logistic regression for facial expression recognition. One could use other classifiers to get better results. The distance metric for the nearest neighbor classifier is the modified Hausdorff distance. The Hausdorff distance provides a distance measurement between subsets of a metric space. By definition, two sets \mathcal{A} and \mathcal{B} are within Hausdorff distance of d of each other iff every point of \mathcal{A} is within distance of d of at least one point of \mathcal{B} , and every point of \mathcal{B} is within distance d of at least one point of \mathcal{A} . Formally, given two sets of points $\mathcal{A} = \{A_1, \dots, A_m\}$ and $\mathcal{B} = \{B_1, \dots, B_n\}$, the Hausdorff distance is defined as: $H(\mathcal{A}, \mathcal{B}) = \max\{h(\mathcal{A}, \mathcal{B}), h(\mathcal{B}, \mathcal{A})\}$, where $h(\mathcal{A}, \mathcal{B}) = \max_{A_i \in \mathcal{A}} \min_{B_j \in \mathcal{B}} \|A_i - B_j\|$. This definition is very sensitive to outliers, so we used a modified version of the Hausdorff distance. In this thesis, we take the distance of bag A and bag B as $H(\mathcal{A}, \mathcal{B}) = \min_{A_i \in \mathcal{A}} \min_{B_j \in \mathcal{B}} \|A_i - B_j\|$. For single instance probe and gallery testing case, we use the nearest neighbor method based on Euclidean distance in the subspace.

4.1.2 Experimental Results

We performed two sets of experiments: one with face recognition and the other one with facial expression recognition.

For face recognition, we used the well known FERET database [100] in our experiments. One reason to use this data set is that it's relatively a large database available,

Table 4.1: Testing combinations for aligned training data

Base 1	single aligned training
Base 2	aligned bag training
Testing 1	single aligned gallery, single aligned probe
Testing 2	single aligned gallery, single noisy probe
Testing 3	aligned bag gallery, noisy bag probe
Testing 4	single aligned gallery, noisy bag probe

and the testing results will have more statistical significance. The training set, which is used to find the optimal FisherFace subspace, consists of 1002 images of 429 subjects, with all subjects at near-frontal pose. The testing set consists of the gallery set and the probe set. The gallery set has 1196 subjects, each subject has one near-frontal image with under normal lighting condition. The probe set has 1195 subjects, each subject has one image with the same condition as the probe set, but with different expressions. For comparison purposes, we have the ground truth positions of the two eye centers for training, probe and gallery images.

In this thesis, we denote *noisy bag* as a bag generated from a noisy image, and *aligned bag* as a bag generated from a well-aligned image. We use "single" in comparison to bag.

Since we have many possible experimental setup combinations (training data, gallery data, probe data, noisy image, well-aligned image, single image and bag of images etc), we use table 4.1 and table 4.3 to explain our experimental setup.

Testing with Well-aligned Training Data

To see how the introduction of the augmented training bags will affect the recognition performance, we first test on the well-aligned training data.

From table 4.2 we have the following notable observations:

- The recognition rate is always higher if we use aligned bag instead of single image

Table 4.2: Results comparison

	base 1	base 2
testing 1	0.9247	0.9749
testing 2	0.8795	0.9665
testing 3	0.9674	0.9849
testing 4	0.9431	0.9774

as training data, which motivates the aforementioned perturbation based robust algorithms. However, it's not true anymore if we don't have well-aligned training data, i.e., we only have some noisy training images, and we add perturbations to generate noisy bags. Using the noisy bags as training data may not necessarily improve recognition performance, since the very poorly aligned images will confuse the classifier.

- If we take the baseline algorithm as the case of single aligned training, single aligned gallery and single aligned probe, then the rank-1 recognition rate for the baseline algorithm is 92.47%.
- If we use aligned bag probe and noisy bag probe, the rank-1 recognition rate is 96.74%, which is better than the baseline algorithm. It means adding perturbations to the gallery and probe set can make the algorithm robust to alignment errors.

Testing with Noisy Training Data

To show that if we don't have well-aligned training data, adding random perturbations to augment the training set may help much, we performed various experiments. More importantly, we also show that after selecting good perturbed images using our multiple-instance based scheme from the set of augmented data, the recognition performance improves a lot. Table 4.4 shows the testing results, and we have the following notable

Table 4.3: Testing combination for noisy training data

Base 1	single noisy training
Base 2	Iteration 1, noisy bag training
Base 3	Iteration 3, noisy bag training
Testing 1	single aligned gallery, single aligned probe
Testing 2	single aligned gallery, single noisy probe
Testing 3	aligned bag gallery, noisy bag probe

Table 4.4: Results comparison

	testing 1	testing 2	testing 3
base 1	0.3213	0.1941	0.5431
base 2	0.9540	0.9364	0.9766
base 3	0.9690	0.9590	0.9833

observations:

- When we use single noisy training image without adding perturbations (base 1), the recognition rate is very low for all the testing combinations. This indicates that the within-subject scatterness for poorly registered face in the training set is so high that they overlap with other subjects' clusters and lead to confusion for the classifiers. For Fisherface, it means the objective function it tries to minimize is ill-conditioned, which will lead to the failure of the algorithm.
- For base 2 case, we augment the noisy images by adding perturbations to generate noisy bags, then the recognition rate increases greatly compared to using noisy images directly.
- Base 3 shows that it's not good to treat all the instances from the noisy bags as the same. We used our multiple-instance based subspace learning method to remove those "bad" instances from the augmented noisy bags. The resulting training set increases the discriminative power of the classifier, but not to disperse the within subject cluster and cause confusion.

- Given only noisy training and probe set, we still achieved much higher recognition rate of 98.33% than the baseline algorithm of 92.47% as shown in table 4.2, and roughly the same as the optimal case of 98.49%, where all noisy bags are generated by perturbing the aligned images.

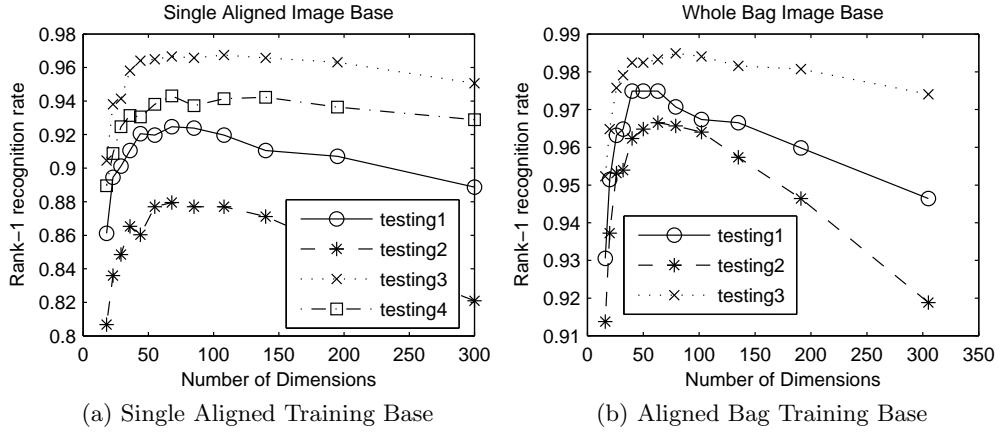


Figure 4.3: Recognition Rate Change for FisherFace w.r.t single aligned training base 4.3a, aligned bag training base 4.3b

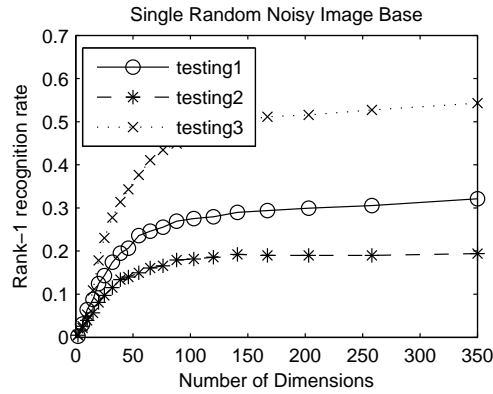


Figure 4.4: Single Noisy Training Base

Figures 4.3a shows testing results with single aligned image as training data. Figure 4.3b shows testing results with aligned bag as the training data. Both figures show the change of recognition rate w.r.t. the change of the number of dimension used by FisherFace. In both cases, the recognition rate has the following order: testing 3 > testing 1 > testing 2, where all the testings have the same meaning as explained in table 4.1.

Figure 4.4 shows how noisy training images could affect the recognition rate. It's obvious that when the training set is not aligned very well, all the testing cases fail, including using probe bags and gallery bags. So it's very important to remove noisy training images from corrupting the training subspace.

Figure 4.5a, 4.5b and 4.6 show recognition error rates on three different testing combinations. The testings have the same meaning as explained in table 4.3. Optimal 1 means training with aligned bags, and optimal 2 means training with aligned single images. Iter1 and Iter3 means the first iteration and the 3rd iteration of the base selection procedure. We can see that in all cases, the 3rd iteration results is better than the 1st iteration results. It supports our claim that extremely poorly registered images will not benefit the learning algorithm. We use our multiple-instance learning algorithm to exclude those bad training images from corrupting the training base. Also interestingly, in all tests, optimal 1 always performs worst, which indicates that by adding perturbations to the training base, even very noisy images, we can improve the robustness of learning algorithms. Note that in all cases, when the number of dimensions increases, the error rate will first decrease and then increase. Normally we get the best recognition rate using around the first 50 dimensions (account for 70% of total energy).

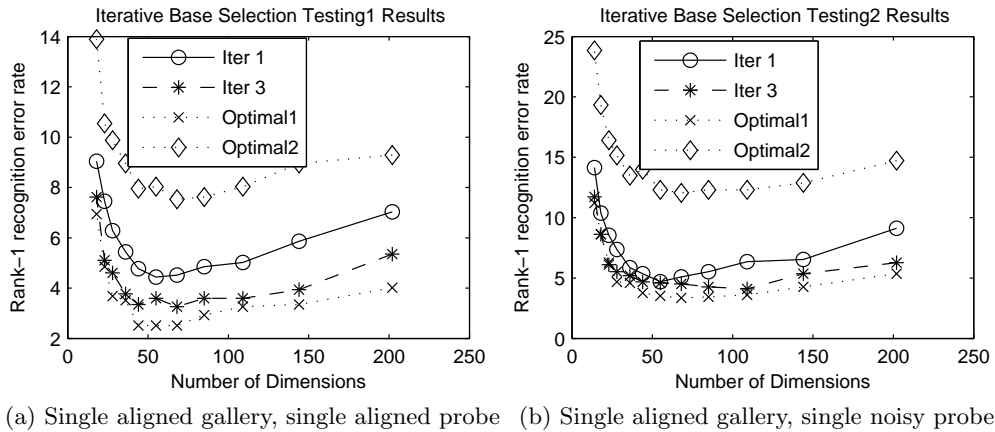


Figure 4.5: Recognition Rate Change for FisherFace w.r.t Single aligned gallery, single aligned probe 4.5a, Single aligned gallery, single noisy probe 4.5b

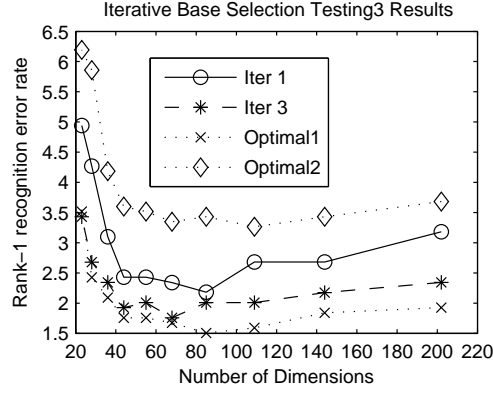


Figure 4.6: Aligned bag gallery, noisy bag probe

For experiments on facial expression recognition, we used the publicly available CMU Cohn-Kanade facial expression database [55]. This AU coded database consists of 100 university students aged from 18 to 30 years, of which 65% are female, 15% are African-American, and 3% are Asian or Latino. Videos were recorded using a camera located directly in front of the subject. Subjects are instructed to perform a series of facial expressions. Each expression sequence starts from neutral and ends with expression apex. Before performing each expression, an experimenter described and modeled the desired expression. In each image sequences the student performs an expression, starting from neutral to expression apex. The videos are stored into 640 by 480 pixel arrays with 8-bit precision for grayscale values. Each video sequence varies in length from 8 to 65 frames. We discarded some of the sequences that do not belong to any of the six commonly studied emotions and the resulting 365 sequences come from 97 subjects, including 30, 43, 52, 96, 69 and 75 sequences of anger, disgust, fear, happiness, sadness and surprise, respectively. There are about 1 to 9 sequences for each subject. The face images were cropped to size of 64x64 pixels. Besides the large variances of the facial expressions, the selected data set also contains considerable lighting variations.

The testing procedure is also not as complex as in face recognition. Multinomial logistic regressor is used as the classifier. We also tried to use nearest neighbor as

Single Aligned Training Base							Aligned Bag Training Base						
	A G	D G	F A	H P	S D	S P		A G	D G	F A	H P	S D	S P
AG	14	2	3	5	6	0	AG	15	3	1	3	6	2
DG	4	36	0	2	1	0	DG	3	37	0	0	3	0
FA	2	2	37	8	3	0	FA	0	3	38	7	4	0
HP	3	0	6	84	3	0	HP	1	1	6	85	1	2
SD	4	3	0	3	58	1	SD	5	2	1	0	57	4
SP	0	1	0	1	2	71	SP	0	0	0	1	1	73

Single Noisy Training Base							Noisy Bag Training Base						
	A G	D G	F A	H P	S D	S P		A G	D G	F A	H P	S D	S P
AG	18	3	1	2	6	0	AG	17	3	2	2	5	1
DG	6	32	2	1	2	0	DG	8	31	2	1	1	0
FA	1	1	32	10	7	1	FA	1	4	30	9	7	1
HP	3	0	7	84	1	1	HP	3	0	6	84	2	1
SD	4	1	4	3	54	3	SD	3	3	3	3	54	3
SP	0	2	3	4	2	64	SP	1	3	3	2	3	63

Table 4.5: Confusion matrices for different training bases. The traces for the matrices are 300 (82.2%), 305 (83.6%), 284 (77.8%) and 279 (76.4%)

the classifier, but the results are nearly as bad as random guess for all the different training bases. Results shown in the confusion matrix 4.5 are based on the multinomial logistic regressor. Table 4.6 shows the confusion matrix for the features selected by the multiple-instance procedure from noisy bags (bag images generated by perturbing noisy input images). From the confusion matrix we can see that the performance is almost as good as perfectly aligned case, and it improves a lot comparing with using noisy bags directly without multiple instance selection. In table 4.5, AG denotes anger, DG denotes disgust, FA denotes fear, HP denotes happiness, SD denotes sadness and SP denote surprise. All the results are based on 5-fold cross validation. The training and testing images are only the last frame of each sequence, i.e., the apex expression. The noisy image are generated by adding Gaussian random noises of $N(5, 0; 0, 5)$ to the two eye positions. The noisy bag images are generated by adding ± 2 pixel to the two perturbed eye positions. The positions of the two eyes are used to crop face images to the size of 176x144.

