UNCONSTRAINED FACE LANDMARK LOCALIZATION: ALGORITHMS AND APPLICATIONS

BY XIANG YU

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, 2015

ABSTRACT OF THE DISSERTATION

Unconstrained Face Landmark Localization: Algorithms and Applications

by Xiang Yu

Dissertation Director: Dimitris N. Metaxas

Nowadays, facial landmark localization in unconstrained environments has attracted increasing attention in computer vision, which is a fundamental step in face recognition, expression recognition, face tracking, editing, face animation, etc. We firstly introduce the problem of facial landmark localization and its relevant canonical and state-of-the-art techniques. Among the existed methods, when facilitating to the facial images under unconstrained environments, they may encounter problems from the large pose variation, partial occlusion, unpredictable illumination, etc. We then separately investigate each of the pose variation and partial occlusion problems. To overcome the shape variation caused by the pose changes, we propose an optimized part mixture model to fast search in the pose manifold and a bi-stage cascaded deformable shape model to refine the local shape variance. For partial occlusion, we propose a consensus of occlusion-specific regressors framework, which resists from the occlusion due to the large amount of regressors and the particularly designed occlusion patterns. Further, we aim at building a unified framework to jointly deal with the pose and occlusion problems. A pose-conditioned hierarchical part based regression method is designed to condition the pose into several pre-defined subspaces and localize the key positions in a hierarchical way, in which the occlusion is detected by the part regressors and further propagated through the hierarchical structure. The proposed facial landmark localization methods have shown more

promising performance than those state-of-the-arts in both accuracy and efficiency, compared on both lab-environmental databases and multiple challenging faces-in-the-wild databases. Our face alignment methods are further applied to some human-computer interaction (HCI) applications, i.e. user-defined expression recognition and face and gesture based visual deception detection. The improved results from the applications further validate the advantages of our method under all kinds of uncontrolled conditions.

Acknowledgements

I would like to take this opportunity to thank all the people, who are more than helpful during my Ph.D. study. Even these words cannot express my gratitude to their kind and encouraging supports.

I am extremely grateful to my advisor, Prof. Dimitris N. Metaxas, for his guidance and supports throughout my Ph.D. years. He provided me excellent environments for the study and research, and the visionary suggestions towards the most exciting challenges and application areas in our field. His wisdom and experiences taught me how to independently think on problems and life. Without his help, none of the work in this thesis would have been achieved.

I also would like to show my sincere thanks to my thesis committee members, Prof. Ahmad Elgammal, Prof. Kostas Bekris, Prof. Norman I. Badler and my qualifying exam committee member Prof. Michael Grigoriadis, for their constructive comments regarding my early proposal and this thesis. I wish Prof. Grigoriadis having a peaceful retired life. It is my honor to have each of them in my committees.

The research work in this thesis has received many helps from my other collaborators besides my advisors. Without them, some of the chapters in this thesis cannot be accomplished. My special thanks go to Prof. Junzhou Huang, Prof. Shaoting Zhang, Dr. Zhe Lin and Dr. Jianchao Yang. They selflessly shared professional and valuable experiences with me, from which I learned a lot in the collaboration.

It has been a most memorable period when I was in my Ph.D. study at the Center of Computational Biomedicine and Imaging Modeling and the department of Computer Science. It is my pleasure to meet all those kind and genius people there. Thanks to them who made my time meaningful and excited.

I am deeply grateful to my family for their understanding and strong support. I owe the time to accompany them as a son, a husband and a father. My special thanks go to my parents

for their consistent encouragement and support. I also would like to show more than thanks to my parents-in-law for their unlimited contribution. Their long-lasting taking care of me and my family saves a lot on my research and this thesis. Without them, I would not have reached the current step. Finally, I am extremely grateful to the gifts in my life, my wife Xin and my daughter Lillian. No word could express my gratitude to Xin for her love and sharing in my life. She gave up her own Ph.D. opportunity and support the family all through. This thesis is evidenced as a part of her efforts. I cannot cherish more from Lillian for her love, trust and the happiness that she brings.

Dedication

To my parents, my parents-in-law, my wife Xin and my daughter Lillian for their love, trust and support.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AAM	Active Appearance Model [35]
AFW	Annotated Faces-in-the-Wild [226]
ASL	American Sign Language
ASM	Active Shape Model [40]
CLM	Constrained Local Model [43]
CK+	Cohn-Kanade Plus Database [93, 123]
CMC	Computer-Mediated Communication
CNN	Convolutional Neural Network
COFW	Caltech Occluded Faces in the Wild [25]
CoR	Consensus of Regression [207]
DPM	Deformable Part Model [62]
EM	Expectation Maximization
FtF	Face-to-Face
GMM	Gaussian Mixture Model [143]
HCI	Human-Computer Interaction
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradient [45]
HPR	Hierarchical Part-based Regression
LBP	Local Binary Pattern [159]
LFPW	Labeled Facial Parts in the Wild [14]
LFW	Labeled Faces in the Wild [84]
MultiPIE	Multi-Pose, Illumination and Expression Face Database [74]
MAP	Maximum A Posteriori
MLE	Maximum Likelihood Estimate
MRF	Markov Random Field
OPM	Optimized Part Mixture Model [206]
PCA	Principal Component Analysis
PDM	Point Distribution Model
RANSAC	RANdom SAmple Consensus [64]
RBM	Restricted Boltzmann Machine
RCPR	Robust Cascaded Pose Regression [25]
SDM	Supervised Descent Method [198]
SIFT	Scale Invariant Feature Transform [120]
SVM	Support Vector Machine
TSPM	Tree Structure Part-based Model [226]
3DFE	3 Dimensional Facial Expression [204]
4DFE	4 Dimensional Facial Expression [203]

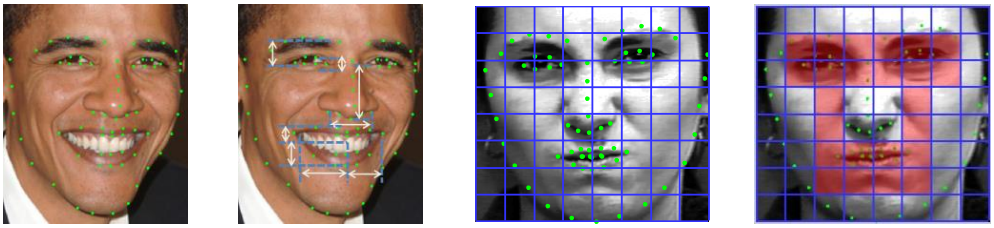
Chapter 1

Introduction

With the rapid development of massive computing and internet, machines have taken more roles in the more efficient automatic process. One important evidence is the popularization of smart phones. Those devices allow fingerprint identification, track users' behavior habits and predict the users' possible actions, conduct verbal interaction with users, operate intelligent house conditioning by monitoring the parameters and remote control. The Human-Computer Interaction (HCI), has been a critical and largely being investigated field, for the realization of artificial intelligence. Many HCI topics have attracted large attention, for example, the human identification (e.g. face recognition), body and face tracking, video surveillance (e.g. airport security camera), human activity monitoring (e.g. children autism investigation), etc.

Among those applications, human is one of the major subjects being investigated, whereas face is a most informative source of human. Localizing, tracking and identifying faces in real world with computers become feasible due to the advances of computers and cameras. In fact, many HCI applications are highly related with faces, e.g. face recognition, facial expression recognition, face tracking, facial motion capture, etc. Face detection and facial key point (landmark) localization are two critical techniques in those applications. Face detection provides the bounding boxes of faces for latter applications. Traditional face recognition or expression recognition extracts features directly based on the region provided by the face detectors. The detected faces are shown in Figure 1.1 (a) and (c).

However, without finely registration, the spatial mismatch brings in large noise. Thus, localizing the landmark points to register all the faces is a prerequisite for many applications, i.e. the detected landmarks, green dots in Figure 1.1 (b), provides accurate geometric information shown as white arrows; The detected landmarks in Figure 1.1 (d) provides automatic consistent region selection denoted as red blocks based on the detected landmarks in green



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

Figure 1.1: A facial image of President Obama (a), (b) and an image from CK+ database [93, 123] (c) and (d). (a) and (c) show the cropped facial area provided by face detectors. (b) shows the geometric feature extracted based on the detected landmarks in green dots. (d) shows the automatic region selection in red blocks based on the detected facial landmarks in green dots.

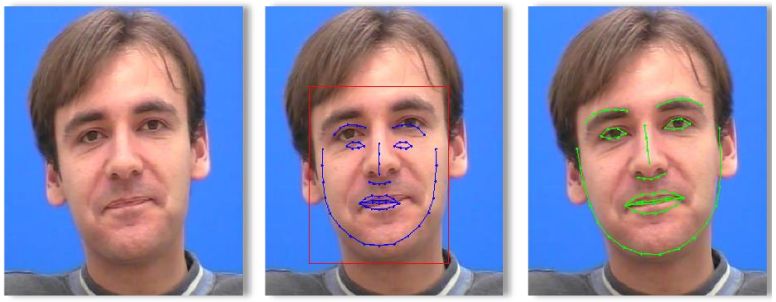
dots. Obviously, the face landmark localization effectively boosts the accuracy of the face and expression recognition tasks.

In this thesis, we focus on the facial landmark localization problem. Faces are objects with fixed components, i.e. eyebrows and eyes on top, nose in the center and mouth at the bottom. They are near-rigid objects with tons of local variances because of the diversity of people. A deformable model simulates such facial variance well. When the faces are explored under both controlled and uncontrolled environments, our effort is to fast and accurately find out the consistent facial patterns and deform the initial landmarks to the optimal positions.

1.1 Problem

The fixed components of faces allow the consistent definition of landmarks. Suggested by Cootes et al. [40], face landmarks can be divided into three categories: the points with application-dependent significance, i.e. the corners of eyes; the points labeling application-independent elements, i.e. curvature extrema (highest point along the bridge of the nose); and points interpolated from points of the previous two types, i.e. points along the face profile. Due to the purposes of different applications and the variant interpolating points, the annotation of the face landmarks varies, e.g. 68-point annotation from MultiPIE [74] and 194-point annotation from Helen [105].

As shown in Figure 1.1 (a), the green dots are the defined landmarks. By arranging them



(a) (b) (c)

Figure 1.2: The illustration of face landmark localization task. (a) a facial image from Talking Face video [1]. (b) the initialization of face landmarks in blue dots and lines from the face detection [90] result denoted in red bounding box. (c) The optimized landmarks denoted in green dots and lines deformed from the initial face landmarks in (b).

into some order, we form a shape $\mathbf{s} = (\mathbf{s}_1, ..., \mathbf{s}_n)$, where $\mathbf{s}_i$ denotes the location of the i-th landmark and n is the number of points (n=66 in Figure 1.1). After definition of the face landmarks, our task is illustrated in Figure 1.2.

Given a facial image as shown in Figure 1.2 (a), the first step is to initialize the face landmarks onto the facial area provided by face detectors, denoted as red bounding box. The initial landmarks are denoted as $\mathbf{s}_0$, blue dots and lines in Figure 1.2 (b). Then the core step is to deform $\mathbf{s}_0$ into the optimal shape $\hat{\mathbf{s}}$, denoted in green dots and lines in Figure 1.2 (c), which best approximate the ground truth $\mathbf{s}^*$.

The landmark deformation generally consists of two stages: in the training stage, a model is learned from the geometric information and/or the appearance variations mapping to the shape variations; in the testing stage, the learned model is applied on a testing facial image to predict the face landmarks. The procedure usually starts from a coarse initialization, followed by a step-by-step refinement until convergence.

1.2 Related Work

Most of the landmark localization methods firmly rely on the face detection results. With respect to face detection, Viola and Jones [90] proposed a widely used boost framework. It is fast and effective for most frontal faces. Many further improvements and modifications are made

based on the Viola and Jones detector. Some focused on extracting more effective features [91] and others concentrated on classification learning methods [62]. Nonetheless, several detection based landmark localization approaches are proposed to directly provide the landmark positions without knowing the face region. Sivic et al. [58] used mixture of tree structure to optimize the landmark positions of the whole face. Karlinsky et al. [95] exhibited face component detector learning to ensemble the facial component detectors and parse the facial attributes. Uricar et al. [177] proposed a seven anchor point detector based on Deformable Part Models (DPM) [63] which depicts an object with several parts and the connection in between each other part. They adopted structure-output SVMs to further localize the landmarks which achieve fast speed and high accuracy. However, when the detection error occurs, seven points are not sufficient to provide steady initial landmarks. Zhu and Ramanan [226] proposed another framework based on mixture of part model. Different such mixtures can handle different view-point faces. However, the size of parts pool in their model is large, which impedes the potential for real-time landmark tracking.

Facial landmark localization methods can be roughly divided into two major categories: parametric vs. non-parametric. Parametric methods are characterized by a model that attempts to capture facial appearance variations in terms of an underlying parameter space. Inference amounts to search in parameter space for the best-fitting model to the given image. In contrast, non-parametric methods learn to predict the face shape via training on a database, or by directly drawing exemplars in a data-driven manner.

The parametric methods essentially apply a rigid transform to connect the arbitrary shape with the reference shape. In other words, researchers attempt to build the relationship between the reference shape and any shape in the testing. A main such transform is the Point Distribution Model (PDM) [40]. It utilizes scaling, rotation (pitch, yaw and roll in 3D and in-plane rotation in 2D) and translation for the spatial rigid transformation. In the meanwhile, it applies the Principal Component Analysis (PCA) to depict the local shape deformation. It assumes the shape space could be represented by the linear combination of the trained shape basis, which is a linear approximation. In addition to PDM, there are several improvements on the prior shape distribution. Sozou et al. [165, 165] explored polynomial regression and multi-layer perception to depict the shape space other than the linear decomposition. Gu and Kanade [76] proposed

a 3D face alignment method based on a single image with a 3D PDM. A weak perspective projection is applied to project the 3D shapes to the 2D plane. De la Torre and Nguyen [102] proposed a kernel PCA-based nonlinear shape model. A mixture of Gaussian is also explored for the shape distribution [38, 58, 163]. Saragih [157] explored principal regression analysis other than PCA to represent the shape space. To further make the shape representation more robust, Li et al. [110, 111] introduced a robust shape model conducting random sample consensus [64].

The Active Shape Model (ASM) [40] and Active Appearance Model (AAM) [35] are both classical, seminal contributions to the parametric approach, with much follow-on work. Roh et al. [151] improved the original ASM by imposing the M-estimator and random sampling as noise is not always Gaussian distributed. Zhou et al. [223] proposed a MAP framework to formulate the shape in the tangent space and an EM algorithm to solve the MAP problem. Vogler et al. [183] proposed a 2D and 3D combination method for the ASM fitting. They proposed a 3D deformable model to constrain over the 2D facial key point tracking. Milborrow et al. [135] improved the original ASM framework by extending the 1D feature (i.e. image gradient) to the 2D feature. ASM with component construction was proposed [85, 112] to reduce the alignment error propagated among facial components. In the connection of those components, tree-structure is widely applied [218]. Markov Random Field (MRF) is also applied as the connection structure [171]. Besides, Le et al. [105] introduced a Viterbi process on facial contour fitting and user interaction model to improve the accuracy.

The AAM decouples the model into a linear shape model and a linear appearance model [68]. The model is originally proposed in [35]. Then a Gauss-Newton optimization method is proposed to solve the optimization problem [36]. Considering the training efficiency, a regression learned from low-dimensional texture difference to the position displacement is proposed [83]. Tresadern et al. [172] explored Haar-like features for the inexpensive linear projection from the texture difference to the position update. For the fitting efficiency, Matthews and Baker [131] revisited the AAM model with a inverse compositional method [9]. Gross et al. [73] further modified the inverse composition algorithm by simultaneously updating the warp parameters and the texture parameters.Another modification of AAM [141] is proposed

as fitting algorithm adaptation and the mean shape adaptation by incorporating the prior information. A mixed inverse-compositional-forward-additive parameter update scheme [155] is proposed to optimize the parameters between the image and the model. Some methods translate the spatial appearance matching problem to the Fourier domain problem [4, 124]. Recently a fast AAM algorithm was presented for real time alignment [176], and an ensemble of AAM [29] was proposed to jointly register landmarks for image sequence. In addition, the linear regression assumption from appearance update to the parameter update does not always hold. The sensitivity problem of AAM is a lasting problem. If the initialization is not proper, the linear regression may fail. Several efforts have been proposed. Liu [118, 119] explored GentleBoost [66] to model the nonlinear relationship between appearance and parameter update. Wu et al. [195] introduced classifiers to predict which set of parameters to use during the fitting processing. Further improvements [67] replaced the GentleBoost classifier to the gradient boosted regression trees [65].