Multiple-instance selected noisy bag

	A	D	F	H	S	S
	G	G	A	P	D	P
AG	16	2	3	3	6	0
DG	4	35	1	2	1	0
FA	2	2	38	8	2	0
HP	3	3	5	81	4	0
SD	4	2	0	3	58	2
SP	1	2	1	1	0	70

Table 4.6: Confusion matrices for multiple instance selected noisy bag images, trace for the matrix 298 (81.6%)

Note that there are some differences between the experimental results for face recognition and expression recognition. Even though our algorithm improves both face recognition and facial expression recognition rates, the improvement for expression recognition is not as obvious as for face recognition. Nearest neighbor classifier works well for face recognition, but fails with expression recognition. Compared with the feature we used in chapter 5, we can see that the holistic subspace based feature is not effective for facial expression recognition. One obvious reason is that it doesn't take into account the facial movement information.

4.1.3 Conclusions and Discussions

In this section, we study the influence of image mis-alignment on face recognition and facial expression problems. We then propose a feature representation scheme based on the multiple-instance learning idea. We propose a multiple-instance learning scheme for subspace training. The algorithm proceeds by iteratively updating the training set. Simple subspace method, such as FisherFace and EigenFace, when augmented with the proposed multiple-instance learning scheme, achieved very high recognition rate. Experimental results show that even with the noisy training and testing set, the features extracted by our multiple-instance learning scheme achieves even higher recognition rate than baseline algorithm where the training and testing images are

aligned accurately. Our algorithm is a meta-algorithm which can be used with other methods.

The algorithm works with the following assumptions: for face recognition, we assume for the same face, when aligned well, they lie near to each other in the subspace; for expression recognition, we assume for the same expression, when aligned well, they lie near to each other in the subspace. When these assumptions are not satisfied, the algorithm will not be as effective. When the assumptions are not valid, we could build a non-linear manifold representation for the data and use the distances on manifold to generate the minimum spanning tree. However, in practice, due to locality geometry preservation, manifold representation is sensitive to noises in data. In our experiments for face recognition and expression recognition, the Euclidean distance worked well.

4.2 With Good Face Alignment

The image features can be classified into two categories: shape features and appearance features. Normally shape features are represented as parameters for deformable models, for example, ASM or 3D deformable model. On the other hand, appearance features include intensity of the facial image, dense optical flow, parameterized optical flow, Gabor features, Haar-like features and Local Binary Pattern (LBP) feature etc. Feature selection is one of the basic problems in pattern recognition and machine learning. Usually, given a large amount of features, we need to select the most discriminative features and remove those irrelevant features in order to avoid the curse of dimensionality and improve generalization. Feature selection methods can be classified into the following categories: filter, wrapper and embedded. In filter based approach, features are selected according to the rankings of the statistical properties of individual features without referring specific classifier. In other words, in filter based feature selection methods, the feature selection step and classifier learning step are separated.

Example filters include signal to noise ratio, univariate association with target variable, mRMR etc. For wrapper based approach, specific classifier is used to measure performances of possible subsets of features and the best subsets would be chosen based on the performance. In other words, the features and the classifier are coupled tightly. Example methods include the SVM recursive feature elimination. For embedded feature selection, we don't consider all the possible subsets of the features, instead the feature selection and model parameter are simultaneously chosen. Example methods include the AdaBoost. One need to note that no single feature selection scheme is best for all data distributions. In practice, in order to get optimal or near optimal feature set, one need to perform different feature selection scheme based on trial and error.

In this work, we empirically compare commonly used features and discuss different feature selection methods, such as AdaBoost [45], Bayesian Multinomial Logistic Regression [75] and minimum-Redundancy-Maximum-Relevance (mRMR) [99] etc. We also give AU recognition experimental results on a large spontaneous facial expression data set.

4.2.1 Representative Features and Feature Selection Methods

In this part, we empirically study the Haar-like features, Gabor features and the Local Binary Pattern features for expression analysis.

Haar-like Features

In our facial expression recognition system, we used three different kinds of features: two-rectangle, three-rectangle and four-rectangle features. Figure 4.7 shows some example Haar-like features. The value of a two-rectangle feature is the difference between the sum of intensities of the pixels within two rectangular regions. The regions have the same size and shape and are horizontally or vertically adjacent. A three-rectangle

feature computes the sum within two outside rectangles subtracted from the sum in a center rectangle. The four-rectangle feature computes the difference between diagonal pairs of rectangles. The set of Haar-like features are complete, i.e., for a 24x24 image, the total feature size is 160,000, which is far larger than the number of pixels in the image. Fortunately, the Haar-like feature could be easily computed using the integral image. With this kind of huge number of image features, it's important to further select the discriminative features for classification purpose.

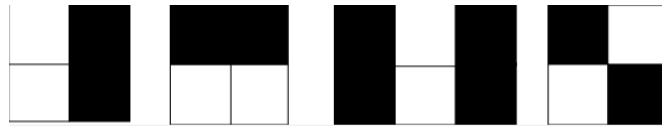


Figure 4.7: Example Haar-like feature

Gabor Features

Gabor filter responses are widely and successfully used as general purpose features in many computer vision tasks, such as in texture segmentation, face detection and recognition, and iris recognition. A Gabor filter is a linear filter whose impulse response, thus the system characteristic, is defined by a harmonic function multiplied by a Gaussian function in the following form

$$g(x, y, f, \theta) = \exp^{-\frac{1}{2}(\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2})} * \exp^{j2\pi f x_\theta} \quad (4.1)$$

$$\begin{cases} x_\theta = x \cos \theta + y \sin \theta \\ y_\theta = -x \sin \theta + y \cos \theta \end{cases} \quad (4.2)$$

where f is the frequency of the sinusoidal wave, θ is the orientation of the Gabor filter, σ_x and σ_y are the standard variations along x and y directions respectively. In a typical feature construction the Gabor filters are utilized via multi-resolution structure, consisting of filters tuned to several different frequencies and orientations. The frequency

and orientation representations of Gabor filter are similar to the primary visual cortex of human beings, and it has been found to be particularly appropriate for texture representation and discrimination. Gabor filter's main weakness is its computational complexity. Suppose the convolutional kernel size is k , and the image size is N , the computational complexity is $O(k^2 * N^2)$ when it's non-separable, and $O(k * N^2)$ when it's separable. When the Gabor filter is separable, it can be separated into 1D band-pass and 1D low-pass to the perpendicular. The Gabor filter is only separable when θ is $k * \pi/2$. For other values of θ , in order to use separable Gabor filter, pixel interpolation is required [6]. It prevents its use in many real-time or near real-time tasks, such as object tracking. Since Gabor filters correspond to any linear filters the most straightforward technique to perform the filtering operation is via the convolution in the spatial domain, or alternatively it could be performed in the frequency domain with a complexity of $O(N^2 * \log N)$ due to the FFT transform. The standard convolution with Gabor filters can be improved by utilizing the separability of Gabor filters for reducing the number of needed multiplications and additions. Figure 4.8 shows some example gabor filters. This Gabor filter bank consists of 4 different orientations and 4 different scales. Figure 4.9a and Figure 4.9b show one facial expression image before and after convolution with the Gabor filters.

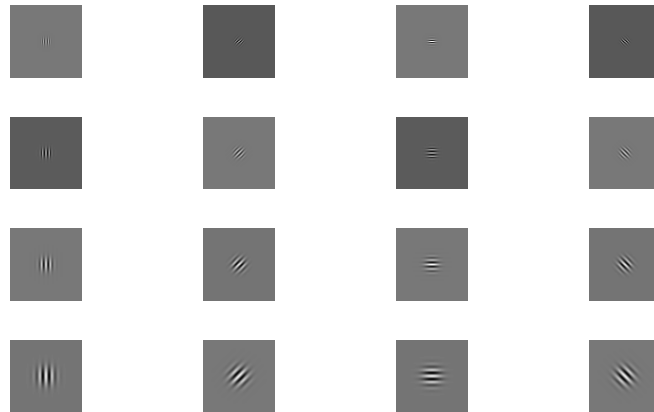


Figure 4.8: Example Gabor filters with 4 orientations and 4 different scale

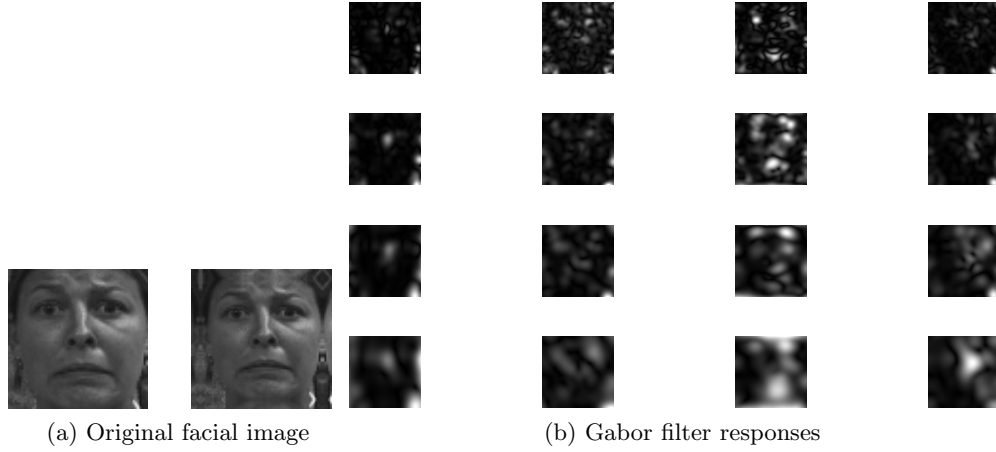


Figure 4.9: Example facial expression image before 4.9a and after 4.9b convolution with Gabor filters

LBP Features

The LBP operator is defined as a gray-scale invariant texture measure, derived from a general definition of texture in a local neighborhood. Through its recent extensions, the LBP operator has been made into a really powerful measure of image texture, showing excellent results in many empirical studies. The most important property of the LBP operator in real-world applications is its invariance against monotonic gray level changes. Another equally important is its computational simplicity, which makes it possible to analyze images in challenging real-time settings. Figure 4.10 shows the process of applying LBP operator on one facial image. After the LBP encoding of an image, the LBP codes are collected into a histogram. The classification is then performed by computing histogram similarities. Since the histogram will lose the relative position information, the facial images are first divided into several local regions from which LBP histograms are extracted and concatenated into an enhanced figure histogram. The χ^2 distance is used for similarity measurement between different histograms.

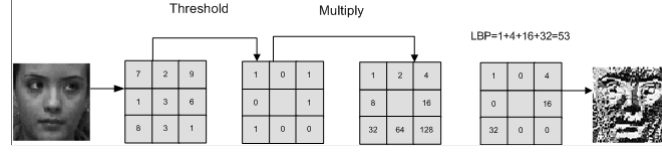


Figure 4.10: Image processing with LBP operator

4.2.2 Feature Selection

In this section, we compare some popular feature selection methods in conjunction with the different features we discussed before.

Bayesian Multinomial Logistic Regression

Madigan et al. [75] proposed the Bayesian multinomial logistic regression method for feature selection and applied the technique on author identification on large-scale dataset. The Bayesian logistic regression was used with a Laplace prior to avoid overfitting and produced sparse predictive models for text data. We will briefly review this method in this thesis and apply it on facial expression recognition problem.

let $x = [x_1, \dots, x_j, \dots, x_d]^T$ be a vector of feature values characterizing a facial expression image to be identified. We encode the fact that a facial expression image belongs to a class (e.g. smile) $k \in \{1, \dots, K\}$ by a K -dimensional 0/1 valued vector $y = (y_1, \dots, y_K)^T$, where $y_k = 1$ and all other coordinates are 0.

Multinomial logistic regression is a conditional probability model of the form

$$p(y_k = 1 | x, B) = \frac{\exp(\beta_k^T x)}{\sum_{k'} \exp(\beta_{k'}^T x)} \quad (4.3)$$

parameterized by the matrix $B = [\beta_1, \dots, \beta_K]$. Each column of B is a parameter vector corresponding to one of the classes: $\beta_k = [\beta_{k1}, \dots, \beta_{kd}]^T$. This is a direct generalization of binary logistic regression to the multiclass case.

Consider a set of training examples $D = (\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_i, \mathbf{y}_i), \dots, (\mathbf{x}_n, \mathbf{y}_n)$. Maximum

likelihood estimation of the parameters B is equivalent to minimizing the negated log-likelihood:

$$l(B|D) = - \sum_i \left[\sum_k y_{ik} \beta_k^T \mathbf{x}_i - \ln \sum_k \exp(\beta_k^T \mathbf{x}_i) \right] \quad (4.4)$$

As with any statistical model, we must avoid overfitting the training data for a multinomial logistic regression model to make accurate predictions on unseen data. One Bayesian approach for this is to use a prior distribution for B that assigns a high probability that most entries of B will have values at or near 0. Sparse parameter estimates can be achieved in the Bayesian framework remarkably easily if we use double exponential (Laplace) prior distribution on the β_{kj} :

$$p(\beta_{kj}|\lambda_{kj}) = \frac{\lambda_{kj}}{2} \exp(-\lambda_{kj}|\beta_{kj}|) \quad (4.5)$$

The prior for B is the product of the priors for its components. For typical data sets and choices of λ 's, most parameters in the MAP estimate for B will be zero. The MAP estimate minimizes:

$$l_{lasso}(B|D) = l(B|D) + \lambda_{kj} \sum_j \sum_k |\beta_{kj}| \quad (4.6)$$

Equation 4.6 is convex, but it does not have a derivative at 0; we will need to take special care with it. Coordinate descent algorithm was used to fit Bayesian multinomial logistic regression model.

Boosting for Feature Selection

Bagging (Bootstrap aggregating) and boosting are algorithms to learn an ensemble of classifiers. It belongs to the more general ensemble learning framework bagging, boosting, AdaBoost, stacked generalization and hierarchical mixture of experts; combination rules: voting strategy, where the objective is to build an ensemble that is as diverse as

possible. Individual classifiers that make up the ensemble are combined in such a way that the correct decision are amplified, and incorrect ones are canceled out. Reasons to use ensemble based system are: more classifiers are just more robust in many cases; too large volume of data (divide the data into subset and for each subset train a classifier); too little data (using bagging [18] etc); divide and conquer (regardless of the amount of data available, certain problems are just too difficult for a given classifier to solve); data fusion (several data sources where the nature of features are different, a single classifier cannot be used to learn the information contained in all of the data). The AdaBoost algorithm [105, 45] was frequently used in machine learning algorithms for feature selection. Ada, short for Adaptive Boosting. AdaBoost generates a set of hypotheses, and combines them through weighted majority voting of the classes predicted by the individual hypotheses. The hypotheses are generated by training a weak classifier, using instances drawn from an iteratively updated distribution of the training data. This distribution update ensures that instances misclassified by the previous classifier are more likely to be included in the training data of the next classifier. Hence, consecutive classifiers' training data are geared towards increasingly hard-to-classify instances.

Freund and Schapire [45] proved the upper bound for training error and generalization error. The upper bound for the generalization error suggests that AdaBoost does have a problem of over fitting, as all other learning algorithms. But empirically, many people have found that even after thousands of iterations, the generalization error continues to decrease. Even after the training error has decreased to 0, the testing error continues to decrease. Schapire et al. [106] gave an alternative analysis in terms of the margins of the training examples, which finally related to the support vector machines.

Algorithm 3 shows the steps of classical boosting process for binary classification task:

The equation to update the distribution D_t is constructed so that:

Algorithm 3 The Boosting algorithm for binary classification

- 1: Given $(x_1, y_1), \dots, (x_m, y_m)$, where $x_i \in X$, $y_i \in Y = \{-1, +1\}$
 - 2: Initialize $D_1(i) = \frac{1}{m}, i = 1, \dots, m$.
 - 3: **for** $t = 1, \dots, T$: **do**
 - 4: Find the classifier $h_t : X \rightarrow \{-1, +1\}$ that minimizes the error with respect to the distribution D_t :
 $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j$, where $\epsilon_t = \sum_{i=1}^m D_t(i)[y_i \neq h_t(x_i)]$
 - 5: Prerequisite: $\epsilon_t < 0.5$, otherwise stop.
 - 6: Choose $\alpha_t \in R$, typically $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$ where ϵ_t is the weighted error rate of classifier h_t
 - 7: Update: $D_{t+1}(i) = \frac{D_t(i)e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$, where Z_t is a normalization factor (chosen so that D_{t+1} will be a probability distribution)
 - 8: **end for**
 - 9: Output the final classifier: $H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$
-

$$e^{-\alpha_t y_i h_t(x_i)} \begin{cases} < 1, & \text{if } y(i) = h_t(x_i) \\ > 1, & \text{if } y(i) \neq h_t(x_i) \end{cases}$$

After selecting an optimal classifier h_t for the distribution D_t , the examples x_i that the classifier h_t identified correctly are weighted less and those that it identified incorrectly are weighted more. Therefore, when the algorithm is testing the classifiers on the distribution D_{t+1} , it will select a classifier that better identifies those examples that the previous classifier missed.

Figure 4.11 shows the most discriminative Haar-like features selected by the AdaBoost algorithm. From the figure we can see that most of the selected features do indeed correspond to the interesting area.



Figure 4.11: Example selected Haar-like features by AdaBoost

Feature Selection Based on Mutual Information

Features can be selected in many different ways. One scheme is to select features that correlate strongest to the classification variable. This has been called maximum-relevance selection. On the other hand, features can be selected to be mutually far away from each other, while they still have "high" correlation to the classification variable. This scheme, termed as minimum-Redundancy-Maximum-Relevance selection (mRMR) [99], has been found to be more powerful than the maximum relevance selection.

As a special case, the "correlation" can be replaced by the statistical dependency between variables. Mutual information can be used to quantify the dependency. In this case, it is shown that mRMR is an approximation to maximizing the dependency between the joint distribution of the selected features and the classification variable.

4.2.3 Comparison of Different Facial Features and Feature Selection Methods

We first use the Cohn-Kanade dataset [55] for baseline comparison, then we choose to use the best performer only for further AU recognition on a large scale spontaneous facial expression dataset. Only the first frame and the last frame of each video sequence are used for static testing. For all the testings, we used 5-fold cross validation.

For the Haar-like feature, we constrain the minimum feature size to be 5x5 pixels, the maximum feature size to be 20x20 pixels, the step size for moving feature window is 3 pixels and the step size for increasing window size is 2 pixels. This greatly reduced the number of total possible feature combinations. For the Gabor feature, we constructed a Gabor filter bank of 4 different orientations and 4 different scales, as shown in figure 4.8, this will also create a lot of features since each pixel would have 16 responses from the Gabor filter bank. For the LBP feature, we construct the local histograms and

Feature	Recognition rate
Haar	89.4
Gabor	88.2
LBP	87.6

Table 4.7: Comparison of different features for expression recognition

Feature Selection Method	Recognition rate
BMR	86.6
AdaBoost	89.4
mRMR	82.1

Table 4.8: Comparison of different features selection methods for face recognition (Haar feature)

concatenate to form a global histogram for each image. We then perform the AdaBoost feature selection algorithm on the above mentioned features and got the results shown in table 4.7:

From table 4.7 we can see that the three features got very similar recognition rate. Due to fast computation of the Haar-like feature, we chose to use the Haar feature for further feature selection analysis. Table 4.8 shows performances for different feature selection methods on the Haar feature. BMR stands for Bayesian Multinomial Logistic Regression.

From our experiments we observed that AdaBoost feature selection in conjunction with the Haar feature got the best recognition results.

Based on the FACS theory [36], each facial expression is a combination of multiple lower level AUs. We tested the AdaBoost selected Haar feature on our facial AUs database. This data set has spontaneous facial expressions from freely behaving individuals. The data is particularly hard to analyze since it has large off-plane head rotations and very subtle AU displays. The images in the data set are of 640 by 480 in size. There are 33 subjects in the data set, and each subject has around 10,000 frames. The

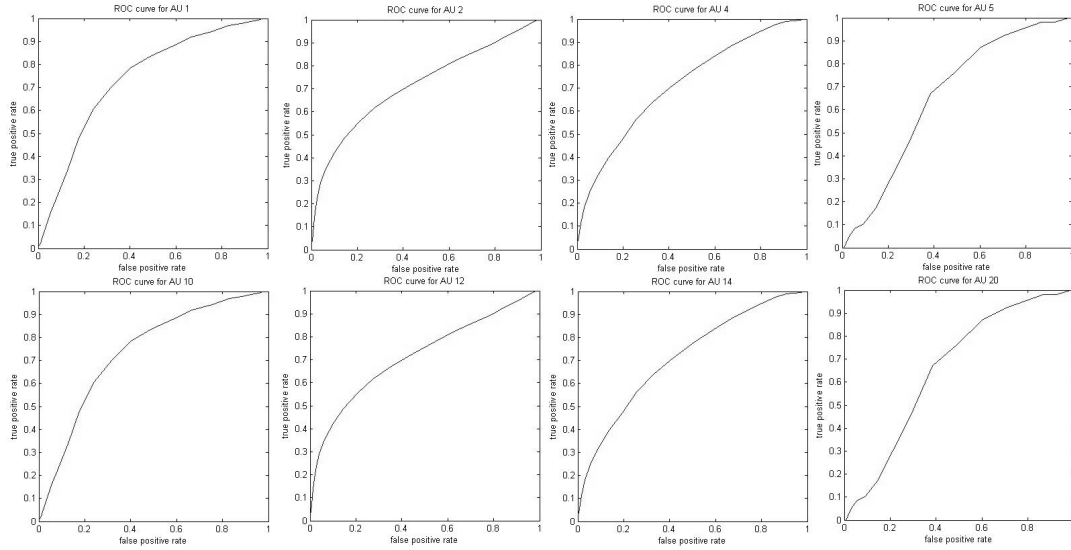


Figure 4.12: ROC curves for major AUs on NSF dataset

ground truth for AUs are labeled carefully by psychologists. We focus on the AUs which are most important for emotion detection purposes. We used 8 AUs, AU1, AU2, AU4, AU5, AU10, AU12, AU14 and AU20 in our experiments. Figure 4.11 shows example selected Haar features by the AdaBoost algorithm. We randomly select 22 subjects as the training set, and the other 11 subject as the testing set. We use the ROC curve as the measurement for AU recognition performances. Figure 4.12 shows the ROC curves for the eight AUs. The area under the ROC curves are 0.79, 0.66, 0.67, 0.76, 0.70, 0.84, 0.68 and 0.77 respectively. Considering the data set is extremely difficult, the performance is acceptable. The Haar feature we used is static. In [138], better results are reported due to the use of boosted dynamic feature. As introduced in related work section, using AdaBoost feature selection for other types of spatial-temporal features would be very interesting.

Chapter 5

Sequential Facial Expression Modeling

A key to progress in computer vision is to find theoretically solid models that are computationally tractable. Both generative models and discriminative models have been applied for facial expression analysis tasks, both in static and dynamic settings. In static setting, the most popular generative model is the naive Bayes classifier. Well known classifiers like SVM, decision trees and perceptrons, on the other hand, belong to the discriminative model. In dynamic setting, the most popular generative model is the Hidden Markov Model (HMM) [103]. The Maximum Entropy Markov Model (MEMM), Conditional Random Fields (CRF) and Discriminative Random Field (DRF) [63] are the most popular discriminative spatial-temporal models. In this section, we will first compare the characteristics of generative and discriminative models, and then we introduce the training and inference algorithms for HMM, CRF and HCRFs, finally we will compare generative models and discriminative models for video based facial expression recognition task. We consider casting the problem as a classification task over an image sequence (usually with different lengths). That is, the aim is to determine the class label specifying the emotion of a given sequence. Even though our experimental results are one of the best up to date, our goal is not to find optimal features and representations for solving a specific object recognition task, but rather to fix on a particular, widely used, feature set and use this as the basis to compare alternative learning methodologies.

The first task in sequential classification is to choose appropriate representation for the input sequence, i.e., feature extraction and selection. In temporal domain, since

natural facial events are dynamic and evolve over time from onset to apex and finally to offset, features that only consider static shape and/or appearance information are often not as effective as dynamic features. Bag-of-words inspired dynamic features usually lose the temporal order of dynamic features, which could be crucial for the sequence classification tasks. In spatial domain, part-based representations have proven to be effective [42] for object detection and recognition due to the fact that they can cope with partial occlusions, clutter and geometric transformations. An ideal representation should be based on dynamic features, and keep the temporal and spatial structure information in the image. In this work, we first use regular grid to divide the aligned face image into small patches (simulating facial parts without actually detecting them), and then extract dynamic features from each of these patches. Each image is represented by concatenating local dynamic features, and each video sequence is then represented by a series of such feature vectors.

Having obtained a series of feature vectors, the different lengths of the sequences present a challenging problem to many traditional classification techniques. Dynamic time warping type of techniques also have difficulties with these sequences because of the high-dimensional continuous-valued features. There are typically three strategies to tackle this problem, depending on the ways in which the feature vectors from individual frames are organized. The first type of approaches classify each frame individually and label the whole sequence by majority voting. This way the sequence classification problem reduces to the traditional classification problem. In these frame-based classification approaches, the temporal dependencies among the frames are missing from the modeling perspective, but can be compensated by using dynamic features. The temporal order information would still be lost unless the dynamic features are extracted over long periods. The second type of approaches are based on sequence tagging, such as in Named Entity Recognition (NER) and Part-Of-Speech (POS) tagging. Similar to the

frame-based classification, each frame also gets its own label, and votes to decide the sequence label. However, the labels of the frames are not determined independently, but rather in a structured way. Finally, there are sequence classification approaches that can classify a sequence without labeling its individual frames. Like sequence tagging, these approaches also explicitly exploit the temporal dependencies among the frames. Some latent variables are often required to bridge the input sequence and the output label, and handle the complex intrinsic dependency structure. These approaches are especially useful when labeling a single frame is not meaningful (e.g., video-based human action recognition), or the labeling tasks in the training stage are time-consuming, tedious, and expensive. Facial expression recognition is one of the tasks that can be and have been approached from all three strategies (namely, frame classification, sequence tagging, and sequence classification), because a single image can still carry a facial expression.

We are interested in the relationships between these approaches and especially their performances on the facial expression analysis problem. In the rest of this chapter, we first propose a part-based dynamic feature representation which keeps both spatial and temporal information; we show a unified view of different sequential data learning algorithms, both generative and discriminative, and investigate their differences from various aspects such as their feature functions, objective functions and optimization strategies; we couple the proposed part-based feature representation with these algorithms for the expression recognition task; and finally we report some of the best expression recognition performances up to date on a publicly available data set.

5.1 Background on Graphical Models

The models we consider in this work are special cases of general graphical models. Specifically, we use linear chain structure to represent image sequences. Graphical

models provide a general framework for modeling and solving problems that involve a large number of random variables which are correlated in complex ways. A graphical model is specified by a graph where each node denotes a random variable and edges connecting two nodes represent conditional dependence structure between random variables. In particular, graphical model can be thought of a tool for intuitive modeling and efficient computation of the joint distribution or marginal distribution for a large number of random variables by considering conditional independence constraints among the random variables encoded by the graph structure. According to the property of edges in graphical model, they can be classified as either undirected or directed. Directed acyclic graphical (DAG) models are also known as Bayesian networks, and undirected graphical models are also commonly known as Markov Random Field (MRF). Directed graph can be converted into undirected graphs by moralization and therefore all the training and inference algorithms developed for undirected graphs could also be applied to directed graphical models. For undirected graphical models, the Hammersley-Clifford theorem states that the conditional independence assumptions hold if and only if the distribution factors as the product of potential functions over cliques. Classical static graphical models include Naive Bayes (NB) 5.1a, Logistic Regression (LR) 5.1b, and their temporal extensions, the Hidden Markov Model (HMM) 5.2a, a special case of Dynamic Bayesian Networks (DBN) and the Conditional Random Fields (CRF) 5.2b. They are simple and special cases of general graphical models.