Subsequently, the Constrained Local Model (CLM) [43, 157] was introduced, which combines each local patch's alignment likelihood and predicts the optimal solution by maximizing the overall alignment likelihood. The local response map plays an important role in the CLM fitting. Originally the response map is modeled as a isotropic Gaussian estimation. Wang et al. [191] proposed an anisotropic Gaussian estimation for each response map. Gu and Kanade [77] applied GMM to approximate the response map. A subspace constraining over the estimation on the response map [156] is introduced to enhance the localization. Local neural expert inference [11] is also proposed to improve the representability of the response map. The CLM framework [157] is extended to 3D situation using depth map [10]. The combination of a part model and CLM [206] was proposed to alleviate pose variations, while other CLM frameworks focused on local patch expert learning [5].

To handle the pose variation during shape fitting, Xiao et al. [197] explored 3D linear face model with the 2D face model and showed that the combination performs better in shape representation and real-time fitting. Retrieval from exemplars provides a way to tell the head pose information [75]. Regression on pose is also proposed on the 3D scenario [6]. 3D PDM and 2D model combination et al. [183, 130] is proposed under a full perspective projection. Depth map is also utilized to recover the 3D model and thus predict the pose information [59, 10].

Multi-view face shape models [110, 226] advocate another way to tackle the problem with discrete view angle intervals. Dantone et al. [46] proposed a pose conditioning tree to categorize the face shapes under different head poses into different subspaces.

With respect to non-parametric methods, the seminal work of Belhumeur et al. [14] proposed a data-driven method that employed RANSAC [64] to robustly fit exemplar landmark configurations drawn from a database to a set of local landmark detections. Similar methods [164, 220] either considered temporal feature similarity for joint face alignment or used graph matching to enhance the landmark localization. Notably, Zhu et al. [226] modeled the landmarks as a tree so that the positions could be efficiently optimized through dynamic programming. Shen et al. [161] employed the retrieval of example faces to help localize the testing face. Deep learning [82] has been explored in aligning faces, i.e. Luo et al. [125] applied Restricted Boltzmann Machine (RBM) in a hierarchical face parsing framework; Sun et al. [] proposed a three level cascaded deep Convolutional Neural Network (CNN) for landmark detection in coarse-to-fine manner.

Regression-based methods represent a significant sub-category of the non-parametric approach that have recently achieved high accuracy on standard benchmarks. An early contribution in this domain is Liang et al. [112], who proposed directional classifiers to predict the direction and step size of a landmark's update. Cristinacce and Cootes employed boosted regression [44] for local landmark alignment. Regression forest voting for accurate shape fitting was proposed by Cootes et al [37]. Valstar et al. [178] combined boosted regression with a graph model. Martinez et al. [129] proposed local evidence aggregation for regression based alignment. Dantone et al. [46] introduced conditional regression forests to treat faces with different poses separately. Dollar et al. [53] proposed cascaded pose regression to approximate 2D pose of objects. Rivera and Martinez [150] use kernel regression to handle low resolution images. Cao et al. [27] proposed a real-time explicit holistic shape regression method with robust shape indexed features. Xiong and De la Torre [198] illustrated an efficient supervised descent method for regression training and inference. Yang, et al. [201] employed dense interest points detection with sieving regression forests to obtain good results on faces in the wild. For efficiency consideration, a feature learning strategy on the regression framework is proposed to achieve three thousand frame per second processing speed [144]. Kazemi et al. [98] proposed

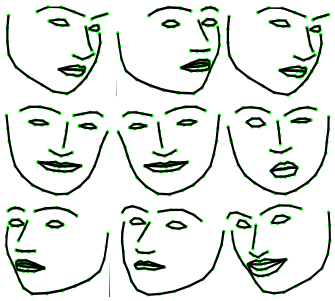


Figure 1.3: The shape samples from training images.

the ensemble of regression trees which achieves one millisecond processing time. To handle occlusion, Artizzu et al. [25] proposed a block-wise strategy to incorporate the occlusion prior for landmark fitting. Yu et al. [207] proposed a consensus of multiple occlusion-specific regressors to robustly predict the landmarks. Recently, the variations of the regression structure are proposed, such as the project-out cascaded regression [175] and the cascaded Gaussian process regression trees [106].

In the following, the subsections introduce the branches of the mainstream methods, i.e. ASM, AAM, CLM and shape regression method. We revisit each of the representative methods and show the similarities and dissimilarities of the methods at the optimization level. Further a general framework is summarized across all the methods. The general framework is expected to potentially provide the inspiration for new methods.

1.2.1 Active Shape Model

The Active Shape Model starts from modeling the shape distribution. During training, suppose we obtain a set of face landmarks to form a training shape space, illustrated in Figure 1.3. The training shape distribution is modeled under Principal Component Analysis (PCA) assumption. The PCA model not only provides a linear partition of the training shape space but also reduces the dimensionality of the shape space. Then any shape can be represented as in (1.1), the linear combination of the mean shape $\bar{\mathbf{s}}$ and the shape basis as the column vectors in $\mathbf{P}$.

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{P}\alpha + \epsilon \tag{1.1}$$

where $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_t)$ contains t eigenvectors of the covariance matrix of the training shapes and α is a t dimensional vector given by

$$\alpha = \mathbf{P}^T(\mathbf{s} - \bar{\mathbf{s}}) \tag{1.2}$$

t is usually determined by preserving 90% to 98% variance (the ratio between the sum of t largest eigenvalues and sum of all eigenvalues). ϵ is the noise random variable confirmed to a Gaussian distribution $\mathcal{N}(0, \sigma^2\mathbb{I})$.

Considering arbitrary shape which is not normalized from the reference shape, Procrustes analysis [99] is applied as the rigid transform to match between the reference shape and the being processed shape. The rigid transform consists of scaling a, rotation $\mathbf{R}(\theta)$ and translation $\mathbf{t}$, which is also known as the Point Distribution Model (PDM) in (1.3).

$$\mathbf{s} = a\mathbf{R}(\theta)\left(\bar{\mathbf{s}} + \mathbf{P}\alpha\right) + \mathbf{t} \tag{1.3}$$

Notice that the shape $\mathbf{s}$ is a vector, which could represent not only 2D coordinates but also 3D and maybe higher dimensions. Correspondingly, the rotation matrix $\mathbf{R}(\theta)$ could be either a 2D or 3D matrix. θ is the rotation angle. In 2D condition, θ represents the in-plane rotation angle. In 3D condition, θ represents three rotation angles, pitch, yaw and roll. The shift vector $\mathbf{t}$ is either 2D or 3D according to the shape vector. Denoting $\mathcal{P} = [a, \mathbf{R}(\theta), \alpha, \mathbf{t}]$, any arbitrary shape $\mathbf{s}$ is directly connected with the mean shape $\bar{\mathbf{s}}$. We denote it as:

$$\mathbf{s} = T(\bar{\mathbf{s}}, \mathcal{P}) \tag{1.4}$$

By assumption that $\epsilon \sim \mathcal{N}(0, \sigma^2\mathbb{I})$, maximum likelihood estimate (MLE) of $p(\epsilon)$ leads to the least square optimization:

$$\arg\min_{\mathcal{P}} \|\mathbf{s}^* - T(\bar{\mathbf{s}}, \mathcal{P})\|^2 \tag{1.5}$$

However, in the testing, we do not know the ground truth positions $\mathbf{s}^*$. Practically, the ASM configures an iterative algorithm: given initial landmarks $\mathbf{s}_0$, the first step is to find the largest response in the neighborhood of each landmark, e.g. the response could be image gradient. After all the local searches, the new shape is denoted as $\mathbf{s}'$. Then by Decomposing shape $\mathbf{s}'$ using (eq:coeff), the reconstructed shape is $\tilde{\mathbf{s}}$. Such alternative iteration continues until convergence.

The first round reconstructed shape $\tilde{\mathbf{s}}$ is actually $\mathbf{s}_0$. Our goal is to maximize a posteriori (MAP) over $p(\mathbf{s}', \mathcal{P}|\tilde{\mathbf{s}})$.

$$\arg\max p(\mathbf{s}', \mathcal{P}|\tilde{\mathbf{s}}) = \arg\max p(\mathcal{P}|\mathbf{s}')p(\mathbf{s}'|\tilde{\mathbf{s}}) \tag{1.6}$$

Assume that parameter $\mathcal{P} \sim \mathcal{N}(0, \Lambda)$. By optimizing () with $-log(.)$, we obtain the equivalent objective formula in (1.7).

$$\begin{aligned} &\arg\min_{\mathcal{P}, \tilde{\mathbf{s}}} \|\mathcal{P}\|^2_\Lambda + \sum_{i=1}^{n} \|\nabla \mathbf{I}(\tilde{\mathbf{s}}_i) - \nabla \mathbf{I}(y_i)\|^2, \\ &y_i = \arg\max_{y' \in \Psi_i} \nabla \mathbf{I}(y') \end{aligned} \tag{1.7}$$

where y_i is the largest gradient response in landmark $\tilde{\mathbf{s}}_i$'s neighborhood Ψ_i. The objective function (1.7) consists of two parts. The first part is the shape regularization from the Gaussian shape prior. The second part is the penalty from the observation, i.e. the image gradient distance from the largest response. By jointly optimizing the two parts iteratively, the shape is expected to converge.

Since ASM only looks into each landmark's neighborhood for the largest response, the algorithm could easily fall into local minima. In the mean while, the image gradient of neighborhood cannot collect sufficient appearance information for the landmarks' update, which weakens the robustness of the method.

1.2.2 Active Appearance Model

Based on the ASM assumption, Cootes et al. proposed another famous framework, the active appearance model (AAM) [35]. They added the holistic facial appearance as a stronger evidence other than the local neighborhood response, expecting better overcoming the local minima problem. They model the holistic appearance distribution as an appearance PCA space. Besides, the shape model is the same as the ASM model, which means the shape is also controlled by the parameters $\mathcal{P}$. The framework is similar to the shape modeling, as shown in (1.8).

$$\mathbf{I}(\mathbf{s}) = \bar{\mathbf{a}} + \mathbf{A}\beta + \eta \tag{1.8}$$

where $\mathbf{I}(\mathbf{s})$ means the facial area cropped out according to the shape $\mathbf{s}$. $\bar{\mathbf{a}}$ is the mean face appearance from the training data. $\mathbf{A} = [\mathbf{A}_0, ..., \mathbf{A}_K]$, formed by the K appearance basis vectors. β is the coefficients linearly combining the appearance basis with the mean appearance, which aims to best represent the test face. η is a random vector confirms to $\mathcal{N}(0, (\sigma')^2\mathbb{I})$.

Combining (1.8) and (1.1), given an initial shape $\mathbf{s}_0$, the goal is to minimize the error between current facial appearance $\mathbf{I}(\mathbf{s})$ and the linear combination $\bar{\mathbf{a}} + \mathbf{A}\beta$. From the assumption that $\eta \sim \mathcal{N}(0, \sigma'^2\mathbb{I})$, the objective function to optimize $\mathcal{P}$ and β is in (1.9).

$$\arg\min_{\mathbf{s},\beta} \|\mathbf{I}(\mathbf{s}) - (\bar{\mathbf{a}} + \mathbf{A}\beta)\|^2 \tag{1.9}$$

Further assuming $\beta \sim \mathcal{N}(0, \Sigma)$, similar MAP optimization is applied on the AAM model.

$$\arg\max p(\mathbf{I}(\mathbf{s}), \mathcal{P}, \beta|\mathbf{s}_0) = \arg\max p(\mathcal{P}|\Lambda)p(\beta|\Sigma)p(\mathbf{I}(\mathbf{s})|\mathcal{P}, \beta, \mathbf{s}_0) \tag{1.10}$$

By taking $-log(.)$ over (1.10), we obtain the objective function in the following.

$$\arg\min_{\mathcal{P},\beta} \|\mathcal{P}\|^2_\Lambda + \|\beta\|^2_\Sigma + \|\mathbf{I}(\mathbf{s}) - (\bar{\mathbf{a}} + \mathbf{A}\beta)\|^2 \tag{1.11}$$

Since $\mathbf{s}$ is determined by $\mathcal{P}$ and $\bar{\mathbf{s}}$ and $\mathbf{P}$, the appearance $\mathbf{I}(s)$ is determined by $\mathcal{P}$. Thus, the optimization in (1.11) is over $\mathcal{P}$ and β. From the objective function, similar to (1.7), it can be divided into two parts. The first part is coefficients' regularization term, including $\|\mathcal{P}\|^2_\Lambda + \|\beta\|^2_\Sigma$. The second part is the penalty term, $\|\mathbf{I}(\mathbf{s}) - (\bar{\mathbf{a}} + \mathbf{A}\beta)\|^2$, indicating the reconstruction error of facial appearance.

The difference from (1.7) is that there are two items to be optimized, $\mathcal{P}$ and β. The problem is convex. Traditional gradient descent method is effective in solving the problem, i.e. Levenberg-Marquardt was used in [158], a stochastic gradient descent algorithm was used in [17, 89]. Solving the problem needs to alternatively optimize each of the two items.

The Lukas-Kanade Image Alignment [122] takes the image alignment as a warp. They took the template image as reference. The image at shape $\mathbf{s}$ after warping with parameter $\mathcal{P}$ should best match the template image. They linearize the update procedure by importing $\nabla\mathcal{P}$. According to the Taylor series expansion, as long as the $\nabla\mathcal{P}$ is as near in the neighborhood of $\mathcal{P}$ as possible, the update of $\mathcal{P}$ is achieved by $\mathcal{P} + \nabla\mathcal{P}$.

The above methods suffers from the low efficiency because the partial derivatives, Jacobian, Hessian and gradient direction need to be recomputed at each iteration. Matthews and Baker

proposed an inverse compositional way to fastly solve the problem [131]. They reverse the warped image and the template appearance by adding the $\nabla\mathcal{P}$ warp onto the template and the warped image keeps the same. In this way, the increment is with respect to the template appearance, in which way the calculation can be achieved only in the preprocessing step. For further details, [131] provides sufficient comparisons on different solving strategies.

1.2.3 Constrained Local Model

The constrained local model shares the same PDM to represent any arbitrary shape. The parameters could also be denoted as $\mathcal{P}$. The method is similar to ASM. The difference is that, other than finding the largest image gradient response at each iteration, the CLM treat each landmarks independently. Each landmark patch is fully utilized to evaluate this landmark's alignment likelihood. The overall alignment likelihood is the product of each landmark's alignment likelihood by the landmark independence assumption. The local patch probabilistic update is considered more robust than finding the largest gradient response.

To evaluate the alignment likelihood, CLM introduces a latent variable v_i for each landmark $\mathbf{s}_i$, which is a binary discrete random variable indicating well alignment ($v_i = 1$) or misalignment ($v_i = 0$). Then the posterior probability of the parameter $\mathcal{P}$ is denoted as:

$$p(\mathcal{P}|\{v_i = 1\}_{i=1}^{n}, I) \propto p(\mathcal{P}) \prod_{i=1}^{n} [p(v_i = 1|x_i, I)] \tag{1.12}$$

The alignment evaluation can be modeled as a logistic regression model in (1.13). In training, the positive samples could only be aligned patches while negative samples could be any misaligned case.

$$p(v_i = 1|x_i, \mathbf{I}) = \frac{1}{1 + exp(\omega f + \gamma)} \tag{1.13}$$

where f is the feature vector extracted at the patch surrounding $\mathbf{s}_i$. The feature vector typically could be HoG [45] or SIFT [120] feature. x_i is the current iteration's landmark position.