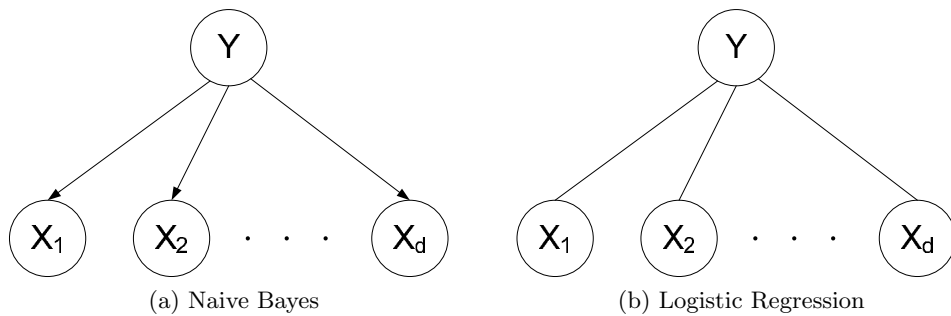


Figure 5.1: Frame-wise Classification

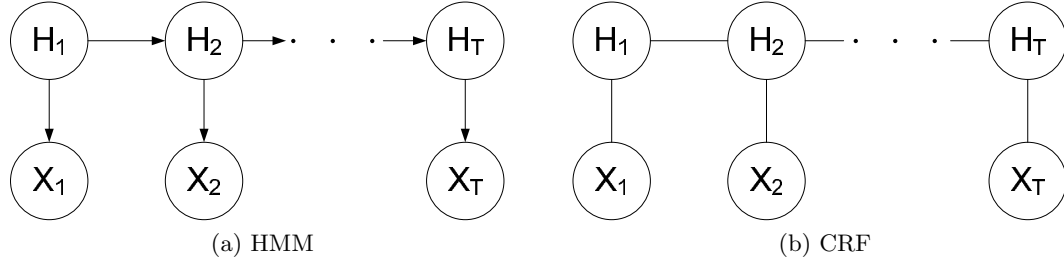


Figure 5.2: sequence tagging

5.1.1 Inference

Inference is the process of calculating conditional probability distribution over unobserved random variables given observed random variables, which are also called evidences. Inference algorithms could be classified as either exact or approximate methods.

Exact Inference

Exact inference algorithms [107, 53, 62] take advantage of the conditional independencies present in the joint distribution as inferred from the missing edges in the graph. One of the exact method for inference is the belief propagation algorithm developed by Pearl [98]. Belief propagation, as its name suggests, is a message passing algorithm. It operates on factor graph. Either directed or undirected graphical models can be represented as factor graphs. It calculates the conditional distribution for each unobserved node, conditional on any observed nodes. The algorithm was first proposed by Judea Pearl in 1982 [97], who formulated this algorithm on trees, and was later extended to polytrees. It has since been shown to be a useful approximate algorithm on general graphs. The algorithm is also commonly known as the sum-product algorithm. The algorithm performs exact inference on tree structured factor graphs, and it terminates after 2 iterations. In the first iteration, the algorithm starts by choosing one node as root, then the algorithm starts message passing from leaf nodes toward the root node.

In the second iteration, the algorithm passes messages from the root node toward the leaf nodes. Upon completion, the marginal distribution of each node is the product of all messages from neighboring factors. The joint distribution of the set of variables belonging to one factor is the product of the factor and the messages from the variables.

Another exact method is the junction tree algorithm. The junction tree algorithm is used for exact marginalization in general graphs. In essence, it performs belief propagation on a modified graph called a junction tree. The basic premise is to eliminate cycles by clustering them into single nodes. Only local computations are needed to perform inference in the junction tree. The junction tree algorithm proceeds as follows:

- If the graph is directed graph, then use moralization to convert it into undirected graph
- Triangulating the graphs to make it chordal. A graph is chordal if each of its cycles of four or more nodes has a chord, which is an edge joining two nodes that are not adjacent in the cycle. An equivalent definition is that any chordless cycles have at most three nodes.
- Identifying maximal cliques.
- Message passing between cliques. At most $2N-1$ message passing required (N cliques, $N-1$ separator sets), where N is the number of nodes. Each message passing step requires marginalization, which for the case of conditional probability tables (CPT) is exponential in the size of the largest clique.

Sometimes it's desirable to only acquire the most probable state instead of computing the marginal probability distributions. The sum-product algorithm could be easily changed into max-product algorithm which solves the maximization problem. Popularly used forward-backward and Viterbi algorithms for HMM are special cases of the sum-product and max-product algorithm respectively.

Approximate Inference

Sometimes it's not necessary to have exact inference if approximate solution is accurate enough, or it's simply too computationally expensive to have exact solution. Approximate inference for graphical models are more commonly used in practice. Example approximate inference algorithms include loopy belief propagation, Monte Carlo estimation and variational methods for graphical models.

Loopy Belief Propagation: When performing nearly the same belief propagation algorithm on general graphs as in trees, it's called loopy belief propagation. Even though currently it's still unclear under which condition the loopy belief propagation procedure will converge, it has been widely used in practice due to empirical success. The assumptions made by BP is valid only for acyclic graphs. The BP is called loopy BP when applied on graphs containing circles, and the convergence is generally not guaranteed. Since in a general graph structure, there is not necessarily any leaf node, the procedure will change a bit. The algorithm initializes all variable messages to 1, and update all messages at each iteration. For tree structure, the modified algorithm converges after a number of iterations equal to the diameter of the tree.

Monte Carlo Estimation [89]: The advantages of Monte Carlo sampling algorithm include their simplicity of implementation and theoretical guarantees of convergence. The disadvantages of the Monte Carlo approach are that the algorithms can be slow to converge and it can be hard to diagnose their convergence.

Variational Methods: Variational methods [54] exploit laws of large numbers to transform the original graphical model into a simplified graphical model in which inference is efficient. Inference in the simplified model provides bounds on probabilities of interest in the original model. The basic intuition underlying variational methods is that complex graphs can be probabilistically simple; in particular, in graphs with dense

connectivity there are averaging phenomena that can render nodes relatively insensitive to particular settings of values of their neighbors.

5.2 Facial Feature Representation

Most spatio-temporal features proposed in the literature [65, 91, 58, 141, 137] are based on densely sampled or salient volumetric features. One of the problem with volumetric feature is that it loses temporal resolution. Instead of using volumetric features, we propose to use part-based optical flow, which is simple but effective. Part-based representation has been shown effective [42] due to the fact that it can cope with partial occlusion. Features based on pixel intensities performed poorly in our testing, mainly due to changes in the appearance of the subjects and lighting conditions. This motivates us to use dynamic feature. Optical flow is the most popular way to approximate 2D motion field from spatio-temporal patterns of image intensity.

The Kanade-Lucas-Tomasi (KLT) [110] is one of the most popular optical flow algorithm for its accuracy and robustness [9]. KLT assumes constant velocity within a local neighborhood, and it tracks points by computing the sum of square difference between two small windows. When computing the difference, larger weight is given to the center than the periphery. KLT selects good features based on magnitudes of spatial gradients and computes sparse flow, which is computed by weighted least square in close-form.

Regular grid is used to divide the whole face image into 30 (5x6) small patches, as shown in figure 5.3. Within each cell, we separate the optical flow into horizontal and vertical components and compute average of each component. Motion for each cell is represented by the average flow. Since motion computed from optical flow is unavoidably noisy, certain amount of averaging or histogramming alleviates the problem. Each image is represented by concatenating average flows from all the local patches.

Note that the flow in our experiments is not dense but robust enough to capture local motions. Unlike the space-time interest points [65], which would be too sparse for this purpose. On the other hand, dense flow would prevent the system from real-time applications. Figure 5.4 shows the average flow for six different facial expressions. From the figure one can see it is hard to model such observations with mixture of Gaussians, assumed by the generative models. To compare the performances of different algorithms more thoroughly, principle component analysis (PCA) is used for feature dimension reduction. As we will show in the experiments, generative models perform relatively better on dimension-reduced features.



Figure 5.3: Regular grid on facial images

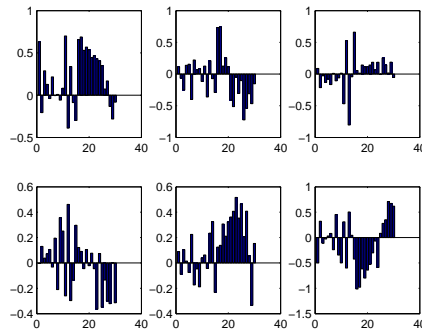


Figure 5.4: Histogram of mean flow for different expressions

The feature representation scheme proposed in chapter 4 is based on static feature. As we will show in the experimental results section, the combination of static features with dynamic modeling causes some interesting properties. Future work could extend the static features to dynamic features.

5.3 Sequential Data Classification algorithms

5.3.1 The problem: a unified view

Given a set of k training sequences $\mathbf{X} = \{\mathbf{x}^{\{1\}}, \mathbf{x}^{\{2\}}, \dots, \mathbf{x}^{\{k\}}\}$ and their labels $\mathbf{Y} = \{y^{\{1\}}, y^{\{2\}}, \dots, y^{\{k\}}\}$, consider a supervised learning problem in which a function $f : \mathbf{x} \rightarrow y$, or equivalently $P(y|\mathbf{x})$, is approximated (we drop the superscripts when no confusion is caused). More specifically, an input sequence $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the d -dimensional feature vector of the i^{th} frame; the output $y \in \mathcal{Y}$, a set of labels, and in our case, $\{\text{anger}, \text{disgust}, \text{fear}, \text{happiness}, \text{sadness}, \text{surprise}\}$.

We introduce another set of latent variables, $\mathbf{h} = \{h_1, h_2, \dots, h_T\}$ where $h_i \in \mathcal{H}$, to tackle the complex intrinsic relationship between the input \mathbf{x} and output y , as well as the dependencies among the individual frames in \mathbf{x} . The entire probabilistic system can be illustrated by a graphical model in fig. 5.5.

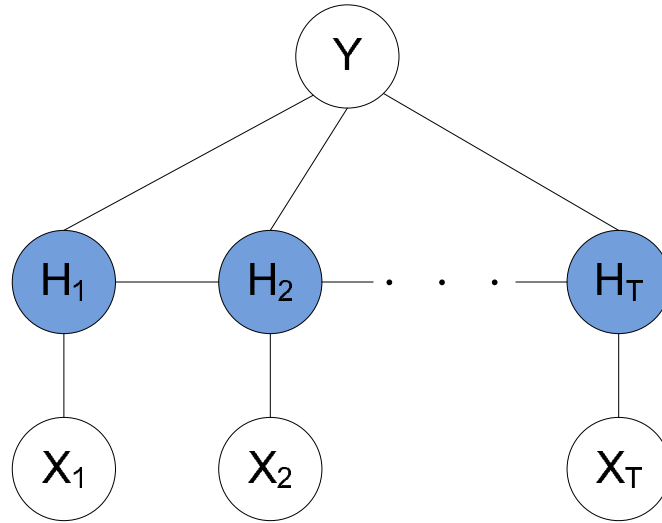


Figure 5.5: Graphical model for sequence classification

As we pointed out previously, there are typically three different ways to approach the sequence classification problem, or equivalently, the learning and inference in the above model, and each strategy can be conducted in the generative or discriminative fashion, which gives us six algorithms that cover both some classical and state-of-the-art

approaches in facial expression analysis. Generally speaking, the generative learning methods make model assumptions of the prior term $P(y)$ and the likelihood term $P(\mathbf{x}|y)$ (assuming we have knowledge about the procedure that *generates* samples \mathbf{x} from class y), and use the Bayes rule for classification. The discriminative methods, on the other hand, model $P(y|\mathbf{x})$ directly, i.e. *discriminating* class label y directly from any given sample \mathbf{x} .

The following are typical characteristics of generative models [125]:

- Learning generative model is to maximize the joint distribution of features and labels $p(\mathbf{x}, y)$, and use the Bayes rule for inference. Generative models implicitly model the distribution of observations, which may be an unnecessary modeling effort.
- Generative model is intuitive, and normally it's easy to train.
- Generative model implicitly imposes regularization to the learning task by assuming some form of model structure.
- The generative model can be viewed as a top-down approach.
- Generative models tends to have lower error rates when small number of training data are available.
- If the model assumption is correct, generative model is proved to have the lowest error bound asymptotically.
- Generative models usually assume strong independence structure among observations in order to ensure computational tractability.
- Generative probabilistic models can incorporate unlabeled data into parameter estimation by maximizing the marginal log-likelihood [92], thus generative models can be used in semi-supervised learning setting. However, if the generative

model is misspecified, unlabeled data may degrade performance on the task of interest [28].

- A new class can be added incrementally by learning its class-conditional density independently of all the previous classes.

The following are common characteristics of discriminative models:

- Learning discriminative model is to maximize the conditional distribution of labels given observation features $p(y|\mathbf{x})$.
- Discriminative model is more suitable for classification task, but hard to train and prone to overfitting, usually it needs regularization to avoid over-fitting.
- Discriminative is data-drive (bottom-up) approach, where the objective is to find the model which is consistent with all the known facts, but otherwise as uniform as possible. Intuitively, the principle is simple: model all that is known and assume nothing about that which is unknown. This is also the principle of maximum entropy learning methods.
- Generally, when large amount of training data is available, discriminative models have been shown to have lower asymptotic error and are thus preferred.
- Discriminative models relax the strong independence assumptions of generative models and allow richer arbitrary, non-independent feature functions.
- The flexibility of the model is used in regions of input space where the posterior probabilities differ significantly from 0 or 1, whereas generative approaches model details of the distribution which may be irrelevant for determining the posterior probabilities.

From the comparison of the characteristics of generative and discriminative models, it almost looks like the generative model is never useful for classification purposes.

However, for large-scale applications such as object recognition, when hand labeling of data is expensive, and there is much interest in semi-supervised techniques based on generative models in which the majority of the training data is unlabeled. It's hard to use unlabeled data in a discriminative setting. Recently, hybrid generative and discriminative models [136, 66] have been proposed which try to combine the best of both approaches. One simple heuristic for training a hybrid generative and discriminative model is by optimizing a convex combination of the generative and discriminative objective functions. In this thesis we do not further investigate the hybrid generative discriminative models or discriminatively trained generative models.

In this work, we propose a unified objective function for various models:

$$\sum_{\mathbf{x}, \mathbf{y} \in \mathbf{T}} \sum_{\mathbf{h}} F(\mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}, y)) + R(\mathbf{w}, \lambda)$$

where \mathbf{X}, \mathbf{Y} are the training set data labels respectively. \mathbf{T} is the training set. R is the regularization term (normally L_1 or L_2 norm, and we use L_2) with balance parameter λ . F is the data log-likelihood (in generative methods) or conditional log-likelihood (in discriminative methods). \mathbf{w} is the model parameter, and the feature function Φ , also called the joint feature map, ties the input, output and latent variables together. Different objectives can be achieved by choosing F accordingly. The following sections show the details of the different algorithms expressed by our model under different assumptions. Note that once the objective function is defined, different properties of the objective function will give different theoretical convergence guarantees. For example, if the objective function is convex, then properly chosen optimization procedure should return global optimum parameter estimation. In our example, the CRF objective function is log linear and convex, while the mixture of Gaussian observation model for HMMc and HMMt will cause them non-convex. The first order and second order derivatives of objective functions will determine the convergence rate for gradient based

optimization procedures, such as Newton based or Conjugate gradient algorithms.

5.3.2 Discriminative Sequence Classification - HCRF

To solve the sequence classification problem discriminatively, our model is equivalent to the Hidden Conditional Random Field (HCRF) model [102]. In this setting, the variables \mathbf{h} are *hidden* states (i.e., not given in training), which capture certain intermediate structures of the input. The conditional likelihood term is of the following form:

$$P(y, \mathbf{h} | \mathbf{x}; \mathbf{w}) = \frac{\exp(\mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}, y))}{\sum_y \sum_{\mathbf{h}} \exp(\mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}, y))}$$

where:

$$\begin{aligned} \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}, y) = & \sum_t \left\{ \sum_{i,j \in \mathcal{H}, k \in \mathcal{Y}} w_{ijk} 1(y = k) 1(h_t = i) 1(h_{t-1} = j) \right. \\ & \left. + \sum_{i \in \mathcal{H}, k \in \mathcal{Y}} w_{ik} 1(y = k) 1(h_t = i) + \sum_{i \in \mathcal{H}} \mathbf{w}_i 1(h_t = i) \mathbf{x}_t \right\} \end{aligned} \quad (5.1)$$

where 1 is the indicator function. With slight abuse of notation, w_{ijk} is the parameter corresponding to the feature $1(y = k)1(h_t = i)1(h_{t-1} = j)$, and it measures the compatibility between sequence label k and neighboring hidden states i and j . The same applies to the other parameters.

Due to the introduction of hidden states, the HCRF objective function is not convex any more. Expectation maximization (EM) algorithm is used to optimize the parameters. During the optimization process, belief propagation is used to compute marginal conditionals, which are needed for objective function and gradient evaluation. Scaled conjugate gradient is used for optimization.

5.3.3 Generative Sequence Classification - HMM-C

The generative counterpart of the HCRF is the Hidden Markov Model (HMM), one of the most popular tool for sequence classification. We call it HMM-C because HMMs can also be used for sequence tagging (named HMM-T). The potential function is similar to that of the HCRF model. The difference lies in the first and last terms in equation 5.1: for HMM-C, it lacks the first term and the its third is modeled with mixture of Gaussians, while for HCRF, it is modeled with the softmax function [114]. Baum-Welch algorithm (essentially EM) is used to train HMM-C. For testing, we evaluate the probability of a testing sequence against different models and classify it according to the likelihood of different models. Note that HMM could also be trained to maximize the conditional likelihood or maximize the margin between correct label and wrong labels. In those cases, the HMM is called discriminatively trained or maximum-margin trained generative model.

5.3.4 Discriminative Sequence Tagging - CRF

To simplify the HCRF modeling, rewrite the conditional likelihood term as follows:

$$P(y, \mathbf{h}|\mathbf{x}) = P(\mathbf{h}|\mathbf{x})P(y|\mathbf{h})$$

we can specifically use h_i to model the expression label of a single frame \mathbf{x}_i , i.e., let $\mathcal{H} = \mathcal{Y}$. \mathbf{h} is not *hidden* any more and can be given according to y in the training. The first term can now be modeled by either the CRF model (discriminatively) or HMM-T (generatively). The second term is essentially a majority voting scheme because y has to be *compatible* to h_1, h_2, \dots, h_T , because, again, h_i was given according to y during the training. This results in the second strategy for sequence classification, i.e., sequence tagging plus majority voting.

For CRFs, the conditional likelihood term is:

$$P(\mathbf{h}|\mathbf{x}; \mathbf{w}) = \frac{\exp(\mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}))}{\sum_{\mathbf{h}} \exp(\mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}))}$$

where:

$$\mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}) = \sum_t \left\{ \sum_{(i,j) \in \mathcal{H}} w_{ij} 1(h_t = i) 1(h_{t-1} = j) + \sum_{i \in \mathcal{H}} \mathbf{w}_i 1(h_t = i) \mathbf{x}_t \right\} \quad (5.2)$$

Note that by sequence tagging and majority voting, we essentially replaced the first term in equation 5.1 with the first term in equation 5.2, losing the effect of y on the compatibilities among \mathbf{h} , and the second term in equation 5.1 is approximated by the voting procedure.

Just as in HMM, the parameters of CRF are tied at different time steps. The number of parameters for a linear chain CRF is $c*d+c^2$, where c is the number of classes. Unlike the HMM for sequence tagging, the CRF doesn't have a close form solution. Normally gradient based algorithms are used to iteratively update CRF parameters. The gradient of the likelihood function w.r.t. model parameters is the difference between empirical feature and expected feature given model parameters. The CRF converges in around 20 iterations with scaled conjugate gradient training in our experiments.

5.3.5 Generative Sequence Tagging - HMM-T

HMM is normally used for sequence classification, but it can also be used for sequence tagging. The joint distribution of a sequence can be written as:

$$P(\mathbf{x}, \mathbf{h}) = P(h_0) \prod_t P(\mathbf{x}_t | h_t) P(h_t | h_{t-1})$$

where $P(\mathbf{x}_t | h_t)$ is the observation model, which in our case is a mixture of Gaussians. The joint distribution can be written the same way as before: $P(\mathbf{x}, \mathbf{h}) = \exp(\mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}))$ where

$$\mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}) = \sum_t \left\{ \sum_{(i,j) \in \mathcal{H}} P(h_t|h_{t-1}) 1(h_t = i) 1(h_{t-1} = j) + P(\mathbf{x}_t|h_t) \sum_{i \in \mathcal{H}} 1(h_t = i) \mathbf{x}_t \right\}$$

and parameter \mathbf{w} consists of the logarithm of transition $P(y_t|y_{t-1})$ and emission $P(\mathbf{x}_t|y_t)$ probabilities. Suppose we use m mixture components for each emission distribution, then the total number of parameters in the HMM tagging problem is $c * m * (d + d^2) + c * (m - 1)$, the first part is the mean and covariance for all the mixture components, and the second part is the mixing coefficients.

During training, the state labels for each time slice is given and the parameters can be estimated in close-form. During testing, due to the linear chain structure of HMM, one can use the efficient dynamic programming algorithm for the inference state labels for each time slice. The overall label for a sequence is given by majority voting.

Common critics for the HMM is that it has to build an specific form of observation model $P(\mathbf{x}|y)$, normally assumed to be mixture of Gaussian for continuous observation. If the data happen to follow the Gaussian distribution, then HMM would be the optimal classifier, however, in practice it is hard to see if high dimensional observations follow a specific distribution. The discriminative model has a more relaxed assumption and may have better performance where no strong assumption can be made on the data.

The Viterbi algorithm is used in both HMM and CRF tagging scenario. The algorithm first performs a forward pass over the input sequence, during the process saving both probability scores and the most probable previous state ending with current state. Then the algorithm uses the stored information to back track the optimal state sequence. This is an efficient dynamic programming algorithm. However, all the data must be available before the optimal state sequence could be recovered. In the conclusion and future work chapter of this thesis, we introduce other research work on applying the Viterbi algorithm to streaming data with high accuracy and low latency.

5.3.6 Discriminative Frame Classification - MNLR

The linear chain CRF is an extension of the Multinomial Logistic Regression (MNLR) in temporal domain: for each time slice, the association between label and observation are modeled the same as in logistic regression. If we make independent assumptions further over \mathbf{h} , the CRF and HMM learning essentially reduces to logistic regression and naive Bayes, respectively. To keep the notations simple, we now use \mathbf{x} to denote the feature vector from a frame and y the frame label. The conditional likelihood of label y given observation \mathbf{x} is:

$$P(y = j|\mathbf{x}) = \frac{\exp(\mathbf{w}^T \Phi(\mathbf{x}, y))}{\sum_y \exp(\mathbf{w}^T \Phi(\mathbf{x}, y))}$$

where

$$\Phi(\mathbf{x}, y) = \mathbf{x}1(y = j)$$

As we'll show shortly, when the class-conditional distribution of Naive Bayes is Gaussian, the class posterior will be in the form of logistic regression. Actually, Poisson, or more generally exponential family class-conditional all lead to the same form of posterior, which indicates logistic regression is more general with less modeling assumptions. The objective function is convex in Φ . The number of parameters is $(c-1)*(d+1)$. Usually gradient based algorithm is used to optimize the parameters, one popular algorithm is the iteratively reweighted least square or IRLS.

5.3.7 Generative Frame Classification - Naive Bayes

Naive Bayes classifier is the generative counterpart of the multinomial logistic regression. In our experiments, we assume class-conditional distribution to be Gaussian $N(\mu_{ij}, \sigma_{ij})$ for each component x_i of \mathbf{x} given class label $y = j$ and uniform prior for

each class.

$$P(\mathbf{x}|y) = \prod_{i=1}^d P(x_i|y) = \prod_{i=1}^d N(\mu_{ij}, \sigma_{ij})$$

The number of parameters is $2*c*d$, where c is the total number of class labels and d is the feature dimension. The maximum likelihood estimate of Naive Bayes parameters is close-form:

$$\mu_{ij} = \frac{\sum x_{ij}1(y=j)}{\sum 1(y=j)}, \sigma_{ij} = \frac{\sum (x_{ij} - \mu_{ij})^2 1(y=j)}{\sum 1(y=j)}$$

One can view the multi-class classification problem as multiple one vs. one binary classification problems. The class posterior $P(y|\mathbf{x})$, where y is binary label, can be written in the same form as logistic regression when assuming pooled variance σ_i^2 , and the decision surface will be hyperplane.

$$P(y|\mathbf{x}) = \frac{1}{1 + \exp(\mathbf{w}^T \Phi(\mathbf{x}, y))}, \Phi(\mathbf{x}, y) = [1 \ x^T]^T$$

where $w_0 = \sum_i \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\sigma_i^2}$, $w_i = \frac{\mu_{i0} - \mu_{i1}}{\sigma_i^2}$

Note that for generative models, no regularization terms will be used since the model assumption itself has specific form of distribution and no "overfitting" would occur.

5.4 Experimental Results

To evaluate the performance of the above algorithms, we used the publicly available CMU Cohn-Kanade facial expression database [55]. For a fair comparison of different algorithms, the same part-based KLT features we propose in section 5.2 are used, and the model parameters (most obviously, the number of possible hidden states in HMM-C and HCRF) were empirically tuned, and the best results for each algorithm are shown below.

Table 5.1 shows the confusion matrices for the three generative models: Naive Bayes, HMM-T and HMM-C. Table 5.2 shows the confusion matrices for the three discriminative models: multinomial logistic regression, CRF and HCRF. All the results are based

Naive Bayes							HMM Sequence Tagging						
	A G	D G	F A	H P	S D	S P		A G	D G	F A	H P	S D	S P
AG	26	2	0	0	2	0	AG	23	2	3	0	2	0
DG	4	33	1	0	5	0	DG	4	37	2	0	0	0
FA	4	1	35	5	7	0	FA	2	0	44	5	0	1
HP	0	0	1	86	9	0	HP	0	0	6	88	1	1
SD	3	0	1	0	65	0	SD	0	4	0	2	59	0
SP	0	0	0	1	8	66	SP	0	0	0	1	0	74

HMM Sequence Classification						
	A G	D G	F A	H P	S D	S P
AG	22	3	3	1	1	0
DG	1	39	2	0	0	1
FA	1	3	41	3	3	1
HP	0	1	1	91	2	1
SD	0	3	1	0	60	1
SP	0	0	1	0	1	73

Table 5.1: Naive Bayes, HMM Tagging, HMM Classification, the traces for the matrices are 311 (85.21%),325 (89.04%),326 (89.32%)

on 5-fold cross validation.

Figure 5.6 shows the convergence rates for some of the models that use the gradient descent optimization. Figure 5.7 shows the error rates for different models when PCA was applied to the feature vectors. The x-axis shows the percentage of features kept in PCA. Figure 5.8 shows the performance changes for the all the models with dynamic features when using different number of frames from each sequence, counting from the apex backward. Note that when using very few frames (1-4), the HCRF doesn't perform well, but when the number of frames used is larger than 7 frames, the advantage compared with other methods is obvious, which indicates our hypothesis that the correlations between frames in a sequence help improve recognition rate. Figure 5.9 shows the performance changes for different models with multiple instance selected features when using different number of frames from each sequence. All the recognition rates are based on 5-fold cross validation.

Even though our goal is not to find optimal features and representations for solving

Multinomial Logistic Regression							Conditional Random Field						
	A G	D G	F A	H P	S D	S P		A G	D G	F A	H P	S D	S P
AG	21	2	0	3	4	0	AG	25	2	0	1	2	0
DG	2	38	0	3	0	0	DG	4	36	0	2	0	1
FA	0	1	28	15	5	3	FA	0	0	44	6	2	0
HP	0	0	0	96	0	0	HP	0	0	3	93	0	0
SD	2	0	0	1	64	2	SD	2	0	1	1	64	1
SP	0	0	0	2	0	73	SP	0	0	0	0	1	74

Hidden Conditional Random Field						
	A G	D G	F A	H P	S D	S P
AG	27	2	0	0	1	0
DG	2	40	1	0	0	0
FA	0	0	49	2	1	0
HP	0	0	0	96	0	0
SD	0	0	1	1	67	0
SP	0	0	0	0	0	75

Table 5.2: MNLR, CRF and HCRF, the traces for the matrices are 320 (87.7%), 336 (92.1%), 354 (97.0%)

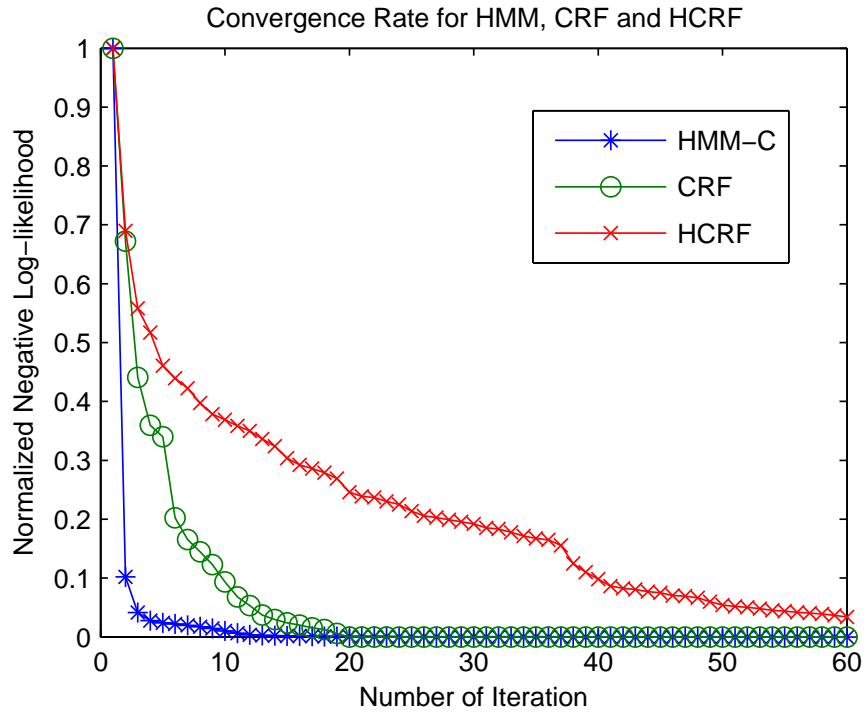


Figure 5.6: Convergence Rate for HMM, CRF and HCRF

	Frame	Seq. Tagg.	Seq. Classi.
Generative	85.21	89.04	89.32
Discriminative	87.67	92.05	96.99