Generally given a near-optimal landmark $\mathbf{s}_i$, the algorithm searches its neighborhood to get the optimal alignment likelihood. The possible optimal candidates y_i form a region Ψ_i. Assume that y_i confirms to Gaussian distribution with mean $\mathbf{s}_i$ and σ_i standard deviation. Hence, the alignment likelihood can be modeled as a Gaussian Mixture Model (GMM) [143] of the

candidates y_i.

$$\begin{aligned} p(v_i = 1|\mathbf{s}_i, \mathbf{I}) &= \sum_{y_i \in \Psi_i} p(v_i = 1|y_i, I) p(y_i|\mathbf{s}_i, \mathbf{I}) \\ &= \sum_{y_i \in \Psi_i} \pi_{y_i} \mathcal{N}(\mathbf{s}_i, \sigma_i^2 \mathbf{I}) \end{aligned} \tag{1.14}$$

where $\pi_{y_i} = p(v_i = 1|y_i, I)$. Taking (1.14) into (1.12), the objective function is shown in (1.15).

$$p(\mathcal{P}|\{v_i = 1\}_{i=1}^n, \mathbf{I}) \propto p(\mathcal{P}) \prod_{i=1}^{n} \left(\sum_{y_i \in \Psi_i} \pi_{y_i} \mathcal{N}(\mathbf{s}_i, \sigma_i^2 \mathbf{I}) \right) \tag{1.15}$$

Further by Bayesian rule, $p(v_i = 1|\mathbf{s}_i, I)$ can be represented as $\frac{p(v_i=1, y_i|\mathbf{s}_i, I)}{p(y_i|v_i=1, \mathbf{s}_i, I)}$. An Expectation Maximization (EM) approach is raised to solve the problem. The **E** step is to solve the posterior probability of latent variable y_i as $p(y_i|v_i = 1, \mathbf{s}_i, I)$.

$$p(y_i|v_i = 1, \mathbf{s}_i, I) = \frac{\pi_{y_i} \mathcal{N}(\mathbf{s}_i, \sigma_i \mathbf{I})}{\sum_{z_i \in \Psi_i} \pi_{z_i} \mathcal{N}(z_i, \sigma_i \mathbf{I})} \tag{1.16}$$

With intermittent latent variable posterior probability, we approximate $p(v_i = 1|\mathbf{s}_i, I)$ as shown in (1.17).

$$E_{p(y_i|v_i=1, \mathbf{s}_i, I)} \left(p(v_i = 1, y_i|\mathbf{s}_i, I) \right) \rightarrow p(v_i = 1|\mathbf{s}_i, I) \tag{1.17}$$

The **M** step is to minimize the expectation of negative log likelihood in (1.18).

$$\arg\min_{\mathcal{P}, \mathbf{s}_i} E_{q(y)} \left[-log \left\{ p(\mathcal{P}) \prod_{i=1}^{n} [p(v_i = 1, y_i|\mathbf{s}_i, I)] \right\} \right] \tag{1.18}$$

where $q(y) = \prod_{i=1}^{n} p(y_i|v_i = 1, \mathbf{s}_i, I)$. Equivalently,(1.18) is simplified as (1.19) shows.

$$\arg\min_{\mathcal{P}, \mathbf{s}_i} ||\mathcal{P}||_\Lambda^2 + \sum_{i=1}^{n} \sum_{y_i \in \Psi_i} \frac{\mu_{y_i}}{\sigma^2} ||\mathbf{s}_i - y_i||^2 \tag{1.19}$$

where μ_{y_i} denotes the probability $p(y_i|v_i = 1, \mathbf{s}_i, \mathbf{I}$. The objective function again consists of two parts. The first term is the parameter $\mathcal{P}$ regularization term. The second term is the penalty term: penalize the loss of current landmark $\mathbf{s}_i$ on the local response map formed by μ_{y_i} in region Ψ_i.

Table 1.1: The optimization structures of ASM, AAM and CLM. $R(\mathcal{P}, \beta)$ stands for the regularization term and $L(\mathbf{s}, \mathbf{I})$ is the penalty term.

Methods	$R(\mathcal{P}, \beta)$	$L(\mathbf{s}, \mathbf{I})$
ASM	$\|\mathcal{P}\|^2_\Lambda$	$\sum_{i=1}^{n} \|\nabla \mathbf{I}(\tilde{\mathbf{s}_i}) - \nabla \mathbf{I}(y_i)\|^2$
AAM	$\|\mathcal{P}\|^2_\Lambda + \|\beta\|^2_\Sigma$	$\|\mathbf{I}(\mathbf{s}) - (\bar{\mathbf{a}} + \mathbf{A}\beta)\|^2$
CLM	$\|\mathcal{P}\|^2_\Lambda$	$\sum_{i=1}^{n} \sum_{y_i \in \Psi_i} \frac{\mu_{y_i}}{\sigma^2} \|\mathbf{s}_i - y_i\|^2$

1.2.4 Structure of Parametric Methods

At this stage, we are clear about the parametric methods' structure. The optimization is usually constructed with two main parts: the parameter regularization part and the penalty part. By different update assumptions, the penalty part may be different. Since the parametric methods apply the same PDM model, the regularization term is almost similar across the methods.

A complete comparison of the three major parametric methods is shown in Table 1.1. Overall the objective functions are divided into two parts, $R(\mathcal{P}, \beta)$, the regularization term and $L(\mathbf{s}, \mathbf{I})$, the loss term or penalty term. We summarize the structure as shown in (1.20).

$$\arg\min_{\mathcal{P}, \beta} R(\mathcal{P}, \beta) + L(\mathbf{s}, \mathbf{I}) \tag{1.20}$$

The ASM assumes the update of landmarks is to find the largest gradient response. Correspondingly, the loss term is the loss of the current landmark's image gradient to the largest gradient. The CLM assumes the update to be the best alignment in the response map. Thus, the loss term is the deviation to each neighborhood position weighted by the likelihood of alignment. The AAM assumes the update should best match the reconstructed appearance. As a consequence, the loss is the appearance reconstruction error. In the parameter regularization, the only difference is the AAM. Because it introduces the appearance coefficients β besides the PDM parameter $\mathcal{P}$. We empirically extrapolate that the improvements of parametric methods are most likely to confirm to the optimization structure in (1.20).

1.2.5 Shape Regression

In the problem introduction, the goal of face shape alignment is to predict the landmark positions $\mathbf{s}$ which best estimates the true position $\mathbf{s}^*$.

$$\arg\min_{\mathbf{s}} \|\mathbf{s} - \mathbf{s}^*\|_2 \tag{1.21}$$

Shape regression methods do not utilize a set of parameters to control the shape, which is different from the parametric methods. Thus, the shape regression methods are the non-parametric methods. They attempt to build the direct relationship between the landmark update and the observation space. The observation space could be the image appearance, or the feature space. (4.1) is more practical than original objective function (1.21) because at some occasions appearance is easy to extract while ground truth landmark positions are absent.

$$\arg\min_{\Delta s} \|h(I(s_0 + \Delta s)) - \Phi^*\|_2^2 \tag{1.22}$$

Let $F(s_0 + \Delta s)$ denote $\|h(I(s_0 + \Delta s)) - \Phi^*\|_2^2$. h is defined as the feature extraction. $\mathbf{I}(\mathbf{s})$ is the facial patches cropped at each landmark $\mathbf{s}_i$. Φ^* is the reference feature vector either from ground truth or from training template. The objective is to find the coordinate displacement Δs such that the feature best matches reference feature in $l2$-norm, which is further denoted in (3.2).

$$\Delta s = -2H^{-1}J^T(\Phi_0 - \Phi^*) \tag{1.23}$$

where H is the Hessian of function F and J is the Jacobian of feature extraction function F. From (3.2), there is a linear relationship between the coordinate displacement Δs and feature difference $\Phi_0 - \Phi^*$. In [198], they proposed a linear regression based framework to model the relationship as shown in (4.2).

$$\Delta s = \mathbf{R}\Phi_0 + b \tag{1.24}$$

$\mathbf{R}$ is a regression matrix to approximate $-2H^{-1}J^T$ especially when H is singular. The ground truth feature Φ^* is represented by an intercept item b, which is advantageous for inference as there is no ground truth when test case appears. Φ_0 is a feature vector concatenated by n feature vectors extracted at each fiducial point $\mathbf{s}_i$, which indicates that each fiducial point's displacement is related with all other fiducial points' appearance.

Usually the initial landmarks may not localize in the neighborhood of the ground truth. Several rounds of the regressions are needed to approach the optima. Thus, cascaded regression framework [27, 198] is proposed. The shape update form is as follows:

$$\mathbf{s}_{t+1} = \mathbf{s}_t + \mathbf{R}_t\phi_{\mathbf{s}_t} + \mathbf{b}_t \tag{1.25}$$

where $\mathbf{R}_t$ is the regression matrix and $\mathbf{b}_t$ is the intercept of iteration t. $\phi_t = \mathbf{h}(I(\mathbf{s}_t))$ denotes a local feature descriptor. Typically the number of iterations is fixed to 4 or 5. The cascaded regression attempts to apply a set of linear regressions sequentially to predict landmark positions. Given ground truth $\mathbf{s}^*$, in the training process, we learn the regression matrix $\mathbf{R}_t$ and the intercept $\mathbf{b}_t$ by minimizing the prediction error over all training samples $\mathcal{T}$ as follows:

$$\arg\min_{\mathbf{R}_t,\mathbf{b}_t} \sum_{z\in\mathcal{T}} \|\mathbf{s}^* - (\mathbf{s}_t + \mathbf{R}_t\phi_{z,\mathbf{s}_t} + \mathbf{b}_t)\|_2^2 \tag{1.26}$$

The training is also a cascaded process. For each round training, the initialization is the output of the previous regression result. One may add in perturbation on top of the output as the input of the next round regression. In (1.26), t means the t-th round iteration. $\phi_{z,\mathbf{s}_t}$ means the feature extracted on sample z at the landmark $\mathbf{s}_t$. The problem is a well-defined convex optimization problem, which can be solved by gradient descent method. We may notice that the optimization term in (1.26) is the penalty term similar to the ones shown in Table 1.1. Since it is a non-parametric method, there is no parameter regularization in the optimization.

1.3 Motivation

The introduced relevant research has made large success in the face landmark localization field. The parametric model reduces the space dimension from the original shape dimension to the parameter dimension, which largely reduces the redundancy among the landmarks and speeds up the fitting. Meanwhile, the parametric model could more robustly localize key points because the holistic shape prior helps to overcome local minima. But because of the holistic shape constraint and the simplified parameter space, the parametric methods lacks capability to well describe the local shape variance. Although AAM methods incorporate the appearance information, the increased complexity of registering the appearance decreases the efficiency. Moreover, since most of the AAM based methods are gradient descent or Gauss-Newton solved, it

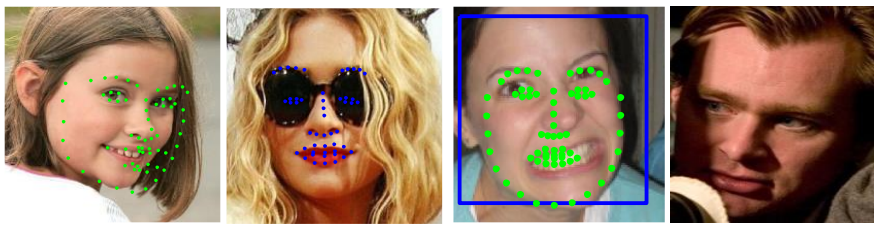


Figure 1.4: Sample images for unconstrained environments.

is sensitive to the initialization because there are many local minimum inside the images.

The developed alignment techniques almost saturate the controlled environment, i.e. lab environment with controlled lighting. In the recent several years, researchers focus more on the face-in-the-wild conditions, most of which are under unconstrained environments. For example, as shown in Figure 1.4, the head pose is arbitrary, which allows side-view faces for extreme conditions. The face appearance may not be complete, part of which may be occluded. Faces in the images may be with exaggerate expressions, i.e. mouth is widely open. The illumination could be arbitrary which brings in much shadow, fake edges, etc.

Under the unconstrained environment assumption, those off-the-shelf methods may encounter problems. A general holistic shape model cannot handle the variant head poses. The occlusion brings in missing of appearance which is a problem for the AAM-based methods because the missed appearance cannot be matched to the reconstructed appearance. The regression-based methods have shown its fast and accurate performance [14, 198]. But the fatal problem is that it is highly sensitive to the initialization. This is because the off-line trained searching space of the cascaded regression is fixed. If the initialization does not lie in the searching space, the latter cascaded regression error is enlarged.

Among the literature, we find several seminal work which attempted to handle one or several of the problems, i.e. the multi-view tree structure part-based model (TSPM) [226] applies a multi-view shape structure to compensate the pose variation while at the same time to detect the landmark positions; the robust cascaded pose regression (RCPR) [25] divides the images into regions and applies statistical analysis over the occlusion region which provides prior for the landmark regression. However, there are still some problems. Though TSPM simultaneously determines head poses and localize landmarks, the tree structure cannot preserve the full facial shape structure information and the detection is inefficient. The fixed region-wise occlusion

prior for RCPR cannot hold for many situations and the regression result is sensitive.

Based on the above investigations, we aim to develop more robust algorithms which could handle one or several of the pose variation, partial occlusion and expression variation problems. We explore both the parametric and regression-based methods and improves from the state-of-the-art methods by introducing novel and effective modules. Unifying those modules into a holistic framework is another effort from us. We also apply the proposed robust alignment methods to the face tracking, expression recognition and video-based deception detection tasks.

1.4 Organization

We arrange the content of this thesis into the following chapters. The algorithms and applications are introduced each at a chapter.

Chapter 2 presents a two-stage cascaded deformable shape model to effectively and efficiently localize facial landmarks with large head pose variations. In initialization stage, a group sparse optimized part mixture model (OPM) is presented to automatically select the most salient facial landmarks. In landmark localization stage, the first step applies mean-shift local search with constrained local model and the second step utilizes component-wise active contours to discriminatively refine the subtle shape variation. This framework simultaneously handles face detection, pose-robust landmark localization and tracking in real time. Extensive experiments are conducted on both laboratory environmental databases and face-in-the-wild databases to verify the method.

Chapter 3 addresses the problem of robust face alignment in the presence of occlusions. Recently, regression-based approaches to localization have produced accurate results in many cases, yet are still subject to significant error when portions of the face are occluded. To overcome this weakness, we propose an occlusion-robust regression method by forming a consensus from estimates arising from a set of occlusion-specific regressors. After localization, the occlusion state for each landmark point is estimated using a Gaussian MRF semi-supervised learning method. To mitigate sensitivity to initialization that is common for regression based methods, we introduce a max-margin learning strategy to normalize position, rotation and scale of the initial detection. Experiments on both non-occluded wild face databases and specific occlusion

wild face databases show the advantage of the proposed algorithm.

In chapter 4, we propose a robust two-stage hierarchical regression approach to deal with both the head pose variation and partial occlusion problems. First, a pose-dependent holistic regression model is introduced to initialize the facial landmarks under different head pose assumptions. Second, to reduce local shape variance, a hierarchical part-based regression method is further proposed to refine the global regression output. The part regressors are directly derived from the holistic regressors by our newly proposed projection optimization method. The occlusion state is simultaneously inferred at the part regression stage and propagate to other landmarks after the localization. Experiments on several challenging faces-in-the-wild datasets demonstrate the advantages of our method which is more robust to pose and occlusion when compared to the state-of-the-art.