Table 5.3: Overall recognition rate for different classifiers

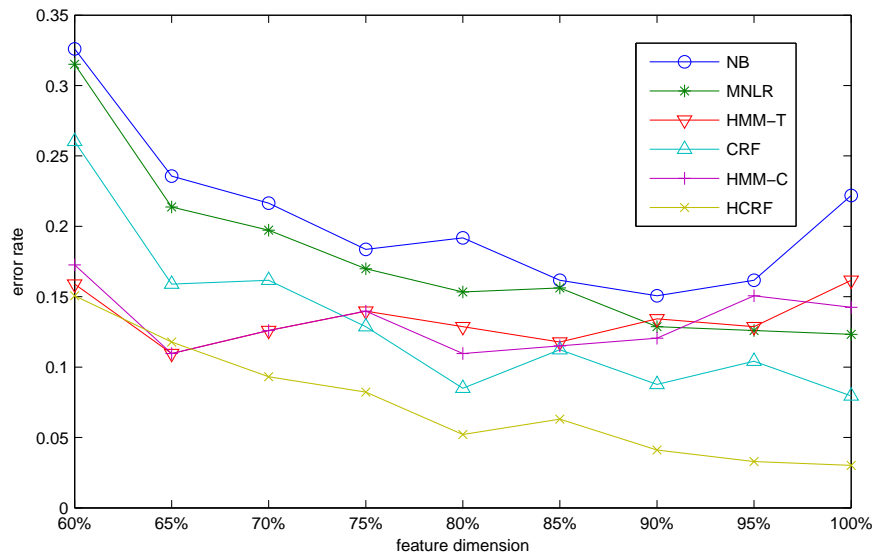


Figure 5.7: Error Rates vs. PCA Dimension Reduction

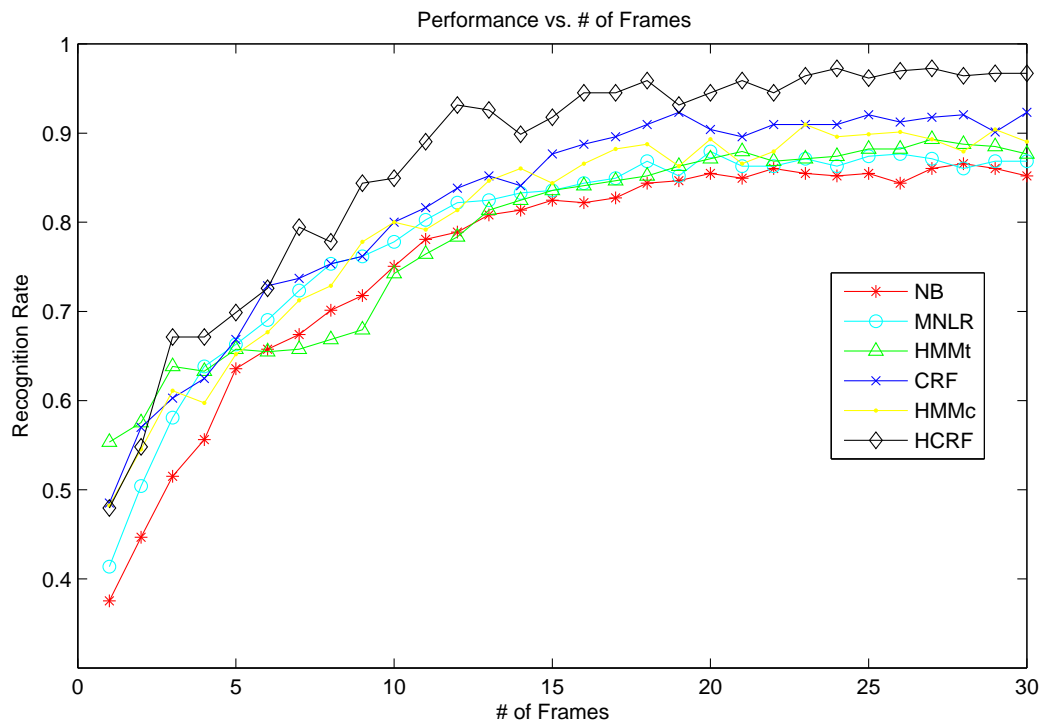


Figure 5.8: Error Rates with Dynamic Feature vs. Number of Frames Used from a Sequence

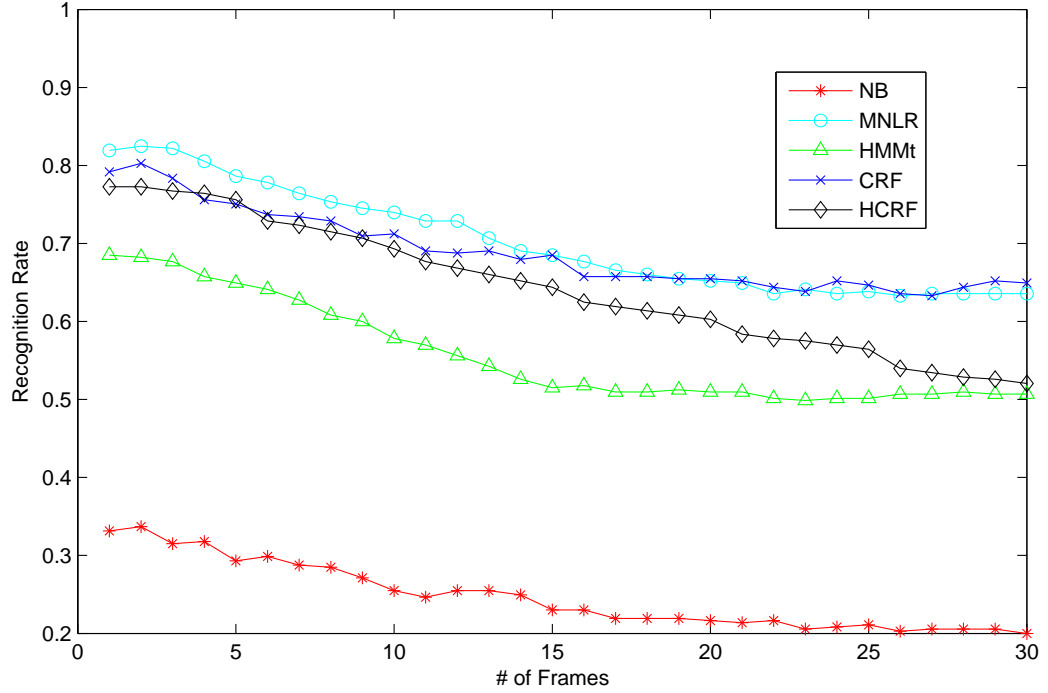


Figure 5.9: Error Rates with Multiple Instance Feature vs. Number of Frames Used from a Sequence

a specific object recognition task, the results we got are one of the best compared with other research.

Table 5.4 shows comparison of our results with other research. Some of the methods are not directly comparable since the number of sequences used are different, even if the number of sequences used are the same, different training and testing data arrangement for N-fold cross validation can also cause fluctuations in performance. But table 5.4

	People#	Sequence#	Class#	Measure	Recognition Rate (%)
[141]	97	374	6	2-fold	95.19
[141]	97	374	6	10-fold	96.26
[108]	96	320	6	10-fold	92.1
[10]	90	313	7	10-fold	86.9
[74]	90	313	7	leave-one-out	93.8
[120]	97	375	6	N/A	93.8
[139]	97	N/A	6	5-fold	90.9
[3]	90	284	6	N/A	93.66
[19]	97	392	6	5-fold	92.86
Ours	97	365	6	5-fold	96.99

Table 5.4: Comparison of our results with other published results

indeed gives the general indication of the performance of different approaches.

5.4.1 Observations and Analysis

We have the following observations from the experimental results:

1. Discriminative vs. Generative

We have expected that the discriminative methods would outperform the generative methods in such classification tasks, and the results prove so. The three discriminative methods unanimously perform better than their generative counterparts. As shown previously, the posterior of the generative models can be written in the same form as their discriminative counterparts, which indicates if the model assumption is valid, they should converge to the same solutions given enough training samples. The results, however, show that the generative models do not perform as well as the discriminative models. The possible reason is that the Gaussian assumption is not valid, or the amount of data is relatively small compared to the number of parameters in Gaussian mixture models. We have observed performance increase from the generative models when the data dimensionality is mildly reduced using PCA, as shown in Fig. 5.7, which can be attributed to the reduced number of model parameters and the denoised data. On the other hand, when feature dimension is high, discriminative models performs significantly better.

2. Static Frame Classification vs. Sequence Tagging

Theoretically, the biggest difference between HMM-T and Naive Bayes, or that between CRF and Logistic Regression, is the added constraints over time through the connections between the latent variables, even though the frame-based classification also uses temporal information via the dynamic features. Since the observed data contains a unique expression in each sequence, the learned transition matrices in HMM-T and CRF are both mostly diagonally dominant. The tagging procedure seems to propagate the

labels of the more distinguishable frames (e.g., from the apex of different expressions) and improve the classification of those more ambiguous frames (e.g., from the neutral stage), hence provides better voting results.

3. Sequence Tagging vs. Sequence Classification

The difference between these two categories of methods is the way in which the latent variables \mathbf{h} are used. In the tagging based methods (i.e., HMM-T and CRF), the latent variables are treated as frame labels that are highly correlated to the sequence label y , while in the sequence classification methods (i.e., HMM-C and HCRF), \mathbf{h} are used more flexibly (e.g., 15 possible hidden states are used in HCRF, as opposed to 6 in CRF), and recall that the sequence tagging based methods lose the feature functions that tie \mathbf{h} and y together. Note that when the number of hidden states for HMMc is set to 1, then HMMc and HMMt would be exactly the same model if the number of mixture components for Gaussian is the same. In some situations, only sequence classification algorithms are applicable, since it's impractical to assign tagging label to each individual time slice. On the other hand, sequence classification methods do not require the explicit assignment of tagging labels, they try to learn the hidden states for each individual time slice instead.

4. Sequential methods performance vs. static methods performance using different feature dimensions

We observe that when feature dimension used is low, the performance difference between frame and sequence wise classification are larger, which indicates when there is not much information in the observation, interdependencies between sequential labels contribute a lot for recognition. Especially note that HCRF performs significantly better than all other classifiers based on 2-dimensional feature. When the features reach around 90% of total energy, the performance becomes stable. While keeping 90% of the total energy, the feature dimension is reduced from 60 to 42, which saves a lot

of computational power and memory requirement.

5. Computational Complexity

Dynamic programming algorithms for HMM and belief propagation for CRF and HCRF are both highly efficient, resulting fast training and testing. In our experiments, the training time of the generative models are mostly in the range of seconds, mostly counting the frequencies and estimating Gaussian parameters. The discriminative methods run from minutes to dozens of minutes, with HCRF taking the longest time. Testing for all models is real time or near real time. This is actually a comparison of EM algorithm to gradient based methods. In NB, HMMt and HMMc, mixture of Gaussian parameters are estimated with EM algorithm, while in MNLr, CRF and HCRF, parameters are estimated using gradient descent method. EM algorithm has several advantages: no step size parameter and simple to implement. However, when there is no close form MLE, the M-step has to be implemented with gradient based methods.

6. Sensitivity to the number of KLT feature patches

Performances are generally not sensitive to grid size (when extracting facial features). We used 30 (5x6) patches in our final experiments, but we observe relatively stable results from 20 to 48 patches. Interestingly, performance decreases when optical flow feature from multiple frames are combined as feature vector.

7. Static Apex Recognition vs. Feature Type

When using only the apex frame for expression recognition, the multiple instance based PCA feature got better results than the KLT histogram feature. A possible explanation is that near apex expression, the relative facial muscle movement is small, in which case the KLT feature captures few information about expression. On the other hand, considering the difference between different facial expressions, obviously at the apex they are farther way from each other.

8. Performance trend vs. number of frames used

The HCRF model performs relatively bad using fewer frames, but its advantage is obvious when around 7 or more frames are used, which indicates the correlations between frames in the sequence help improving recognition rate compared with independent recognition.

From figure 5.8 and figure 5.9, we notice another interesting phenomenon: when using the KLT histogram feature, the more number of frames used per sequence, the higher the recognition rate; on the other hand, when using the multiple instance selected PCA feature, the more number of frames used, the worse the performance. And when using very small number of frames, the performances for PCA feature is better than that of the KLT histogram feature.

Because when it's near to the apex expression, the facial movement is relatively small and thus the KLT feature will be relatively indiscriminative for all the expressions. On the other hand, the intensity of different facial expressions are very different for facial expression images that are near to the apex.

When the number of frames used increases, the KLT histogram based feature quickly outperform the PCA feature, which indicates that when more frames are used, KLT histogram becomes more discriminative for different expressions, while the image frames at the beginning of each sequence (near neutral expressions) cause more confusion for the PCA feature.

9. Parameter tuning

The flexibility of sequence classification algorithms comes with a cost: the number of hidden states in HCRF and HMMc is a parameter that needs to be fine tuned to get best results. For models that are learned gradient based optimization techniques, such as the Multinomial logistic regressor, CRF and HCRF, setting up the appropriate stopping condition is important for recognition performance.

10. Combining dynamic feature with sequential modeling

Last but not least, the combination of dynamic feature and sequential modeling could lead to best performances. The using of dynamic features alone, such as all variants of spatial-temporal features, by previous work is not enough to get best results. On the other hand, a dynamic model with poorly chosen feature representation is also not ideal. As shown in our experiments, the combination of multiple instance learned feature with sequential modeling techniques doesn't prove to be effective. A good paradigm need to consider both dynamic feature and dynamic modeling, and couple the two factors to benefit each other.

5.5 Conclusions and Discussions

In this chapter we first proposed a part-based dynamic feature that captures facial movement information both spatially and temporally. We then proposed a probabilistic learning framework that unifies several classical and state-of-the-art algorithms, both generative and discriminative. We investigated the similarities and dissimilarities of these algorithms in various aspects. We also empirically studied the facial expression recognition performances of these algorithms on a publicly available data set, and the observations from those experiments are one of the main contributions for this work. From our experimental results, sequential classification methods normally outperform sequential tagging methods and majority voting methods due to the flexibility of model structure, with the cost of longer learning time and the risk of overfitting. We also observed that either dynamic features alone or dynamic models alone are not enough for good performance. However, the coupling of simple dynamic feature and temporal modeling improves performance a lot. We reported some of the best experimental results in the literature up to date by coupling the dynamic feature representation with the discriminative sequential modeling methods. Future work could extend the

proposed method to the case of human action recognition.

Chapter 6

Conclusions and Future Research Direction

Despite the progress researchers have made in building automated computer systems for various vision tasks, such as image segmentation and object recognition etc, it is embarrassingly ineffective compared with how human beings could easily handle these tasks. Is it the power of the human sensors (eyes) or is it the human computational power (brain) that makes human perform those tasks so easily? To study the physiology and psychology of human beings may shed some light on future research directions, however, we could not count on our ability to discover the secret since we are not clear how the first cell come into life. Even though a lot of computational models and methods have been proposed in vision research, when it comes to real world application, it seems nothing is adequate, the feature representation, the appropriate models etc.

This thesis is one tiny particle of ongoing research efforts of the large computer vision community. Particularly, in this thesis we focus various aspects of a typical video-based facial expression system. Granted, to build an automated system which are robust and reliable for video based facial expression recognition needs a lot of efforts, as proven by thousands of research articles published. We focus and make contributions on face tracking, facial feature representation and feature selection under different face alignment conditions, and finally sequential modeling for dynamic expression recognition.

The first part of this thesis is about improving face tracking by bringing together the best of both worlds: the 3D deformable model and the 2D active shape model. We develop a framework for robust 3D face tracking that combines the strengths of both

3D deformable models and ASMs. In particular we use ASMs to track reliably 2D image features in a video sequence. In order to track large rotations we train multiple ASMs that correspond to frontal and several side head poses. The 2D features from the ASMs are then used for parameter estimation of the 3D deformable model. After the 3D parameter-space outlier rejection and occlusion handling, the updated 3D model points are projected to 2D image to initialize the 2D ASM search, which becomes more robust due to the better model initialization. In addition we couple stochastically 3D deformable models and ASM by using the 3D pose of the deformable model to switch among the different ASMs, as well as to deal with occlusion. This integration allows the robust tracking of faces and the estimation of both their rigid and nonrigid motions. We demonstrate the strength of the framework in experiments that include automated 3D model fitting and facial expression tracking for a variety of applications including American Sign Language.

Even under very good conditions, current face tracking technique still cannot always accurately track facial movement. In order to perform accurate facial expression analysis, we need to make the feature extraction part robust to imprecisely located face images. The second part of the thesis is about how to extract effective feature representations for poorly aligned face images. We first systematically investigate the effect of mis-aligned face images on face recognition systems. To make classifiers robust to the unavoidable face registration error, we formulate the facial feature representation and selection with poorly aligned face images as a multiple-instance learning task. We propose a novel multiple-instance based subspace learning scheme for facial feature representation and feature dimension reduction. In this algorithm, noisy training image bags are modeled as the mixture of Gaussians, and we introduce a method to iteratively select better subspace learning samples. Compared with previous methods, our algorithm does not require accurately aligned *training and testing* images, and can achieve

the same or better performance as manually aligned faces for face recognition and facial expression recognition tasks. In this thesis, we used the term "*noisy images*" to denote poorly aligned images. We also empirically studied various other feature representations and feature selection methods for face and expression recognition tasks. We performed large scale facial action units recognition on large facial expression dataset with spontaneous facial expression and large head pose change.

An efficient feature representation and classifier is the most important part of the expression recognition system. In the last part of the thesis, we first proposed a part-based dynamic feature that captures facial movement information both spatially and temporally. We then proposed a probabilistic learning framework that unifies several classical and state-of-the-art algorithms, both generative and discriminative. We investigated the similarities and dissimilarities of these algorithms in various aspects. We also empirically studied the facial expression recognition performances of these algorithms on a publicly available data set, and the observations from those experiments are one of the main contributions for this work. From our experimental results, sequential classification methods normally outperform sequential tagging methods and majority voting methods due to the flexibility of model structure, with the cost of longer learning time and the risk of overfitting. We also observed that either dynamic features alone or dynamic models alone are not enough for good performance. However, the coupling of simple dynamic feature and temporal modeling improves performance a lot. We report some of the best experimental results in the literature up to date by coupling the dynamic feature representation with the discriminative sequential modeling methods. Future work could extend the proposed method to the case of human action recognition.

Despite the many advances researchers have made, current video based facial expression analysis system is still far from perfect. The biggest challenges include robust

face tracking under real world conditions, including lighting, background clutter, occlusion, out-of-plane rotation, fast facial movement etc. Video segmentation for facial expression recognition is an active research topic. It could be accomplished as part of the dynamic modeling process, such as the hierarchical HMMs [43]. Hierarchical hidden Markov models [43] extends the popular hidden Markov model to hierarchical structure to model complex multi-scale structure existing in many natural sequences, particularly in language and speech. In the hierarchical HMM, emissions at higher levels are sequences computed by sub-models at a lower level in the hierarchy. Fine et al [43] derived efficient algorithm for estimating model parameters from unlabeled data and used the trained model for automatic hierarchical parsing of observation sequences, specifically, natural English text and cursive handwriting.

Future research direction on video-based facial analysis will focus on the following topics:

- On-line Learning and Decoding

For on-line expression recognition, video sequences should be treated as streaming data. Streaming data is ubiquitous and there is a real need to store, query and analyze such rapid large volumes of data. Examples of other data streams include: data generated from wireless sensor networks, web logs and click streams, ATM transactions, search engine logs, phone call records and surveillance cameras. There are wide range of potential applications for streaming data analysis. For example, on-line shopping experiences could be improved by analyzing customer web logs, and detecting changes in surveillance camera data streams can be used for security purposes. One distinguishing characteristic setting streaming data apart from pre-stored data is that streaming data usually exhibits time-changing data characteristics, often called concept drift in data mining community. It

often requires on-line incremental learning to deal with concept drift. In our facial expression analysis scenario, we consider the learning environment static with no concept drift since facial expressions have relatively fixed pattern over time.

We currently perform expression recognition off-line. The original Viterbi decoding algorithm works offline, i.e., the entire input sequence must be observed before the optimal state path could be generated. It cannot be directly applied to on-line or streaming scenario without incurring significant delays. Widely used method to apply Viterbi on streaming data is to divide the input stream into fixed length windows and apply the Viterbi algorithm within each window. On-line learning could use the fixed-lag smoothing inference procedure instead of the fixed-interval smoothing inference [87]. Larger window is supposed to have better accuracy, but with longer delays. Narasimhan et al [88] developed an approach where they don't need to select the windows size up front, instead, their algorithm dynamically select window size to balance the latency and accuracy. Compared with fixed window Viterbi, their approach achieved both higher accuracy and smaller latency.

- Structured Output Learning

From machine learning perspective, the graphical model approaches studied in this thesis for dynamic modeling belongs to the so-called structured output learning problems. Other typical structured output learning algorithms include SVM-struct [5, 122, 123], Maximum margin Markov network (M3N) [117] etc. Typical applications of structured output learning methods include text mining and bio-informatics. Nam et.al. [90] compared the performances of different sequential tagging algorithms on POS tasks and handwritten character recognition (OCR).

With due tuning efforts of the parameters of each model, based on their experiments on POS tasks and OCR, they drew the conclusion that SVM-struct performs better than other approaches. However, Hoefel et al. [50] and [59] made the comments that Nguyen’s conclusion is not precise since their model is based on ineffective features for CRFs. With properly designed feature functions, the performance of the CRF is on par with the SVM-struct. It would be interesting to experiment with general structured output learning algorithms for expression recognition.

- Extending multiple instance based feature representation to spatial-temporal domain

As shown in our experiments, even though the multiple instance based feature representation and selection scheme works great for static face recognition and apex facial expression recognition, when combined with dynamic models, it does not work as well as the KLT histogram feature representation. One obvious reason is that it is static in nature. To extend the feature selection scheme to spatial-temporal domain would make it efficient for video-based expression recognition. Another important point is the algorithm will not work when the assumptions are not valid, i.e., for face recognition, if the face images which are well aligned and belong to the same people does not lie near to each other in the subspace, then we need to study to use other distance metric for subspace selection, such as non-linear manifold distance between instances.

- Various other topics

Our current face tracking module couples 2D ASM with 3D deformable model. As shown in [81], 3D models are a more natural and effective representation compared with 2D models. We hypothesize that combining multiple image cues

with learned 3D morphable model [15] could get better tracking results.

Lie detection application needs the study of the difference between spontaneous facial expressions and posed expressions, from the perspective of expression intensity and duration [86, 22]. The connections of spontaneous expressions with emotions for different cultures are also interesting [80].

As mentioned before, video segmentation should be an integral part of the video-based expression recognition system. Researches on segmentation of hand gestures [68, 4], called gesture spotting, could be borrowed in expression video segmentation. In [68], a threshold model is trained by using the union of all states for each specific gesture model and the threshold model, as its name suggests, acts as a threshold for classifying video segment into either a meaningful gesture or transitions between gestures by comparing the likelihood of the best gesture model with that of the threshold model. A gesture is recognized only when the likelihood of the best gesture model is higher than that of the threshold model.

References

- [1] G. Aggarwal, A. K. R. Chowdhury, and R. Chellappa. A system identification approach for video-based face recognition. *Pattern Recognition, International Conference on*, 4:175–178, 2004.
- [2] G. Aggarwal and A. Veeraraghavan. 3d facial pose tracking in uncalibrated videos. In *In International Conference on Pattern Recognition and Machine Intelligence (PReMI)*, 2005.
- [3] P. S. Aleksic and A. K. Katsaggelos. Automatic facial expression recognition using facial animation parameters and multi-stream hmms. In *Information Forensics and Security, IEEE Transactions on Volume 1, Issue 1, March 2006 Page(s):3*, page 11, 2005.
- [4] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff. A unified framework for gesture recognition and spatiotemporal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(1), 2008.
- [5] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden markov support vector machines. *Proceedings of International Conference on Machine Learning*, 2003.
- [6] G. Amayeh, A. Tavakkoli, and G. Bebis. Accurate and efficient computation of gabor features in real-time applications. In *ISVC '09: Proceedings of the 5th International Symposium on Advances in Visual Computing*, pages 243–252, Berlin, Heidelberg, 2009. Springer-Verlag.
- [7] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Proceedings of Neural Information Processing Systems*, pages 561–568, 2002.
- [8] L. Ballan, M. Bertini, A. Bimbo, and G. Serra. Video event classification using bag of words and string kernels. In *ICIAP '09: Proceedings of the 15th International Conference on Image Analysis and Processing*, pages 170–178, 2009.
- [9] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.
- [10] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan. Real time face detection and facial expression recognition: Development and applications to human computer interaction. In *In CVPR Workshop on CVPR for HCI*, 2003.
- [11] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing facial expression: Machine learning and application to spontaneous behavior. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, volume 2, pages 568–573, 2005.
- [12] P. N. Belhumeur, J. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.
- [13] M. J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 25(1):23–48, 1997.

- [14] M. J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 25(1):23–48, 1997.
- [15] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [16] C. Blum and M. J. Blesa. New metaheuristic approaches for the edge-weighted k-cardinality tree problem. *Computers and Operations Research*, 32(6):1355–1377, 2005.
- [17] R. N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill, New York, 1965.
- [18] L. Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, 1996.
- [19] K. Chang, T. Liu, and S. Lai. Learning partially-observed hidden conditional random fields for facial expression recognition. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 533–540, 2009.
- [20] Y. Chang, C. Hu, and M. Turk. Probabilistic expression analysis on manifolds. *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2:520–527, 2004.
- [21] I. Cohen, N. Sebe, A. Garg, L. S. Chen, and T. S. Huang. Facial expression recognition from video sequences: temporal and static modeling. *Computer Vision and Image Understanding*, 91(1-2):160–187, 2003.
- [22] J. Cohn and K. Schmidt. The timing of facial motion in posed and spontaneous smiles. *International Journal of Wavelets, Multiresolution and Information Processing*, 2:121–132, 2004.
- [23] J. Cohn, A. Zlochower, J.-J. J. Lien, and T. Kanade. Feature-point tracking by optical flow discriminates subtle differences in facial expression. In *Proceedings of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition (FG '98)*, pages 396 – 401, 1998.
- [24] M. Cooper, T. Liu, and E. Rieffel. Temporal video segmentation: A survey. *IEEE transactions on multimedia*, 9(3):610–618, 2007.
- [25] T. Cootes and C. Taylor. Active shape models — their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.
- [26] T. F. Cootes, G. Edwards, and C. Taylor. Comparing active shape models with active appearance models. In *in Proc. British Machine Vision Conf*, pages 173–182. BMVA Press, 1999.
- [27] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:681–685, 2001.
- [28] F. G. Cozman and I. Cohen. Unlabeled data can degrade classification performance of generative classifiers. In *in Fifteenth International Florida Artificial Intelligence Society Conference*, pages 327–331, 2002.
- [29] C. Darwin. *The expression of the emotions in man and animals*. John Murray, London, 1872.
- [30] D. de Carlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, 38(2):99–127, July 2000.
- [31] D. DeCarlo and D. Metaxas. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *CVPR '96*:

- Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, page 231, Washington, DC, USA, 1996. IEEE Computer Society.
- [32] D. Decarlo and D. Metaxas. Deformable model-based shape and motion analysis from images using motion residual error. In *in International Conference on Computer Vision*, pages 113–119, 1998.
 - [33] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski. Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):974–989, 1999.
 - [34] P. Ekman. Facial signs: Facts, fantasies, and possibilities. In T. Sebeok, editor, *Sight, Sound and Sense*. Bloomington: Indiana University Press, 1978.
 - [35] P. Ekman and W. Friesen. Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, 17(2):124–129, 1971.
 - [36] P. Ekman and W. Friesen. Facial action coding system: A technique for the measurement of facial movement. *Journal of Personality and Social Psychology*, 1978.
 - [37] P. Ekman, T. Huang, T. Sejnowski, and J. Hager. Final report to nsf of the planning workshop on facial expression understanding. Technical report, Human Interaction Lab., Univ. of California, 1993.
 - [38] I. Essa. *Analysis, Interpretation, and Synthesis of Facial Expressions*. PhD thesis, MIT, 1995.
 - [39] I. A. Essa and A. P. Pentland. Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):757–763, 1997.
 - [40] K. Etemad and R. Chellappa. Discriminant analysis for recognition of human face images. *Lecture Notes in Computer Science*, 1206:127–142, 1997.
 - [41] B. Fasel and J. Luetttin. Automatic Facial Expression Analysis: A Survey. *Pattern Recognition*, 36(1):259–275, 2003.
 - [42] P. Felzenszwalb, D. Mcallester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2008.
 - [43] S. Fine and Y. Singer. The hierarchical hidden markov model: Analysis and applications. In *MACHINE LEARNING*, pages 41–62, 1998.
 - [44] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47, 2000.
 - [45] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT '95: Proceedings of the Second European Conference on Computational Learning Theory*, pages 23–37, 1995.
 - [46] S. Goldenstein and C. Vogler. When occlusions are outliers. In *Workshop on the 25 Years of RANSAC (in conjunction with CVPR)*, 2006.
 - [47] S. Goldenstein, C. Vogler, and D. Metaxas. Statistical Cue Integration in DAG Deformable Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):801–813, 2003.
 - [48] S. Goldenstein, C. Vogler, and D. Metaxas. 3D facial tracking from corrupted movie sequences. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2004.