In chapter 5, we investigate the user-defined expression recognition, which is an extended concept from the prototypic six expressions, i.e. angry, disgust, fear, happy, sad and surprise. A fused convolutional neural network (f-CNN) feature is designed to describe all the expressions. The CNN structure is trained on prototypic expression databases. Training on prototypic expressions and testing on customized ones may encounter domain bias problem. To overcome it, a regularizer, handcrafted feature is proposed, consisting of geometric feature and the structured patch learning appearance feature. An ensemble of few-shot SVM classifiers is designed to alleviate the few training sample problem, which is common in real-time situations. Based on the SVM result, finally an HMM-based temporal smoothing is applied at the score level. Experiments on both typical expression databases and a self-built user-defined expression database demonstrate the effectiveness of the proposed scheme.

Detecting deception in interpersonal dialogue is challenging since deceivers take advantage of the give-and-take of interaction to adapt to any sign of skepticism in an interlocutor's verbal and nonverbal feedback. Human detection accuracy is poor, often with no better than chance performance. In chapter 6, we consider whether automated methods can produce better results and if emphasizing the possible disruption in interactional synchrony can signal whether an interactant is truthful or deceptive. We propose a data-driven and unobtrusive framework using visual cues that consists of face tracking, head movement detection, facial expression recognition and interactional synchrony estimation. Analysis were conducted on 242 video samples

from an experiment in which deceivers and truth-tellers interacted with professional interviewers either face-to-face or through computer mediation. Results revealed that the framework is able to automatically track head movements and expressions of both interlocutors to extract normalized meaningful synchrony features and to learn classification models for deception recognition. Further experiments show that these features reliably capture interactional synchrony and efficiently discriminate deception from truth.

Chapter 7 concludes this thesis, summarizes the contributions of my thesis work, and outlines directions for the future research. The improvements in face alignment algorithms benefit the applications in localization accuracy and computation while the better results in the applications also complementary strengthen the proposed alignment algorithms.

Chapter 2

Pose-robust Landmark Localization

In this chapter, we investigate the head pose variation problem in the wild face alignment. The multi-view parametric models [197, 183] inspired a way to deal with pose variations. Recently, a deformable part model (DPM) based facial key point detection [226] further verifies the multi-view shape models. We start with an optimized key point selection scheme for more efficient landmark initialization. Then a bi-stage landmark fitting algorithm is presented, to further refine the shape variance. The proposed framework is shown real-time and has applied to the video tracking.

2.1 Introduction

Many face alignment algorithms rely on the facial area detection results, which is the first and key step in landmark localization. For example, in CLM [157] and SDM [198], their frameworks apply the famous face detector Viola and Jones [90]. Though general face detectors show good performance, it cannot handle the conditions with large pose variation. Therefore, the more state-of-the-art face detection algorithms are proposed to overcome the extreme conditions [211, 109]. Even provided with good region of interests under extreme conditions, improper landmark initialization still leads to misalignment. For example, Active Appearance Models (AAMs) [35] are very sensitive to initial positions, because complex appearance with illumination and noise may result in local minima.

Pose variation leads to the self-occlusion by viewing from certain view-points. The missing appearance provides no evidence for the algorithms to evaluate whether the landmarks (especially the landmarks in the missing region) are in proper positions. To alleviate the problem, Cootes and Taylor [38] imported mixture model for representing shape variation. Zhou et al. [224] also provided a Bayesian mixture model for multi-view face alignment. Although

multi-view face shape models partially solve the pose variation problem, they cannot cover unlimited possibilities of view changes. Therefore, 3D shape models [183] are proposed to handle continuous view change. There are two possible ways to explicitly project 3D shape onto 2D images. One way is to use facial anchor points, e.g. eye corners and mouth corners, mapping from 3D shape; The other is to leverage the view information from head pose estimators. Since most pose estimators [31] are based on face detectors, which makes the problem recursive. A better choice is to train fast and accurate facial anchor point detectors.

2.2 Related Work

For face detection, Viola and Jones [90] proposed a widely used framework. It is fast and effective for most near-frontal faces, but lacks flexibility dealing with large pose variations. Sivic et al. [58] used mixture of tree structure to estimate landmarks. Uricar et al. [177] proposed a seven-anchor point detector based on deformable part models (DPM) [62] and structure-output SVMs which achieve fast speed and high accuracy. However, when the detection error occurs, seven points are not sufficient to provide steady initial landmarks. Zhu and Ramanan [226] proposed another framework based on mixture of part model. However, the size of parts pool in their model is large, which impedes the potential for real-time landmark tracking.

Parametric models have been widely used in face alignment. Active Shape Model (ASM) [40, 44] and Active Appearance Model (AAM) [35, 131] have achieved good performance in face alignment. But it is difficult to represent face shapes merely using linear shape combination or appearance subspace in extremely varying views . Constrained Local Model (CLM) [43, 157, 191], another successful deformable fitting model, performs exhaustive local search and optimizes the overall likelihood of the landmarks' alignment. To alleviate the varying view problem, multi-view shape models [38, 224] were proposed either by local search to estimate the head pose or by incrementally combining models from different views.

2.3 Optimized Part Mixtures

Before shape alignment or landmark tracking, robust initialization promotes the performance and prevents the fitting process from falling into local minima. We follow a pictorial structure [63, 226] to organize the landmarks. Subsection 2.3.1 serves as preliminary background introduced in [226]. Based on that, we aim to simplify the dense structure of TSPM. Observing that not all dense landmarks are needed in order to localize the facial area, we introduced group sparse constraint over all the landmarks in subsection 2.3.2. To obtain the coefficients of each landmark, a max-margin learning framework is proposed in subsection 2.3.3. In the meanwhile, we further introduced an iterative updating algorithm to efficiently solve the group-sparse constraint problem.

2.3.1 Mixtures of Part Model

Every facial landmark with predefined patch neighborhood is a part. Same landmark in different viewpoints may be different parts. As a consequence, the landmarks of a face are a mixture of those parts. We define the shared pool of parts as V. The connection between two parts forms an edge in E. In connecting the landmarks, specific tree structures are superior to general complete graphical models for not only the simplicity of representation but also the efficiency in inference [226].

For each viewpoint i, we define a tree $T_i = (V_i, E_i), i \in \{1, 2, ..., M\}$. Given a facial image $I^{H \times W}$, the j^{th} landmark position $s_j = (x_j, y_j) \in \mathcal{S}_j \subset \{1, ..., H\} \times \{1, ..., W\}, j \in \{1, 2, ..., N\}$. The measuring of a landmark configuration $\mathbf{s} = (s_1, ..., s_N)$ is defined by a scoring function $f : I \times \mathcal{S} \to \mathbb{R}, \mathcal{S} = \{\mathcal{S}_1, ..., \mathcal{S}_N\}$.

$$f_i(I, \mathbf{s}) = \sum_{j \in V_i} q_i(I, s_j) + \sum_{(j,k) \in E_i} g_i(s_j, s_k) \tag{2.1}$$

The first term in (2.1) is a local patch appearance evaluation function $q_i : I \times \mathcal{S}_i \to \mathbb{R}, i \in (1, N)$, defined as,

$$q_i(I, s_j) = \left\langle \mathbf{w}_j^{iq}, \Phi_j^{iq}(I, s_j) \right\rangle \tag{2.2}$$

indicating how likely a landmark is in an aligned position. The second term is the shape deformation cost $g_i : \mathcal{S}_j \times \mathcal{S}_k \rightarrow \mathbb{R}, (j,k) \in E$, defined as,

$$g_i(s_j, s_k) = \left\langle \mathbf{w}_{jk}^{ig}, \Phi_{jk}^{ig}(s_j, s_k) \right\rangle \tag{2.3}$$

balancing the relative positions of neighboring landmarks. $\mathbf{w}_j^{iq}$ is the weight vector convolving the feature descriptor of patch j, $\Phi_j^{iq}(I, s_j)$. $\mathbf{w}_{jk}^{ig}$ are the weights controlling the shape displacement function defined as $\Phi_{jk}^{ig}(s_j, s_k) = (dx, dy, dx^2, dy^2)$, where $(dx, dy) = s_k - s_j$. Such quadratic deformation cost controls the model with only four parameters and has shown its effectiveness in face alignment [226]. Further, we formulate the two evaluation functions in a uniform way to obtain a more compact representation

$$f_i(I, \mathbf{s}) = \left\langle \tilde{\mathbf{w}}_i, \tilde{\Phi}_i \right\rangle \tag{2.4}$$

where $\tilde{\mathbf{w}}_i = [\mathbf{w}_j^{iq}, \mathbf{w}_{jk}^{ig}]$ and $\tilde{\Phi}_i = [\Phi_j^{iq}(I, s_j), \Phi_{jk}^{ig}(s_j, s_k)]$ for each viewpoint i.

Given an image I, for each possible configuration of landmark positions, we evaluate the score of each configuration in each viewpoint. The largest score potentially provides the most likely localization of the landmarks. Thus the landmark positions can be obtained by maximizing (2.5).

$$\mathbf{s}^* = \underset{\mathbf{s} \in \mathcal{S}, i \in (1,M)}{\arg\max} \; f_i(I, \mathbf{s}) \tag{2.5}$$

2.3.2 Group Sparse Learning for Landmark Selection

Facial landmarks are usually defined manually or human-selected without any consistent rules. Evidence is that the annotation among different face datasets is largely different, e.g. LFPW [14] database has 29 points, LFW [84] database has 7 points, while AR [128] database has 22 labels. However, we observe that there are some common points defined by those different datasets, such as eye corners, eyebrow corners, mouth corners, upper lip and lower lip points, etc. Although one can manually select the most common landmark points for a new facial structure, we intend to automatically select those landmarks by learning from training data to well represent facial structures and the number of landmarks should meet real-time requirement for inference.

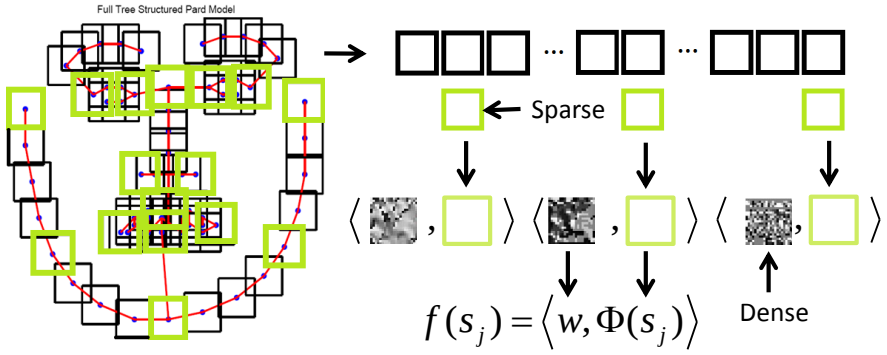


Figure 2.1: The group sparse structure illustration. The most salient boxes denoted as green comparing to all the boxes are sparse. Each box is considered a group. At the group level, the selection is sparse. While inside each box, the corresponding coefficient matrix denoted as the gray patch is dense because inside the area of each box, all the pixels contribute to the score $f(s_j)$ calculation.

The goal of sparse selection is to robustly initialize the landmark positions more than accurately localize the landmark positions. In fact, the landmarks which are more salient than others, i.e. the corner points of eyes and mouth, in some sense would be easier to detect. However, considering the harder landmarks to localize as TSPM [226] does, it is far from robust when severe pose variation or occlusion is present. Removing the vague points decreases the false alarms. In our framework, the learning based selection serving as initial detection and boosting the localization accuracy is placed on the latter two-stage deformable shape fitting, not the detection step. Based on the above observations, we intend to build an optimization algorithm to simplify the dense structure.

Visually salient key points have a higher probability to be selected as the optimized structure. Since the saliency significance is different among the original TSPM's landmarks, selecting the most salient landmarks is feasible. Technically, as shown in Figure 2.1. each landmark is denoted as a patch which is centered at the landmark with certain square size. Each such patch is with a weight matrix of the same size. On the landmark level, the weight matrices should be sparse; on the matrices' element level, the weight elements are all either non-zeros or near zeros. The sparse constraint is on the group level while inside each group the weights are not necessarily sparse. Such property is well characterized as group sparsity [114].

The sparse group constraint is defined as in (2.6). assume a partition $\cup_{j=1}^{m} G_j$ of β, which is rearranging the elements inside the vector β and grouping the neighboring elements as group G_j. The m groups are disjoint $G_1, G_2, ..., G_m : G_i \cap G_j = \varnothing$ when $i \neq j$. In this way, the coefficient vector becomes $\beta = [\beta_{G_1}, \beta_{G_2}, ..., \beta_{G_m}]$. We expect inside each group, only a subset $F \subset \{1, ..., m\}$ of those groups are non-zero elements while those non-zero elements are small.

$$\hat{\beta} = \arg\min_{\beta} \{ \frac{1}{n} \|X\beta - y\|_2^2 + \sum_{j=1}^{m} \|\beta_{G_j}\|_2 \} \tag{2.6}$$

Notice that the constraint on β is $L_{2,1}$ norm. At the inter-group level $G_j, j = 1, ..., m$, the constraint is L_1 regularized while at the intra-group level the coefficients are L_2 regularized which is considered dense but small. The $L_{2,1}$ constraint is considered the group sparse property.

2.3.3 Max-Margin Learning for Landmark Parameters

In our learning process, we collect positive samples from MultiPIE database [74], which contains annotations and viewpoint information, denoted as $\mathcal{C}_+$. Negative samples are collected from arbitrary natural scenes but without faces, denoted as $\mathcal{C}_-$. The overall training set is $\mathcal{C} = \mathcal{C}_+ \cup \mathcal{C}_-$. For each viewpoint i, we need to train the weights $\tilde{\mathbf{w}}_i$. For each landmark, we know that $\tilde{\mathbf{w}}_i = [\mathbf{w}_j^{iq}, \mathbf{w}_{jk}^{ig}]$, which is the weight vector consists of unary weights $\mathbf{w}_j^{iq}$ and pair-wise weights $\mathbf{w}_{jk}^{ig}$. The pair-wise weights are set according to the tree structure edge set E. $\mathbf{w}_{jk}^{ig}$ includes all the weights that are connected to node j. Similarly, in node k, there is such edge weight $\mathbf{w}_{kj}^{ig}$. They are not necessarily equivalent since parent node and child node may take each other in different importance. For simplicity, we denote $\tilde{\mathbf{w}}_i$ as $\tilde{\mathbf{w}}$ in the following notations. Based on (2.1), considering the group sparse constraint from section 2.3.2, we establish a max-margin framework in (2.7).

$$\begin{aligned} \arg\min_{\tilde{\mathbf{w}}, \varepsilon \geq 0} & \left(\sum_{n \in \mathcal{C}} \varepsilon_n + \lambda_1 \|\tilde{\mathbf{w}}\|_2^2 + \lambda_2 \sum_{t=1}^{m} \|\tilde{\mathbf{w}}_t\|_2 \right) \\ s.t. \forall n \in \mathcal{C}_+, & \left\langle \tilde{\mathbf{w}}, \tilde{\Phi}(I_n, \mathbf{s}_n) \right\rangle \geq 1 - \varepsilon_n \\ \forall n \in \mathcal{C}_-, \forall \mathbf{s}, & \left\langle \tilde{\mathbf{w}}, \tilde{\Phi}(I_n, \mathbf{s}) \right\rangle \leq -1 + \varepsilon_n \end{aligned} \tag{2.7}$$

where $\tilde{\mathbf{w}} = [\tilde{\mathbf{w}}_1, ..., \tilde{\mathbf{w}}_m]$. We omit the pose angle i here because the optimization is a unified framework to all the pose angles. Then the holistic weight vector $\tilde{w}$ is constructed by $\tilde{\mathbf{w}}_t$, each of which is a rearranged weight vector at part t combining both the unary weights and pair-wise weights. $\tilde{\Phi}$ is a feature descriptor, i.e. hog feature all through the optimization. The positive features are extracted over positive samples with ground truth and the negative features are extracted with arbitrary configurations.

To solve the problem, a group sparse optimization method is used. We refer the readers to [114] for details of algorithms. From the objective function 2.7, we know that:

$$\varepsilon_n \geq 1 - y_n \tilde{\mathbf{w}}^T \tilde{\Phi} \tag{2.8}$$

Minimizing objective function 2.7 pushes ε_n to $1 - y_n \tilde{\mathbf{w}}^T \tilde{\Phi}$, where y_n is the class label of node n. For simplicity, we denote $\tilde{\mathbf{w}}$ as W and $\tilde{\Phi}$ as Φ. We define the search process as:

$$S_{i+1} = W_{i+1} + \beta_i (W_{i+1} - W_i) \tag{2.9}$$

where sequence $\{W_i\}$ are the approximate solutions and $\{S_i\}$ are the search points. Learning rate β_i is a properly chosen coefficient. To compute W_{i+1}, an approximating model was proposed in [114] as (2.10). The loss function in (2.10) is defined in (2.11). Y is a matrix variable of the loss function $F_{L,W}(Y)$. In (2.10), W_{i+1} is achieved by minimize the loss function $F_{L,W}(Y)$ over Y.

$$\begin{aligned} F_{L,W}(Y) = &\left[loss(W) + \left\langle loss'(W), Y - W \right\rangle\right] \\ &+\lambda_2 \sum_{t=1}^{m} \|Y_t\|_2 + \frac{L}{2}\|Y - W\|_2^2 \end{aligned} \tag{2.10}$$

$$loss(W) = 1 - yW^T\Phi + \lambda_1 \|W\|_2^2 \tag{2.11}$$

Combining the above two equations, W_{i+1} is derived by minimizing the approximating model $F_{L,W}(Y)$ as shown in (2.12).

$$\begin{aligned} W_{i+1} &= \arg\min_Y F_{L_i,S_i}(Y) \\ &= \arg\min_Y \frac{1}{2}\|Y - S_i\|_2^2 + C\sum_{t=1}^{m} \|Y_t\|_2 \end{aligned} \tag{2.12}$$

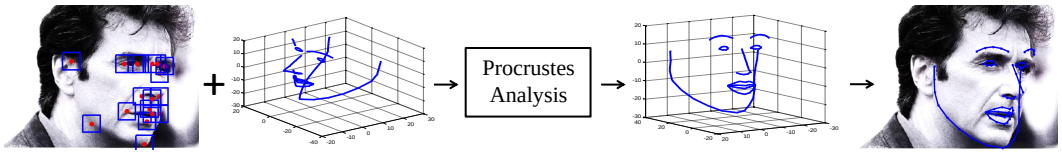


Figure 2.2: Pose-free facial landmark initialization using Procrustes analysis on 3D reference shape and detected optimized part mixture.

Further we notice that for each group in m groups, the optimization is independent, which is denoted as following.

$$W_{i+1,t} = \arg\min_{Y_t} \left(\frac{1}{2} \|Y_t - S_{i,t}\|_2^2 + C\|Y_t\|_2 \right) \tag{2.13}$$

The above objective function is convex and smooth, which can be solved by gradient descent methods.

Then we can select the most salient $\tilde{\mathbf{w}}_i$ to form a new tree. Those nodes with small weights are eliminated. Because the tree structure is changed, we have to re-train our weights. Training is achieved by solving the traditional max-margin problem:

$$\begin{aligned} &\arg\min_{\tilde{\mathbf{w}},\varepsilon\geq 0} \left(\sum_{n\in\mathcal{C}} \varepsilon_n + \lambda_1 \|\tilde{\mathbf{w}}\|_2^2 \right) \\ &s.t. \forall n \in \mathcal{C}_+, \left\langle \tilde{\mathbf{w}}, \tilde{\Phi}(I_n, \mathbf{s}_n) \right\rangle \geq 1 - \varepsilon_n \\ &\forall n \in \mathcal{C}_-, \forall \mathbf{s}, \left\langle \tilde{\mathbf{w}}, \tilde{\Phi}(I_n, \mathbf{s}) \right\rangle \leq -1 + \varepsilon_n \end{aligned} \tag{2.14}$$

which is a classic quadratic programming problem. We use the dual coordinate descent method proposed in [226] to obtain the optimized weights.

2.4 Cascaded Deformable Shape Fitting

With initial anchor points detection, we use general Procrustes analysis to project our 3D shape model onto the facial image. The 3D model is trained off line based on a 3D labeled face shape dataset [204] which serves as a prior all through our method. As head is a near-rigid object in 3D space, the 3D to 2D mapping is unique. The process is illustrated in Figure 2.2. In this section, we firstly formulate the problem into parametric forms. Assuming the aligning of neighborhood landmarks conditionally independent, we apply Bayesian inference to build

a probabilistic model. Further assuming the response map of each landmark patch mixture of Gaussian, we propose a two-step cascaded deformable shape model to refine the locations of landmarks.

2.4.1 Problem Formulation

In subsection 2.3.1, we have defined the landmarks as vector $\mathbf{s} = [s_1, ..., s_N]$, each landmark s_j is formed by concatenating the x and y coordinates. Let I denote the image potentially containing faces. The task is to infer $\mathbf{s}$ from I. Proposed by Coots et al. [40], ASM represents face shapes by a mean shape and a linear combination of k selected shape basis, $\mathbf{s} = \bar{\mathbf{s}} + \mathcal{Q}u$, where $\bar{\mathbf{s}}$ is the mean shape vector, $\mathcal{Q} = [Q_1, ..., Q_k]$ contains the k shape basis, $u \in \mathbb{R}^k$ is the coefficient vector.

The general Point Distribution Model (PDM) introduced by Cootes and Taylor [38], takes global transformation into consideration. Considering rigid transformation in 3D space, scaling, rotation and translation are the only 3 deterministic factors. Considering local deformation, the ASM shape basis is able to depict it as long as the training set contains enough variate shapes and the number of basis k is large enough. Hence we establish the relationship between any two points in 3D space in (2.15).

$$s_j = aR(\bar{s}_j + \mathcal{Q}u_j) + T \tag{2.15}$$

s_j is one of the defined landmarks, R is a rotation matrix, a is a scaling factor and T is the shift vector. The PDM provides us a way to depict arbitrary shape from a mean shape by deforming the parameter $\mathcal{P} = \{a, R, u, T\}$. The problem is to find such parameter $\mathcal{P}$ to map the 3D reference shape to a fitted shape which best depicts the faces in an image.

2.4.2 The Two-step Cascaded Model

We introduce a random variable vector $v = [v_1, ..., v_N]$ to indicate the likelihood of alignment, $v = 1$ means landmarks are well aligned and $v = 0$ means not. In this way, maximizing

$p(\mathbf{s}|v=1,I)$ demonstrates the aim that we are pursuing.

$$\mathbf{s}^* = \arg\max_{\mathbf{s}} p(\mathbf{s}|\{v_i=1\}_1^N, I) \tag{2.16}$$

$$\propto \arg\max_{\mathbf{s}} p(\mathbf{s}) p(\{v_i=1\}_{i=1}^n|\mathbf{s}, I) \tag{2.17}$$

$$= \arg\max_{\mathcal{P}} p(\mathcal{P}) \prod_{i=1}^{n} p(v_i=1|s_i, I) \tag{2.18}$$

Bayesian rule allows (2.16) being derived to (2.17). From (2.17) to (2.18), we assume that the degree of landmark i's alignment is independent to other landmarks' alignment given current landmarks' positions and the image. Since $\mathbf{s}$ is uniquely determined by parameter $\mathcal{P}$ given 3D shape model, $p(\mathcal{P}) = p(\mathbf{s})$.

We build a logistic regressor to represent the likelihood in (2.19).

$$p(v_i=1|s_i, I) = \frac{1}{1+exp\{\vartheta\varphi + b\}} \tag{2.19}$$

which has shown its effectiveness in [191]. φ is the feature descriptor of landmark patch i, ϑ and b are the regressor weights trained from collected positive and negative samples.

The parameter $\mathcal{P}$ are set from the PDM model which applies PCA to a set of registered shapes. The distance in PCA subspace is measured by Mahalanobis distance, which is a kernel l_2-norm measurement. Thus, we assume that the prior conforms to Gaussian distribution.

$$p(\mathcal{P}) \propto \mathcal{N}(\mu; \Lambda), \Lambda = diag\{[\lambda_1, ..., \lambda_k]\} \tag{2.20}$$

where λ_i is the i^{th} eigenvalue corresponding to the i^{th} shape basis in $\mathcal{Q}$ from the nonrigid PCA approach, μ is the mean parameter vector respectively.

Step 1: local patch mean-shift. Given a near-optimal landmark s_i, we intend to search its neighborhood to get the optimal alignment likelihood. Naturally the possible optimal candidates y_i form a region Ψ_i. We assume y_i conforms to Gaussian distribution $\mathcal{N}(s_i, \sigma_i\mathbf{I})$. Hence, the alignment likelihood is modeled as a mixture of Gaussian of the candidates y_i.

$$p(v_i=1|s_i, I) = \sum_{y_i\in\Psi_i} \pi_{y_i}\mathcal{N}(y_i, \sigma_i\mathbf{I}) \tag{2.21}$$

where $\pi_{y_i} = p(v_i=1|y_i, I)$. By Bayesian rule, $p(y_i|v_i, s_i, I) = \frac{p(v_i=1, y_i|s_i, I)}{p(v_i=1|s_i, I)}$, we obtain

$$p(y_i|v_i, s_i, I) = \beta_{y_i} = \frac{\pi_{y_i}\mathcal{N}(y_i, \sigma_i\mathbf{I})}{\sum_{z_i\in\Psi_i}\pi_{z_i}\mathcal{N}(z_i, \sigma_i\mathbf{I})} \tag{2.22}$$

An Expectation Maximization (EM) approach is raised to solve the problem of (2.18). Assuming all the landmarks' candidate distribution has the same deviation σ, we derive the objective function in (2.23).

$$\arg\min_{\mathcal{P},s_i} \left(||\mathcal{P}-\mu||^2_{\Lambda^{-1}} + \sum_{i=1}^{n} \sum_{y_i \in \Psi_i} \frac{\beta_{y_i}}{\sigma^2} ||s_i - y_i||^2 \right) \tag{2.23}$$

Taking the first order approximation $\mathbf{s} = \mathbf{s}^* + J\Delta\mathcal{P}$, $J = \frac{\partial \mathbf{s}}{\partial \mathcal{P}}$ the Jacobian of shape points, we obtain the updating function of parameter $\mathcal{P}$.

$$\Delta\mathcal{P} = \left(\sigma^2\Lambda^{-1} + J^T J\right)^{-1} \left[J^T U - \sigma^2\Lambda^{-1}(\mathcal{P}-\mu)\right] \tag{2.24}$$

In (2.24), $U = [U_1, ..., U_N], U_i = \sum_{y_i \in \Psi_i} \beta_{y_i} y_i - s_i$. Actually U is the mean-shift vector on response map Ψ. By iteratively updating the mean-shift vectors on each local patch response map, the parameter $\mathcal{P}$ is updated until converging to the global optimum.

Step 2: component-wise active contour. Local patch mean-shift performance relies heavily on the response map. We found in some cases merely mean-shift strategy cannot find the correct positions. Possibly the global constrain of $\mathcal{P}$ after mean-shift does not guarantee fitting each component exactly. But the result of mean-shift is expected to fall in the convergence basin of the global minima. We aim to take external force constrain to push the landmarks in each component aligning to its global minimum. It is component-wise because there is seldom such general external force for all the landmarks. By adding shape constrain similar as $\Phi^{ig}_{jk}(s_j, s_k) = (dx, dy, dx^2, dy^2)$ defined in section 2.3.1, we expect to preserve the structure of shape.

For each landmark, we evaluate its alignment by another measurement $exp(-\eta e_i)$. e_i is positive energy item including shape constrain, appearance constrain and external force constrain. Combining with objective function (2.18), we obtain a refined objective function as (2.25).

$$\arg\max_{\mathcal{P}} p(\mathcal{P}) \prod_{i=1}^{n} p(v_i = 1|s_i, I) \prod_{i=1}^{n} exp(-\eta e_i) \tag{2.25}$$

η is a regularization term. We take the linear combination of the three constraints as shown in

(2.26).

$$e_i = \gamma \begin{bmatrix} ds = [\Delta x \Delta y] \\ ds^2 = [x'' y''] \\ \nabla I \\ exp(-d) + log(1+d) \end{bmatrix} = \gamma \Gamma \mathbf{s} \tag{2.26}$$

where γ is the linear combination coefficients and d is a distance measure. We choose the Mahananobis distance of pixel value as d, which is the distance between the value of current landmark's pixel and the average value of face skin pixels. We notice that ∇I is the function of I and $\mathbf{s}$ while d is the function of I and $\mathbf{s}$ too. Once I is known, they are just the function of $\mathbf{s}$.

$$\Delta \mathcal{P} = \left(\sigma^2 \Lambda^{-1} + J^T J\right)^{-1} \cdot \left[J^T (U + \frac{1}{2}\eta\gamma\Gamma) - \sigma^2 \Lambda^{-1} (\mathcal{P} - \mu)\right] \tag{2.27}$$

Similarly we give out the overall rule for parameter update in (2.27), which can be achieved by gradient descent method. The reason not merging the two steps together is because in step 1, some patches' mean-shift may deviate due to low quality of response map before global shape constraint. If we directly raise the component-wise active contour on the deviated landmarks, the error may propagate. But if step 1's result is regularized by global shape constraint, the deviation is mediated and step 2 finds the convergence point with fewer iterations. Our bi-stage fitting procedure is summarized in Algorithm 1.

Algorithm 1 Two-stage deformable shape fitting with optimized part mixtures.

Require: facial image I.

Ensure: optimized $\mathcal{P}$.

1: **Initialization:** given trained $\tilde{\mathbf{w}}$, run landmark detection (9) to get $\mathbf{s}_0$.

2: run Procrust process on $\mathbf{s}_0$ and 3D shape model $\mathbf{s}_{3d}$ to obtain initial $\mathcal{P}$.

3: **repeat**

4: Local Patch Mean-shift: run $\mathcal{P}$ updating function (27), $\mathcal{P} \leftarrow \mathcal{P} + \Delta\mathcal{P}$

5: Component-wise Active Contour: run $\mathcal{P}$ updating function (30), $\mathcal{P} \leftarrow \mathcal{P} + \Delta\mathcal{P}$

6: **until** $\mathcal{P}$ converges

2.5 Experiments

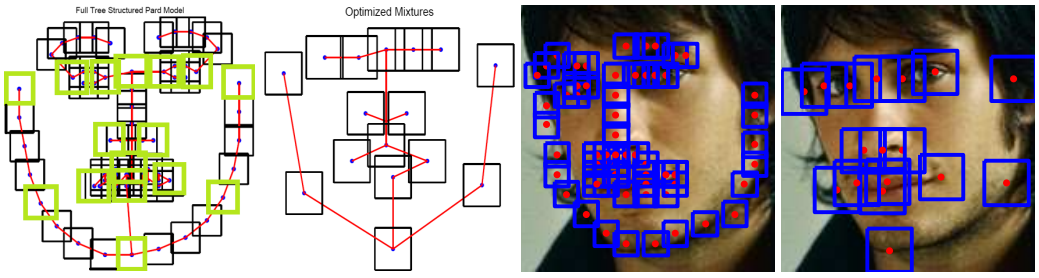
To evaluate our method, we introduce six main face databases used in our experiments, i.e. MultiPIE, AR, LFPW, LFW, AFW and iBug. They are collected either under specific experimental conditions or under natural conditions. All of them present challenges in different aspects.

MultiPIE [74] contains images of 337 people with different poses, illumination and expressions. We collected 1300 images from it, which include 13 different poses and each pose contains 100 images from different people. The training of optimized part mixtures is based on this database.

Images in AR [128] are frontal with different facial expressions, illumination and occlusion. We take 509 images of 126 people with different facial expressions to raise the experiment.