- [49] 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*, 2005.
- [50] G. Hoefel and C. Elkan. Learning a two-stage svm/crf sequence classifier. *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 271–278, 2008.
- [51] C. Hu, Y. Chang, R. Feris, and M. Turk. Manifold based analysis of facial expression. *CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop Volume 5*, 05:81, 2004.
- [52] M. Isard and A. Blake. Condensation: conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.
- [53] F. V. Jensen. *Introduction to Bayesian Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996.
- [54] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical methods. In *Machine Learning*, pages 183–233. MIT Press, 1998.
- [55] T. Kanade, Y. Tian, and J. F. Cohn. Comprehensive database for facial expression analysis. In *Proceedings of International Conference on Automatic Face and Gesture Recognition*, page 46, 2000.
- [56] A. Kapoor, Y. Qi, and R. W. Picard. Fully automatic upper facial action recognition. *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, 00:195, 2003.
- [57] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 1(4):321–331, 1988.
- [58] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *Proceedings of International Conference of Computer Vision*, pages 166–173, 2005.
- [59] S. S. Keerthi and S. Sundararajan. Crf versus svm-struct for sequence labeling. *Technical report, Yahoo Research*, 2007.
- [60] M. Kim and V. Pavlovic. Discriminative learning for dynamic state prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1847–1861, 2009.
- [61] I. Koprinska, S. Carrato, and D. S. Carrato. Temporal video segmentation: A survey. *Signal Processing: Image Communication*, 16(5):477–500, 2001.
- [62] F. Kschischang, S. Member, B. J. Frey, and H. andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519, 2001.
- [63] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2003.
- [64] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.
- [65] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

- [66] J. A. Lasserre, C. M. Bishop, and T. P. Minka. Principled hybrids of generative and discriminative models. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 87–94, Washington, DC, USA, 2006. IEEE Computer Society.
- [67] C. Lee and A. M. Elgammal. Facial expression analysis using nonlinear decomposable generative models. *Analysis and Modelling of Faces and Gestures, Second International Workshop, AMFG 2005, Beijing, China, October 16, 2005, Proceedings*, 3723:17–31, 2005.
- [68] H.-K. Lee and J. H. Kim. An hmm-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:961–973, 1999.
- [69] Y. Lee, D. Terzopoulos, and K. Waters. Realistic modeling for facial animation. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 55–62, New York, NY, USA, 1995. ACM.
- [70] Z. Li, Q. Liu, and D. Metaxas. Face mis-alignment analysis by multiple-instance subspace. In *ACCV'07: Proceedings of the 8th Asian conference on Computer vision*, pages 901–910, Berlin, Heidelberg, 2007. Springer-Verlag.
- [71] Y. li Tian, T. Kanade, and J. F. Cohn. Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):97–115, 2001.
- [72] J.-J. J. Lien. *Automatic Recognition of Facial Expressions Using Hidden Markov Models and Estimation of Expression Intensity*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, April 1998.
- [73] X. Ling, O. Yuanxin, L. Huan, and X. Zhang. A method for fast shot boundary detection based on svm. In *CISP '08: Proceedings of the 2008 Congress on Image and Signal Processing, Vol. 2*, pages 445–449, Washington, DC, USA, 2008. IEEE Computer Society.
- [74] G. Littlewort, M. S. Bartlett, I. Fasel, J. Susskind, and J. Movellan. Dynamics of facial expression extracted automatically from video. In *Journal Image and Vision Computing*, pages 615–625, 2004.
- [75] D. Madigan, A. Genkin, D. D. Lewis, and D. Fradkin. Bayesian multinomial logistic regression for author identification. *25th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, pages 1226–1238, 2005.
- [76] S. Mallat. *A wavelet tour of signal processing*. Academic Press, 1999.
- [77] O. Maron and T. Lozano-Perez. A framework for multiple-instance learning. In *Proceedings of Neural Information Processing Systems*, pages 570–576, 1998.
- [78] A. M. Martinez. Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):748–763, 2002.
- [79] K. Mase and A. Pentland. Lipreading by optical flow. *Systems and Computers*, 22(6):67–76, 1991.
- [80] D. Matsumoto, A. Olide, J. Schug, B. Willingham, and M. Callan. Cross-cultural judgments of spontaneous facial expressions of emotion. *Journal of Nonverbal Behavior*, 33(4):121–132, 2009.
- [81] I. Matthews, J. Xiao, and S. Baker. 2d vs. 3d deformable face models: Representational power, construction, and real-time fitting. *Int. J. Comput. Vision*, 75(1):93–113, 2007.

- [82] T. Mcinerney and D. Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1:91–108, 1996.
- [83] D. N. Metaxas. *Physics-Based Deformable Models: Applications to Computer Vision, Graphics, and Medical Imaging*. Kluwer Academic Publishers, Norwell, MA, USA, 1996.
- [84] T. P. Minka. From hidden markov models to linear dynamical systems. Technical report, Tech. Rep. 531, Vision and Modeling Group of Media Lab, MIT, 1999.
- [85] L.-P. Morency, A. Quattoni, and T. Darrell. Latent-dynamic discriminative models for continuous gesture recognition. In *CVPR '07: Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.
- [86] M. T. Motley and C. T. Camden. Facial expression of emotion: A comparison of posed versus spontaneous expressions in an interpersonal communication setting. *Western Journal of Speech Communication*, 52:1–22, 1988.
- [87] K. P. Murphy. *Dynamic bayesian networks: representation, inference and learning*. PhD thesis, University of California, Berkeley, 2002.
- [88] M. Narasimhan, P. Viola, and M. Shilman. Online decoding of markov models under latency constraints. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 657–664, New York, NY, USA, 2006. ACM.
- [89] R. M. Neal. Probabilistic inference using markov chain monte carlo methods, 1993.
- [90] N. Nguyen and Y. Guo. Comparisons of sequence labeling algorithms and extensions. *Proceedings of International Conference on Machine Learning*, pages 681–688, 2007.
- [91] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, 2008.
- [92] K. Nigam, A. McCallum, and T. Mitchell. Semi-supervised text classification using em, 2006.
- [93] A. O'Toole, D. Roark, and H. Abdi. Recognizing moving faces: A psychological and neural synthesis. *Trends in Cognitive Science*, pages 261–266, 2002.
- [94] M. Pantic and L. J. M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1424–1445, 2000.
- [95] V. Pavlovic, J. M. Rehg, T.-J. Cham, and K. P. Murphy. A dynamic bayesian network approach to figure tracking using learned dynamic models. *Computer Vision, IEEE International Conference on*, 1:94, 1999.
- [96] V. Pavlovic, J. M. Rehg, and J. Maccormick. Learning switching linear models of human motion. *Proc. of Neural Information Processing Systems*, pages 981–987, 2000.
- [97] J. Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. *the American Association for Artificial Intelligence*, pages 133–136, 1982.
- [98] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.
- [99] M.-H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

- [100] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.
- [101] F. Pighin, R. Szeliski, and D. Salesin. Modeling and animating realistic faces from images. *International Journal of Computer Vision*, 50(2):143–169, 2002.
- [102] A. Quattoni, M. Collins, and T. Darrel. Conditional random fields for object recognition. In *NIPS '04: In Advances in Neural Information Processing Systems*, volume 17, 2004.
- [103] L. R. RABINER. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [104] H. Sakoe and S. Chiba. *Dynamic programming algorithm optimization for spoken word recognition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.
- [105] R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.
- [106] R. E. Schapire. A brief introduction to boosting. In *IJCAI'99: Proceedings of the 16th international joint conference on Artificial intelligence*, pages 1401–1406, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.
- [107] R. D. Shachter and S. K. Andersen. Global conditioning for probabilistic inference in belief networks. In *In Proc. Tenth Conference on Uncertainty in AI*, pages 514–522. Morgan Kaufmann, 1994.
- [108] C. Shan, S. Gong, and P. McOwan. Robust facial expression recognition using local binary patterns. *IEEE International Conference on Image Processing (ICIP)*, pages 370–373, 2005.
- [109] S. Shan, Y. Chang, W. Gao, and B. Cao. Curse of mis-alignment in face recognition: Problem and a novel mis-alignment learning solution. In *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pages 314–320, 2004.
- [110] J. Shi and C. Tomasi. Good features to track. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 593 – 600, 1994.
- [111] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional random fields for contextual human motion recognition. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1808–1815, Washington, DC, USA, 2005. IEEE Computer Society.
- [112] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 390–397, Washington, DC, USA, 2005. IEEE Computer Society.
- [113] J. Sung, T. Kanade, and D. Kim. Pose robust face tracking by combining active appearance models and cylinder head models. *Int. J. Comput. Vision*, 80(2):260–274, 2008.
- [114] C. Sutton and A. McCallum. Introduction to conditional random fields for relational learning, 2006.
- [115] M. Suwa, N. Sugie, and K. Fujimora. A preliminary note on pattern recognition of human emotional expression. *Proceedings of the 4th International Joint Conference on Pattern Recognition*, pages 408–410, 1978.

- [116] H. Tao and T. Huang. Visual Estimation and Compression of Facial Motion Parameters: Elements of a 3D Model-Based Video Coding System. *International Journal of Computer Vision*, 50(2):111–125, 2002.
- [117] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. *Proceedings of Neural Information Processing Systems*, 2003.
- [118] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *ACM SIGGRAPH Computer Graphics*, 21(4):205–214, 1987.
- [119] D. Terzopoulos and K. Waters. Physically-based facial modeling, analysis and animation. *Journal of Visualization and Computer Animation*, 1(4):73–80, 1990.
- [120] Y.-l. Tian. Evaluation of face resolution for expression analysis. In *CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 5*, page 82, Washington, DC, USA, 2004. IEEE Computer Society.
- [121] Y. Tong, W. Liao, and Q. Ji. Inferring facial action units with causal relations. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 1623–1630, 2006.
- [122] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. *Proceedings of International Conference on Machine Learning*, pages 823–830, 2004.
- [123] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, pages 1453–1484, 2005.
- [124] A. Tsymbal. The problem of concept drift: Definitions and related work. Technical report, Department of Computer Science, Trinity College Dublin, 2004.
- [125] I. Ulusoy and C. M. Bishop. Generative versus discriminative methods for object recognition. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 258–265, Washington, DC, USA, 2005. IEEE Computer Society.
- [126] M. Valstar, M. Pantic, and I. Patras. Motion history for facial action detection in video. In *IEEE Int'l Conf. on Systems, Man and Cybernetics 2004*, pages 635–640, October 2004.
- [127] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.
- [128] P. Viola, J. C. Platt, and C. Zhang. Multiple instance boosting for object detection. In *Proceedings of Neural Information Processing Systems*, 2005.
- [129] C. Vogler, S. Goldenstein, J. Stolfi, V. Pavlovic, and D. Metaxas. Outlier rejection in high-dimensional deformable models. *Image and Vision Computing*, 25(3):274–284, 2007.
- [130] C. Vogler, Z. Li, A. Kanaujia, S. K. Goldenstein, and D. Metaxas. The best of both worlds: Combining 3d deformable models with active shape models. *Proceedings of International Conference of Computer Vision*, 2007.
- [131] J. Wang and J.-D. Zucker. Solving multiple-instance problem: A lazy learning approach. In *Proceedings of International Conference on Machine Learning*, pages 1119–1125, 2000.
- [132] S. B. Wang, A. Quattoni, L.-P. Morency, and D. Demirdjian. Hidden conditional random fields for gesture recognition. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 1521–1527, 2006.

- [133] Y. Wang, X. Huang, C. su Lee, S. Zhang, Z. Li, D. Samaras, D. Metaxas, A. Elgammal, and P. Huang. High resolution acquisition, learning and transfer of dynamic 3-d facial expressions. In *Computer Graphics Forum*, pages 677–686, 2004.
- [134] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+3d active appearance models. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 535–542, 2004.
- [135] Y. Yacoob and L. S. Davis. Recognizing human facial expressions from long image sequences using optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):636–642, 1996.
- [136] O. Yakhnenko, A. Silvescu, and V. Honavar. Discriminatively trained markov model for sequence classification. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 498–505, Washington, DC, USA, 2005. IEEE Computer Society.
- [137] P. Yang, Q. Liu, X. Cui, and D. N. Metaxas. Facial expression recognition using encoded dynamic features. *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 1–8, 2008.
- [138] P. Yang, Q. Liu, and D. N. Metaxas. Boosting coded dynamic features for facial action units and facial expression recognition. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–6, 2007.
- [139] M. Yeasin, B. Bullot, and R. Sharma. From facial expression to level of interest: A spatio-temporal approach. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:922–927, 2004.
- [140] Y. Zhang and Q. Ji. Active and dynamic information fusion for facial expression understanding from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):699–714, 2005.
- [141] G. Zhao. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928, 2007.

Vita

Zhiguo Li

Education

2010, Ph.D. in Computer Science, Rutgers University, New Jersey
2008, M.Sc. in Statistics, Rutgers University, New Jersey
2002, M.Sc. in Computer Science, Chinese Academy of Sciences, Beijing, China
1999, B.Sc. in Computer Science, Xidian University, Xi'an, China

Experiences

May 2008 - Aug. 2008, Intern at NEC Laboratories America, Princeton, NJ
June 2002 - Jan. 2003, Software Engineer at Fanyou Technology, Shenzhen, China