LFPW [14, 153], LFW [84] and AFW [226] are image databases collected in wild conditions. The images contain large variations in pose, illumination, expression and occlusion. For LFPW, we collected 801 training images and 222 testing images. For LFW, we selected 12007 of 13233 images which have valid annotations. For AFW, we collected 205 testing images. iBug [153] is a recently published even more challenging dataset in which the head pose variation and occlusion are the extreme conditions. The test dataset consists of 135 images with labeled ground truth.

As each of them has different number of annotation landmarks, when evaluating different algorithms on the same database, we use the landmarks from database annotation which are common in all the algorithms. We firstly verify the group sparse learning selection based landmark detectors by comparing to the Tree Structure Part Model (TSPM) [226] algorithm. We then conduct the near-frontal face alignment comparison with Multi-view ASMs [85], CLM [157], Oxford landmark detector [58], TSPM, Kernel Regression [150], SDM [198] and RCPR [25]. The databases are AR and near-frontal images from MultiPIE. In evaluating the pose robustness of our method, we introduced a more recent and challenging dataset iBug [153]. However, since not all the comparing methods can provided proper results on this dataset, i.e. ASM performs pool on the dataset, we do not present the comparison over all the method on



(a) TSPM Structure (b) OPM Structure (c) TSPM Sample Result (d) OPM Sample Result

Figure 2.3: Facial landmark models of TSPM and Optimized Part Mixtures. (a) TSPM landmark model with 68 red dots as landmark positions and blue rectangles as local patches. (b) The Optimized Part Mixture model with only 17 red-dot landmarks and blue rectangles as local patches.

this dataset as shown in Table 2.4. Based on LFPW, LFW and AFW, we compare the algorithms on the unconstrained cases. In addition, our method is potentially capable of tracking facial landmarks because of its fast update between two consecutive frames. We test it on talking face video [1] and compare it with CLM and Multi-ASM algorithms.

Quantitatively, the alignment error is measured by normalizing the absolute pixel error over the square root of face size, reflected by the rectangle hull of aligned landmarks. We uniformly apply the face size other than inter-ocular distance as the normalization measure because there are many cases, in which not both eyes are visible.

2.5.1 OPM vs. TSPM

Zhu and Ramanan [226] proposed a tree structure part model to simultaneously detect face and localize landmarks. The landmarks in their model are densely distributed. We propose a group sparse learning method to select the most representative landmarks. We conduct the comparison of the average localization error on AR and LFPW datasets. As the code provided by the authors is based on Matlab, we compare the running time on the same Matlab platform.

Figure 2.3 visualizes the TSPM dense model and our optimized mixture model. The TSPM consists of 68 points surrounded with 68 black bounding boxes. Each point is a node of the node set V. The neighboring points are connected in red lines. Each line is an element of the

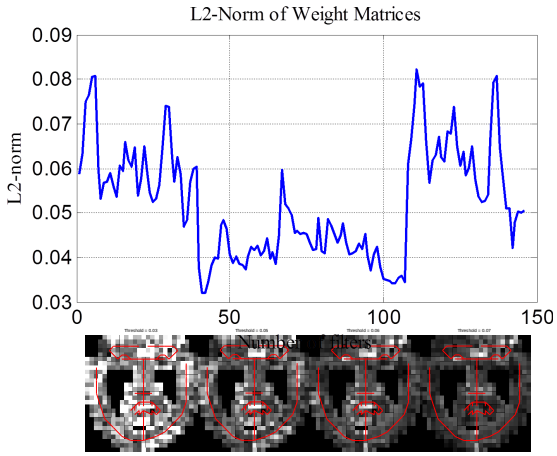


Figure 2.4: The visualization of weight vector norms and the gray scale patch image to show the weight distributions at various norm thresholds. The top part is the plot of weight vector norm of each filter. The bottom part are the gray scale patch images under norm threshold 0.03, 0.05, 0.06 and 0.07

line set E. The graph T = (V, E) consists of the node set V and the edge set E. The center is located at the center of the nose. It is expanded in an undirected and noncyclic way. Thus, the graph is a tree structure. In Figure 2.3 (b), the structure is a simplified one from Figure 2.3 (a). All the blue dots and surrounded black bounding boxes are the ones selected from the dense Tree Structure Part Model, also denoted as the green rectangles in Figure 2.3 (a). We could see that the OPM maintains a part of the original structure of TSPM but neglects the intermediate nodes in between. Though with fewer nodes, the OPM depicts a face completely.

From the TSPM structure, how much portion that the OPM should preserve is also an interesting point to investigate. In implementation, each landmark patch is related with several filters. Each filter corresponds to a weight vector for one view-point. Several view points at the same landmark may share a common filter. Different filters at the same landmark deal with different view-points. For all the filters, we firstly calculate the Euclidean norm of the weight vectors and plot the curve as shown at the top of Figure 2.4.

From Figure 2.4, we notice that for each of the peaks, the width of the peaks is significant, which means that the filters in the neighborhood all have significant or insignificant weights. Meanwhile, the filters with indices less than 50 and larger than 100 are more significant weighted than the filters from 50 to 100. It again verifies that the importance of different filters

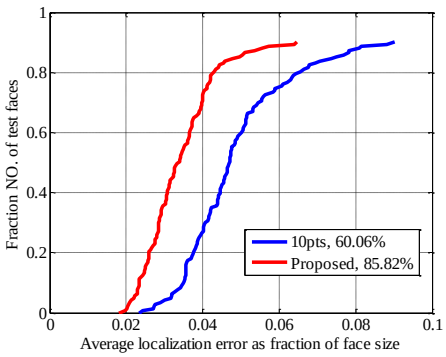
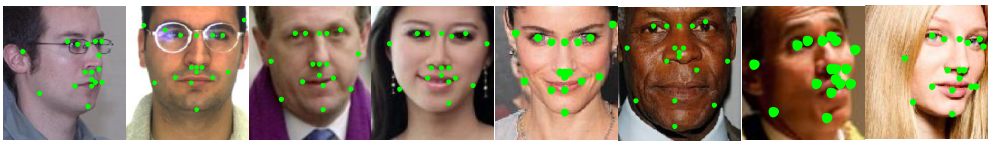


Figure 2.5: Cumulative error distribution curves on MultiPIE comparing proposed method with 10-point baseline method. The proportion reported in the legend is under the relative error 0.05.

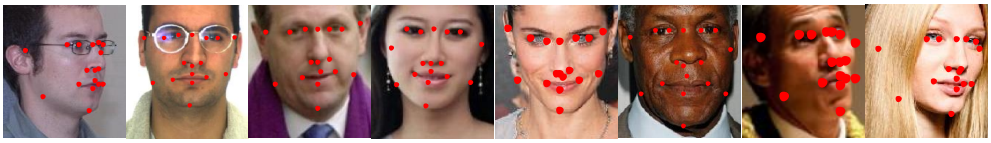
is different across all the landmarks. By setting threshold varying from 0.03 to 0.09, the weight vector distribution changes shown at the bottom of Figure 2.4. When threshold is low, most of the patches are selected. When threshold is high, only the most salient patches (shown in brighter blocks) are selected. We could visually find the most salient patches at the bottom of Figure 2.4 matches the simplified structure shown in Figure 2.3 (b).

An interesting and important argument is whether the group sparse selection is beneficial. From the visual saliency map in Figure 2.4, we manually selected the most salient points, i.e. four eye corner points, one nose tip point, one eyebrow center point, two mouth corner points, one upper lip and one lower lip points, in total 10-point setup as the baseline model. By independently train the baseline and our proposed 17-point model, we test the two methods on MultiPIE. As shown in Figure 2.5, the proposed optimized part mixture performs 25% proportion gap over the manual selection, which verifies the effectiveness of the group sparse selection.

We present more visual comparisons between the two algorithms across different databases as shown in Figure 2.6. In Figure 2.6, the result of TSPM (68 points) is manually selected to match with the 17 points' result of OPM, in which the definition of the 17 points in both TSPM and OPM is the same. We could see from the results that the proposed OPM provides more reasonable initial anchor points than TSPM, e.g. the red dots in Figure 2.6 distributes with less deviation and are more consistent than the green dots.



(a) TSPM selected anchor points in green dots to match the 17 point setup of OPM



(b) OPM detected anchor points in red dots

Figure 2.6: Visual comparison of converted TSPM with OPM. The converted TSPM is the manual selected 17 point setup which matches the 17 point setup in OPM. The results are evaluated on MultiPIE, AR, LFW, LFPW and AFW. The first column is from MultiPIE. The second column is from AR. The third and fourth columns are from LFW. The fifth and sixth columns are from LFPW and the last two columns are from AFW. (a) Result of converted TSPM in green dots as anchor points. (b) Result of OPM in red dots as anchor points.

Quantitatively, in Table 2.1, we observe that TSPM performs slightly better than the optimized mixture model on AR database, of which the gap under 5% relative error is only 2.81%. But the running time for proposed method is 2 times less. The performance gap is because the images in AR database are near-frontal with harmonic illumination conditions. The assumption of large pose variation and partial occlusion may not hold at this database. If without pose variation or occlusion, the landmark's detection error is expected to be small. In this situation, the more landmarks of detection evidence, the better of the detection result.

In contrast, the proposed method on LFPW is marginally better than TSPM and running time is 4 times less. The case for LFPW is that pose variation or partial occlusion exists. Each single landmark's detection error is expected to be enlarged. If the landmarks are dense, the error of each landmark influences its neighborhood more than the sparse structure, which is the case of Table 2.1 on LFPW. Our main purpose designing the group sparse structure on top of the original TSPM is to simplify the dense structure and speed up the process. In well-constrained environments, the TSPM performs better as shown in Table 2.1 AR database. While in unconstrained environments, the OPM shows some advantage on accuracy over TSPM

Table 2.1: Percentage of images less than given relative error level of TSPM and the proposed optimized mixtures on AR and LFPW datasets and average running time per image.

		$< 5\%$	$< 10\%$	$< 15\%$	time(s)
AR	TSPM	**69.4%**	**97.0%**	**99.3%**	14.03
	OPM	57.0%	85.4%	96.2%	**5.81**
LFPW	TSPM	71.8%	95.2%	97.7%	8.23
	OPM	**80.1%**	**96.1%**	**98.5%**	**2.25**

to overcome the error propagation. We cannot guarantee the OPM as an initialization can always provide sufficiently good result. Otherwise, there is no need for the following two-stage deformable fitting algorithm. On the other side, low-quality initialization such as gross failures obviously results in wrong localization. Thus, it is necessary for the OPM to provide reasonably accurate initialization. As indicating in (2.7), our goal is to minimize the localization error margin ϵ_n, which suggests that we require the detection to be as good as possible. From our experimental results, the initialization from OPM provides good enough accuracy such that the latter process achieves competitive while sometimes better results than other state-of-the-art methods. Therefore, our OPM effectively handles challenging situations in practice, and is an important module to improve the overall accuracy and efficiency.

2.5.2 Algorithm Component Analysis

In our framework, we have introduced Optimized Part Mixtures (OPM) to simplify TSPM's dense structure and initialize the landmarks. The two-step cascaded deformable shape fitting consists of local patch mean-shift and component-wise active contours. We investigate all modules of the framework to reveal the effectiveness of each component.

Since the TSPM's annotation is different from the OPM's, we manually select the landmarks from TSPM to match the landmark setup in our optimized part model for fair comparison, which denoted as TSPM-convert. For active contour, we directly compare the performance of our framework with and without such refinement. The experiment is conducted on LFPW and AFW wild face databases. We evaluate on the proportion of image volume when relative error

Table 2.2: Proportion of image volume less than given relative error level on LFPW and AFW comparing with TSPM-convert, the proposed method and the proposed method without component-wise active contour (No Snake).

		$< 5\%$	$< 10\%$	$< 15\%$
LFPW	TSPM-convert	71.8%	95.2%	97.7%
	proposed	**81.1%**	**96.1%**	**98.4%**
	No Snake	79.2%	94.3%	96.7%
AFW	TSPM-convert	60.4%	92.5%	97.9%
	proposed	**71.4%**	**95.8%**	**99.7%**
	No Snake	66.7%	93.7%	98.2%

is under 5%, 10% or 15%. Quantitative results are shown in Table 2.2.

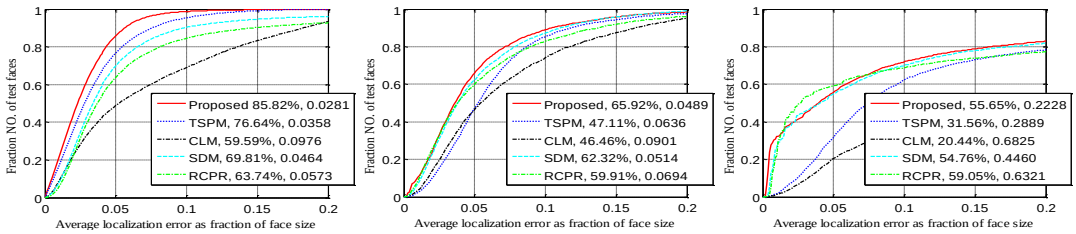
Using converted TSPM as initialization, we see the accuracy drops significantly from the proposed one. Thus applying group sparse selection of anchor points is a novel and key step in our framework. Comparing to the result without active contour, the proposed method is consistently and marginally better, which reveals Snake's effectiveness in improving performance.

2.5.3 Evaluation on Pose Robustness

In our work, a major task is to align the face shape under severe pose variation. We adopt the MultiPIE, selected images with large pose variation from LFPW (LFPW-P) and images with large pose variation from iBug (iBug-P) for the evaluation.

For LFPW-P, 112 images out of 215 test images are manually selected with significantly large head pose variation. The annotation is from 300-W and all the comparing methods are tested on the sub-dataset. Though the landmark setup for the comparing methods is different, the proportion of images well aligned against the normalized alignment error is a fair measure for all the methods.

For iBug-P, 81 images out of 135 test images are manually selected for testing. Most of the 135 test images are with large pose variation. However, since some of the images contain



(a) MultiPIE (b) LFPW-P (c) iBug-P

Figure 2.7: Cumulative error distribution curves for landmark localization on large pose variation databases. (a) Error distribution tested on MultiPIE. (b) Error distribution tested on LFPW-P. (c) Error distribution tested on iBug-P.

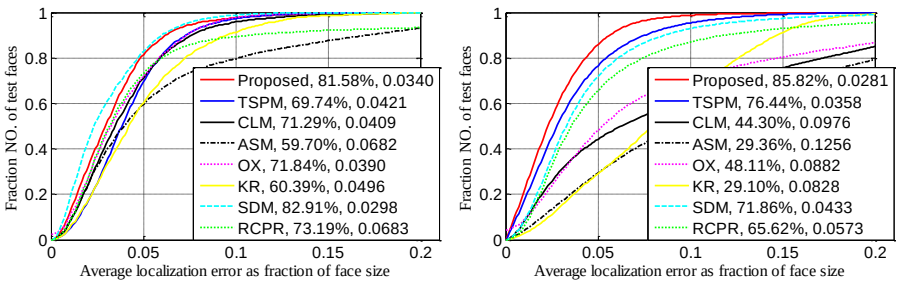
multiple faces, in which some comparing methods only predict one face shape, there is a mismatch to evaluate the alignment accuracy. Thus, we remove the images in this situation for fair comparisons.

The evaluation is conducted on the success rate of all the methods on the three databases and the curve of proportion of database volume vs. the normalized error. The success rate is shown in Table 2.3.

In MultiPIE, the TSPM and the proposed method achieves 100% detection rate on the test dataset while Viola-Jones is far less. It is because the detector almost all failed on the pose variations larger than 60 degree in MultiPIE. Even if the multi-view face detector succeeds in the large pose variation, those localization methods cannot facilitate to the large pose situations because their shape fitting schemes may fail in searching such large pose space. To validate this assessment, we equally provide all the compared methods the same face bounding boxes and evaluate their localization accuracy in Figure 2.7. The advantageous performance may

Table 2.3: The success rate of the detection, the proportion of successfully detected images over the database volume on MultiPIE, LFPW-P and iBug-P.

succ. rate	Viola-Jones	TSPM	Proposed
MultiPIE	0.54	1.0	**1.0**
LFPW-P	0.92	0.91	**0.94**
iBug-P	0.88	0.69	**0.91**



(a) AR (b) Near-frontal MultiPIE

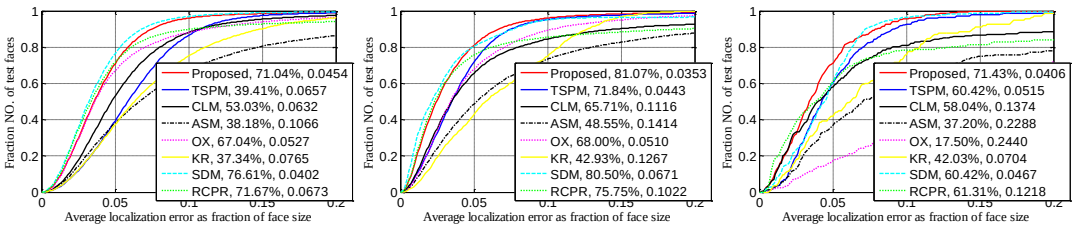
Figure 2.8: Cumulative error distribution curves for landmark localization on near-frontal images. (a) Error distribution tested on near-frontal AR database. The numbers in legend are the percentage of testing faces that have average error below 5% of the pupil distance. (b) Error distribution tested on near-frontal MultiPIE database. The percentage is the ratio of error less than 5% of ground truth face size.

result from that the TSPM and the proposed method simultaneously detect the key landmarks and the face region utilizing a multi-view deformable part model. The multi-view DPM breaks the entire pose search space into discrete subspaces. The propagation of part detection results enhance the overall success rate.

A more quantitative comparison is on the cumulative error function over the relative error as shown in Figure 2.7. The proposed method performs consistently better across the three large pose variation datasets. On MultiPIE, our method achieves 10% better than other state-of-the-art methods while on LFPW-P and iBug-P, the gap shrinks because the the faces in these two datasets are more challenging in head poses and all types of occlusion. Even so, our method manages to maintain high performance. Note that we equally provide all the compared methods with the same face bounding boxes. The difference shows only the capability of deforming the initial shapes to handle all kinds of pose variations. The results from Fig. 2.7 indicate that the proposed method is advantageous in dealing with pose variation.

2.5.4 Comparison with Previous Work

We compare our approach (optimized mixtures with cascaded deformable shape model) with the following methods. (1) Multi-view ASMs [85], (2) Constrained local model (CLM) [157],



(a) LFW (b) LFPW (c) AFW

Figure 2.9: Cumulative error distribution curves for landmark localization on face-in-the-wild databases. (a) Error distribution tested on Life Face in the Wild (LFW) dataset. (b) Error distribution tested on Labeled Face Parts in the Wild (LFPW). (c) Error distribution tested on Annotated Face in the Wild (AFW).

(3) Oxford facial landmark detector (OX) [58], (4) tree structure part model (TSPM) [226], (5) Kernel Regression (KR) [150], (6) Supervised Descent Method (SDM) [198] and (7) Robust Cascaded Pose Regression (RCPR) [25]. For wild faces, TSPM, RCPR and SDM has reported superior performance over many other state-of-the-art methods. For non-frontal comparison, we hard code ground truth face rectangle to Multi-ASMs, CLM and Oxford as face detection results because in those cases such methods may fail to locate faces merely using Viola-Jones detector.

We firstly evaluate performance on frontal and near-frontal faces in AR and MultiPIE database. For MultiPIE, we select the near-frontal portion of all the pose-variant images. The near-frontal is defined as faces with yaw angle varying from -45° to 45°, in which case all landmarks are visible. For the relative error (Figure 2.8 (a)), our proposed method achieves top performance except 1.4% gap below SDM. In Figure 2.8(b), the proposed method shows superior performance with significant margin to other methods.

Quantitatively, we evaluate all the algorithms on percentage of database volume against the normalized error shown in Table 2.4. In AR and MultiPIE, our method stands in the top two performance with only 0.01 alignment gap at error 0.05 compared to SDM. While compared to all other methods, the advantage in alignment accuracy is significant.

Further investigation is focused on the performance of all the methods on LFW, LFPW and AFW. Figure 2.9 shows that our method consistently outperforms other methods with a

(a) CLM

(b) TSPM full 1050 model

(c) proposed

Figure 2.10: Visual comparison of CLM, TSPM with full 1050 independent part model and our proposed method evaluated on MultiPIE, AR, LFW, LFPW and AFW databases. The first column is a test sample from MultiPIE. The second column is from AR database. The third and fourth columns are from LFW database. The fifth and sixth columns are with LFPW images and the last two columns are from AFW dataset. (a) Localization result by CLM. (b) Localization result by Tree Structure Part Model with full independent 1050 parts model which achieves the highest accuracy among all its models. (c) Localization result from proposed method.

significant margin. For fair comparison, we provide ideal face bounding boxes for compared methods , CLM, Multi-ASM and Oxford, as they may fail to detect faces in side-view face images. Although giving advantage to those methods, the proposed method achieves 71.0% of total face volume within relative error 5% on LFW, 81.1% fraction on LFPW and 71.4% on AFW, which consistently retains the localization accuracy in a very high level. Quantitative percentage results in Table 2.4 also supports the conclusion. One may notice that there is a small gap between SDM and our proposed method on LFW. One reason is that images in LFW are with less pose variations comparing to the other two wild face databases. The other thing is that SDM is trained based on LFW and MultiPIE. The smaller training and testing sample

Table 2.4: Proportion of images on AR, MultiPIE, LFW, LFPW and AFW comparing with Oxford detector (Ox), ASM, Kernel Regression (KR), CLM, TSPM, RCPR and SDM, at relative error level less than or equal to 5%, 10% and 15%, respectively.

Method	AR			MultiPIE			LFW			LFPW			AFW		
	5%	10%	15%	5%	10%	15%	5%	10%	15%	5%	10%	15%	5%	10%	15%
Ox	0.72	0.97	0.99	0.48	0.71	0.80	0.67	0.88	0.94	0.68	0.89	0.95	0.18	0.33	0.54
ASM	0.59	0.79	0.87	0.29	0.53	0.68	0.38	0.67	0.80	0.48	0.73	0.83	0.37	0.62	0.75
KR	0.60	0.91	0.98	0.29	0.64	0.91	0.37	0.75	0.90	0.43	0.75	0.96	0.42	0.76	0.91
CLM	0.71	0.95	0.99	0.44	0.63	0.76	0.53	0.88	0.95	0.66	0.85	0.90	0.58	0.81	0.87
TSPM	0.69	0.97	0.99	0.77	0.95	0.99	0.39	0.87	0.97	0.72	0.95	0.97	0.60	0.93	0.98
RCPR	0.73	0.89	0.92	0.66	0.87	0.93	0.71	0.89	0.93	0.76	0.85	0.88	0.61	0.79	0.81
SDM	**0.82**	**0.98**	**0.99**	0.72	0.93	0.97	**0.76**	**0.96**	**0.99**	0.80	0.95	0.96	0.60	0.96	0.98
Proposed	0.81	0.97	0.99	**0.85**	**0.98**	**0.99**	0.71	0.96	0.99	**0.81**	**0.96**	**0.98**	**0.71**	**0.96**	**0.99**

distribution gap within the same database may also result in the advantage.

From visualization point of view, we present some localization results of TSPM (full independent 1050 part model), CLM (implemented by the ourselves) and our proposed method in Figure 2.10. The CLM results pertain most of the landmarks in good positions. But for pose variation and subtle local variance, the output is not promising. TSPM can handle different kinds of head poses due to its multi-view model. However, its local shape constraint is too strong such that the holistic face shape is not precise. In contrast, the proposed method attempts to strike balance between the global shape constraint and the local shape constraint. Inside the bi-step procedure, the first step emphasizes on the global shape constraining while the second step aims at refining each facial component locally. By alternatively applying the two steps fitting, our method achieves state-of-the-art performance.

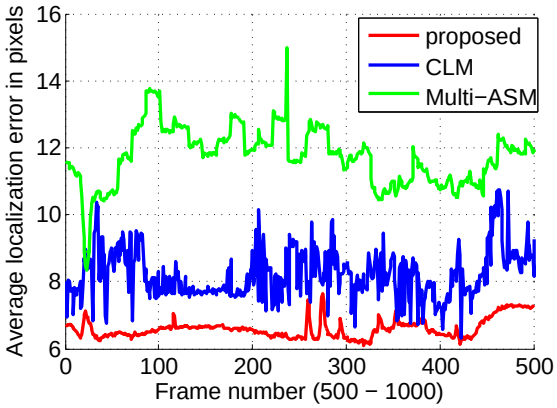


Figure 2.11: Average landmark tracking error in pixels of talking face video from frame 500 to frame 1000.

2.5.5 Evaluation on Talking Face Video

We claim that the proposed method (optimized mixtures with cascaded deformable shape model) has potential to track videos and image sequences. The reason is that in our model, initialization is simplified from TSPM which is claimed real-time detection performance and the two-step cascaded strategy is based on mean-shift and component-wise active contour. We can directly use information from past frames as the initialization for next frames.

Table 2.5: Percentage of talking face image frames less than given relative error level and Mean Average Pixel Error (MAPE) in pixels.

Relative error	$< 5\%$	$< 10\%$	$< 15\%$	MAPE
Multi-ASM	38.07%	73.72%	95.67%	12.22
CLM	73.16%	98.01%	99.80%	8.59
proposed	**79.19%**	**99.70%**	**99.98%**	**7.31**

Since TSPM is a detection based method without any plug-in of tracking strategy, we only compare the results on talking face video with CLM and Multi-ASM, which are able to raise video tracking. The relative error is defined as the fraction of average localization error over pupil distance. Table 2.5 shows that our method outperforms the other two methods with distinct margin. Visualization from Figure 2.11 convinces our conclusion that the error by proposed method is consistently smaller than the other two methods.

Computational complexity: Our algorithm consists of three parts. (1) Optimized part mixtures. Restricting the part model tree structure, a dynamic programming and distance transform strategy [63] is used in pursuing (2.5). It achieves $O(Nh)$ running time, where N is the number of landmarks and h is grid size defined in distance transform bounded by image size. (2) Local patch mean-shift. Assuming response map size ρ, the running time is $O(N\rho)$. (3) Component-wise active contour. The component number is a constant. For each component, we should evaluate ds, ds^2 and $exp(-d) + log(1 + d)$. Each takes $O(N)$. With k iterations, the running time is $O(Nk)$. Overall, our algorithm achieves $O(N(h + \rho + k)) \approx O(Nh)$, which is because in practice, $k \leq 3$ and $h \gg \rho$. Comparing TSPM running time $O(Nh)$, CLM with $O(N\rho)$ and ASM with $O(N\rho)$ (assuming the same patch size ρ for CLM and ASM), our algorithm is at the same running time level of those real-time methods. Typically, our N is a quarter of that in TSPM, which leads to the result that our method is at least two times faster than TSPM as shown in Table 2.1.

2.6 Summary

This chapter has presented a two-stage cascaded deformable shape fitting method for face landmark localization and tracking. By introducing 3D shape model with optimized part mixtures, we achieve pose-robust landmark initialization. The OPM has shown its advantage over TSPM not only in computation but also in localization accuracy. The sparse structure provides less computation volume and decreases the error propagation between neighboring landmarks during shape fitting. The combination of OPM and two-stage deformable shape fitting takes the advantages of both end: the OPM provides pose-aware initial landmark detection; the shape fitting constrains the holistic landmarks in facial shape and refines the local shape variance. The algorithm component analysis has shown the effectiveness of each module, each of which shows the improvement from previous stage with significant margin. Extensive experiments demonstrate the advantage of our method in aligning wild faces with large pose variation. It also outperforms CLM and Multi-ASM in face landmark tracking.

Chapter 3

Occlusion-robust Landmark Localization

In this chapter, we propose to overcome the difficulties of occlusion and initialization to improve the regression-based approach to facial feature localization. Our approach is based on the "consensus of experts" concept in machine learning. In our case, the "experts" are regressors that are each trained specifically to predict facial feature locations under the precondition that a particular region of the face is occluded. The occlusion region for each regressor is different from, yet overlapping with others. This enables a robust consensus to be formed using Bayesian inference. Note that regressor training requires no occlusion ground truth information because occlusion information is not used for each specific regressor. Once the landmark locations are determined, we employ a semi-supervised Gaussian MRF to smoothly propagate occlusion state labels from high-confident areas to the rest of the face. Finally, we propose a simple yet robust initialization strategy to compensate for the sensitivity of the regression-based approach to noisy initial detections.

3.1 Introduction

Early successes in facial feature localization, epitomized by the Active Shape Model(ASM) [40] and Active Appearance Model(AAM) [35, 131], are characterized by a parametric template that is fit to a given image by optimizing over the template's parameter space. Although effective for many cases, these parametric approaches tend to break down under extreme pose, lighting and expression, due to lack of flexibility in the representation.

Recently, regression-based methods [27, 198, 37, 44] have been shown to overcome some of these difficulties, and have achieved high accuracy, largely due to their greater flexibility as compared to parametric methods, as well as effective sub-pixel localization capability. Despite these successes, a major weakness of the regression-based approach is occlusion, which occurs

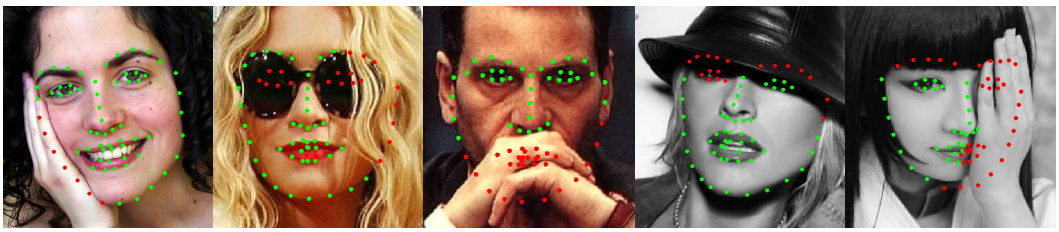


Figure 3.1: Sample visual results from Helen, LFPW and COFW databases. Landmarks estimated by proposed method with occlusion detection (red: occluded, green: non-occluded).

too often on faces in the wild (for example, in Figure 3.1). Regression depends heavily on local appearance to update feature location estimates. Occluded regions produce noisy features and result in erroneous location updates that not only affect the predicted locations of the occluded landmarks, but result in biased estimates of the visible landmarks as well.

The weakness of the regression-based approaches is the sensitivity to initialization. If the initial predicted feature locations are far from their true locations, the location updates predicted by the regressor are likely to be inaccurate, and likely to result in eventual drift away from the true locations. Most previous methods initialize with a scaled frontal face shape based on the detection box obtained from face detectors. However, face detectors are typically noisy in their output. Therefore robust initialization in the presence of noisy detections is a key factor to improve regression-based methods.

3.2 Related Work

The regression methods directly build up the relationship between the coordinate displacement and the feature difference. In the early decades, the regression is used within the parametric methods, which is to connect the parameters' update and the appearance difference [222]. Later, Valstar et al. [178] proposed support vector regression to predict the landmark positions from the local patch features. Cao et al. [27] designed a two-level cascaded regression framework. Between each two consecutive regression steps, a second level random fern regressors are selected to boost a strong regressor. Xiong and De la Torre [198] simplified the structure into a one-level cascaded linear regression. They proposed that the regression matrix could approximate the parametric update while sometimes more robust as the parametric update maybe

singular. Martinez et al. [129] argued that all the previous steps' regression results should also contribute to the current shape's update other than merely the current regression. Dantone et al. [46] imported the regression of random forest for landmark update.

The regression-based approach seems to produce the best accuracy in many instances among the competing methods. However, most methods suffer in the presence of occlusion, and regression methods suffer in particular since the iterative updates are driven largely by local appearance. To overcome this shortcoming, some researchers have proposed methods that in one way or another cope with the possibility of occlusion among the landmarks. For example, Roh et al. [151] used a large amount of facial feature detectors to provide over-sufficient landmark candidates and a RANSAC-based hypothesis and test method to robustly determine the whole shape. This method relies heavily on the facial feature detectors and is consequently computationally demanding. In [199], occlusion is modeled as a sparse outlier and the sparse constraint is applied during the optimization process. The sparse error could be from either occluded landmarks or perturbation of visible landmarks. Supervised occlusion detection methods are also proposed [190, 208]. However, if a particular occlusion case is missing in the training set, these methods may fail. A recent work on face alignment with occlusion [25] attempts to use regression to predict the occlusion likelihood of landmarks. They divide the facial area into 3 by 3 blocks and use one non-occluded block per time to predict the landmark positions. The approach shows its positive effects but the statistical prior of each block's occlusion condition is fixed. In contrast to [25] which considers one block of non-occluded features each time, our approach attempts to use all the features from the non-occluded regions. Though there is no occlusion prior, the proposed method applies Bayesian consensus over all regressors to recover the occlusion.

3.3 Occlusion-robust Localization

Linear regression has proven its effectiveness in facial landmark localization [27, 198]. In order to tackle occlusion, we design a set of regressors which are designed specific to different occlusion conditions. For instance, a right eye regressor extracts features over all the landmarks except the landmarks of right eye, which we denote as an occlusion-specific regressor. Then a

Bayesian inference framework is introduced to predict the landmark positions by jointly considering all the regressor outputs and evidence of low-level appearance models. Encouraged by the regression results, we further apply SVM and Gaussian MRF regularization to identify the occluded landmarks.

3.3.1 Occlusion-specific Regressors

As defined in [198], feature based objective functions i.e. (4.1) are practical because at some occasions appearance is easy to extract while ground truth landmark positions are absent.

$$\arg\min_{\Delta s} \|h(I(s_0 + \Delta s)) - \Phi^*\|_2^2 \tag{3.1}$$

We define $F(s_0 + \Delta s) = \|h(I(s_0 + \Delta s)) - \Phi^*\|_2^2$. h is the feature extraction, which is SIFT feature here. Given the shape configuration s consisting of n landmarks, I(s) is defined as the set of image patches which are sampled around the current n landmarks. s is a shape configuration denoted as $s = [x_1, x_2, , x_n]$, consisting of n landmarks' coordinates. s_0 is an initialized shape configuration in (3.2) and Δs is the shape displacement which is the gap between the initialized shape configuration and the optimal shape positions. Φ^* is the reference feature either from ground truth or from training template. The objective is to find the coordinate displacement Δs such that the feature best matches reference feature in $l2$-norm,which further denoted in (3.2).

$$\Delta s = -2H^{-1}J^T(\Phi_0 - \Phi^*) \tag{3.2}$$

where H is the Hessian of function F and J is the Jacobian of feature extraction function h. From (3.2), there is a linear relationship between the coordinate displacement Δs and feature difference $\Phi_0 - \Phi^*$. In [198], they proposed a linear regression based framework to model the relationship as shown in (4.2).

$$\Delta s = a\Phi_0 + b, R = (a, b) \tag{3.3}$$

a is a regression matrix to approximate $-2H^{-1}J^T$ especially when H is singular. The ground truth feature Φ^* is represented by an intercept item b, which is advantageous for inference as there is no ground truth when test case appears. Φ_0 is a feature vector concatenated by n feature vectors extracted at each fiducial point, which indicates that each fiducial point's displacement is related with all other fiducial points' appearance.

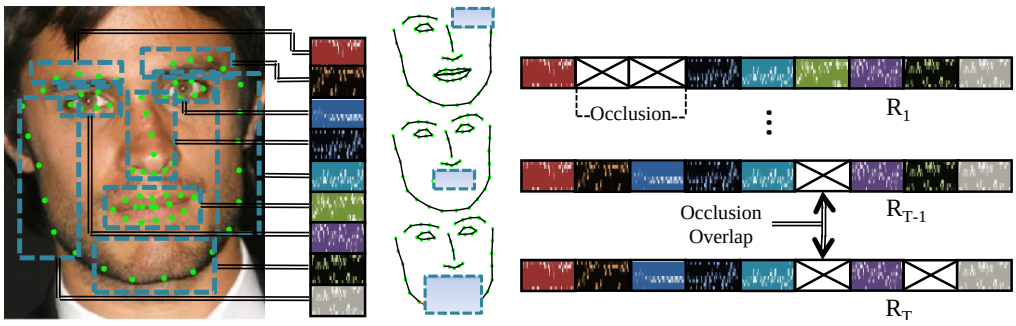


Figure 3.2: Illustration of occlusion-specific regressors. Color blocks are regression weights for different components, i.e. left profile, mouth, etc. For different occlusion states, i.e. right eyebrow and right eye occlusion, the regressors are designed not to use the features from occluded region. Those occlusion states are defined to have occlusion overlap with each other, e.g. mouth occlusion and mouth chin occlusion have overlap of mouth occlusion.

Based on this observation, we propose to train an ensemble of regressors, each of which handles one type of occlusions. The occlusions are combinations of different facial components, i.e. eyebrow, nose, left profile etc. The illustration is shown in Figure 3.2. The training is almost the same as supervised descent method (SDM) [198]. The difference is that here we only extract features at those non-occluded landmarks, i.e. for training the mouth occlusion regressor, we only extract features at non-mouth landmarks. For robustness, the layouts of landmarks between different regressors overlap with each other. In this way, it is expected to be more than one regression result approaching optimal solution, which provides potential to conduct consensus of regressors.

Suppose there are T such regressors. We define those T regressors as right-eyebrow-eye, right-eyebrow, right-eye, right-contour, left-eyebrow-eye, left-eyebrow, left-eye, left-contour, chin, both-eyebrow, all-contour, both-eyes, chin-mouth, nose-mouth and mouth respectively as shown in Figure 3.3. All of them are visually different because they are designed for different occlusions. In the training part, the goal is to minimize the regression error over all the training faces and all initialized landmark positions $s_t^k, t = 1, ..., T, k = 1, ..., K$. The superscript k means the k^{th} iteration of regressor R_t in the training. Advantageous to other occlusion detection methods, our method needs no occlusion information because the occlusion-specific

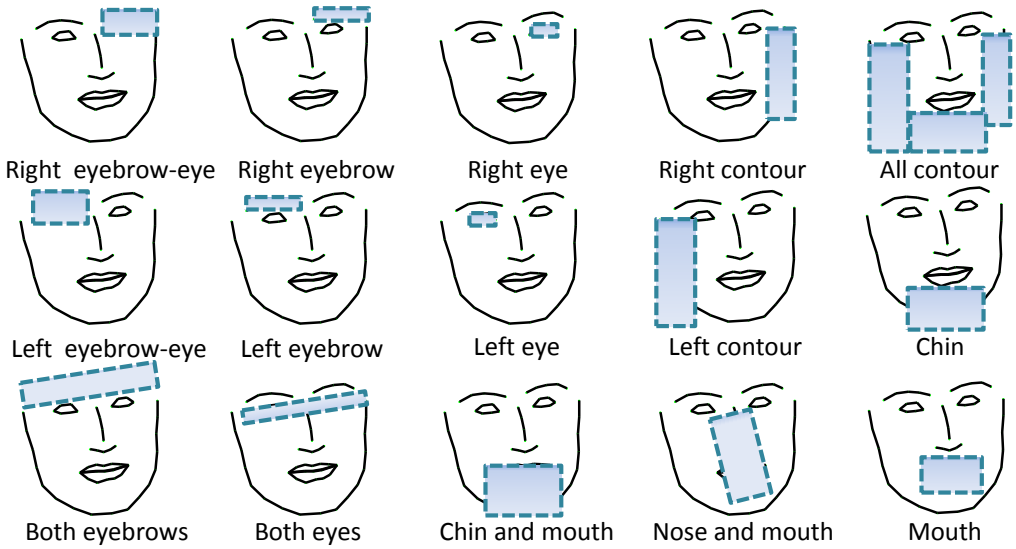


Figure 3.3: The illustration of 15 specific occlusion definitions.

regressors do not take the occlusion part into consideration. For instance, to train left-eye occlusion regressor, based on general non-occluded facial images, we only consider the features from all other areas except left eye, no matter whether the left eye is occluded or not. Thus general face image database is sufficient for the training of our method. As in SDM, practically four to five steps of linear regression steps are needed to reach the convergence. We learn T regressors $R_1, ..., R_T$, each of which consists of K cascaded single regressor $R_t = \{R_t^1, ..., R_t^K\}$.

3.3.2 Consensus of Occlusion-specific Regressions

Given the multiple landmark predictions resulting from the T cascaded regressors, it is necessary to select which of these is uncorrupted by occluded features, and thereby determine the optimal landmark positions. The response map provides the local appearance of a landmark neighborhood. Given the T regressors' observations, updating the landmark estimations by aggregating the T observations weighted by their likelihood is a more robust method. It is because the likelihood indicates how similar or close it is to the optimal position which is provided by the response map. If only applying geometric mean or weighted mean by the geometric distance, it is sensitive to the observation outliers. Moreover, the geometric mean does not necessarily approach the global optima if the number of observations is not sufficiently large.

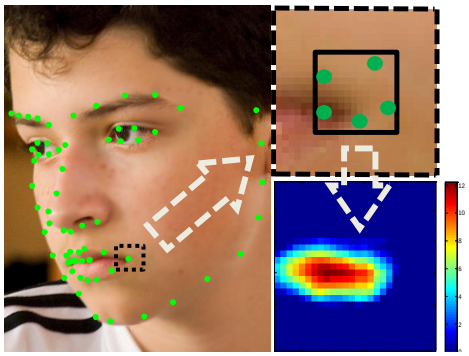


Figure 3.4: Illustration of response map.

Based on the response maps $\mathcal{M} = \{M_j\}, j = 1...n$, we propose a Bayesian inference framework to conduct a local exhaustive search. A response map is defined as: inside the neighborhood bounding box of the landmark, how likely each neighborhood pixel is the true position. We assume a linear regression model for each pixel to predict the likelihood, which is defined in (3.4). where κ and τ are the coefficients in the linear regression, which could be efficiently achieved by SVM. x is the current landmark position. Given a patch which is centered at the observations $\hat{x}_j^i, j = 1,,N, i = 1,,M$ (N is the number of landmarks, M is the number of regressors), each pixel inside the patch is evaluated using the linear regression model above. After all the pixels' likelihood is computed, a response map with the same size of the patch is formed, in which the elements are the pixels' likelihood. ϕ stands for Histogram of Oriented Gradient (HOG) feature. The generation of response map M_j for a landmark is illustrated in Figure 3.4. Firstly, a local region is cropped out as shown in Figure 3.4, which is formed by bounding the estimated points (denoted as green dots) from all the regressors. For each point inside the local region, its likelihood of being the true landmark is evaluated by SVM trained off-line. After all points are calculated, the response map is formed as in Figure 3.4.

$$p(y|x) = \kappa\phi(x) + \tau \tag{3.4}$$

Given the response maps, our objective function can be probabilistically formulated as (3.5).

$$\arg\max_s p(s|\hat{s}_1, \hat{s}_2, ..., \hat{s}_T, \mathcal{M}) \tag{3.5}$$

where $\hat{s}_1, \hat{s}_2, ..., \hat{s}_T$ denote the shape predictions from the T regressors.

To handle occlusion, we introduce $v_i, i = 1, ..., T$, which is a binary variable that is true if regressor R_i's landmarks are non-occluded. Thus, the probability that the regression result $\hat{s}_i$ approximates the true position can be represented as $p(v_i = 1|\hat{s}_1, \hat{s}_2, ..., \hat{s}_T)$. Originally, if there are sufficient such regressors, weighted mean is a straight forward way to estimate the optimum. But this naive method ignores the cue from the response maps. Our Bayesian framework takes the response maps into consideration by computing $p(s|v_i = 1, \mathcal{M})$. Consequently, (3.5) can be rewritten as:

$$p(s|\hat{s}_1, \hat{s}_2, ..., \hat{s}_t, \mathcal{M}) = \sum_{i=1}^{T} p(s|v_i = 1, \mathcal{M}) p(v_i = 1|\hat{s}_1, \hat{s}_2, ..., \hat{s}_T) \tag{3.6}$$

where the second term models the deviation of regressor R_i's output from the majority, which can be expressed as:

$$p(v_i|\hat{s}_1, \hat{s}_2, ..., \hat{s}_T) = exp(-\eta \|\hat{s}_i - \dot{s}\|_2^2) \tag{3.7}$$

In the above model, we define $\dot{s}$ as the approximate of optimal position, which is obtained by an iterative outlier removal and averaging algorithm based on the T observations $\hat{s}_1, ..., \hat{s}_T$. The goal is to compute a robust mean while excluding the effect of outliers caused by non-compatible regressors. One may argue that since the overlapped occluded regions between different occlusion-specific regressors are very few, the majority of estimate will deviate from the true estimation. In our implementation, in order to obtain sufficient observations, for each occlusion-specific regressor, we provide a number of different initializations. Such initializations are slightly disturbed from the mean initialization according to Gaussian distribution. If the current occlusion-specific regressor well predicts the occlusion pattern, those different initializations should converge to the same optimum. Although those not-matching regressors are the majority, their diverged results do not form the localization majority at least in the occlusion region. Since their initializations are Gaussian disturbed, the linear regression from the not-matching regressors do not provide the converged majority localization result.

For those not occluded landmarks, the observation majority will provide the approximate of the true positions. For those occluded landmarks, as above discussed, the deviated results are treated as outliers while the results from the regressors which well predict the occlusion

pattern are the majority localization. To evaluate the outliers, we apply an online clustering algorithm to dynamically remove the regression result, which contributes heavily in increasing the intra-class covariance. After removing the outliers, a geometric mean of all the majority positions is computed as the approximate of optima.

Given conditional independence assumption of individual landmarks, the shape alignment probability (the first term) in the objective function can be modeled as:

$$p(s|v_i, \mathcal{M}) = \prod_{j=1}^{n} p(x_j|\hat{x}_j^i, \mathcal{M}) \tag{3.8}$$

where $s = (x_1, x_2, ..., x_n)$ and $\hat{s}_i = (\hat{x}_1^i, \hat{x}_2^i, ..., \hat{x}_n^i)$, x_j denotes a landmark prediction and $\hat{x}_j^i$ denotes a landmark observation.

Each landmark's alignment probability can be modeled as a response map update problem:

$$p(x_j|\hat{x}_j^i, \mathcal{M}) = \sum_{y \in \phi \subset \mathcal{M}} p(x_j|y) p(y|\hat{x}_j^i) \tag{3.9}$$

Given the current estimate $\hat{x}_j^i$, we consider all the neighboring points y which forms neighborhood $\phi \subset \mathcal{M}$ of $\hat{x}_j^i$ to indicate the alignment likelihood of the next update position x_j. The posterior $p(x_j|y)$ is assumed Gaussian distribution $p(x_j|y) \sim N(x_j; y, \sigma_j I)$.

The probability map $p(y|\hat{x}_j^i)$ is obtained from the response map which is learned from SVM on training data. Consequently, the response map update can be achieved by fitting a Mixture of Gaussian (MoG) model:

$$p(x_j|\hat{x}_j^i, \mathcal{M}) = \sum_{y \in \phi \subset \mathcal{M}} \gamma_y^i N(x_j; y, \sigma_j I) \tag{3.10}$$

where $\gamma_y^i = p(y|\hat{x}_j^i)$. The overall objective function now becomes:

$$\arg\max_{s} \sum_{i=1}^{T} p(v_i|\hat{s}_1, \hat{s}_2, ..., \hat{s}_T) \prod_{j=1}^{n} \sum_{y \in \phi \subset \mathcal{M}} \gamma_y^i N(x_j; y, \sigma_j I) \tag{3.11}$$

For optimization, we take an alternating scheme: fixing $p(x_k|\hat{x}_k^i, \mathcal{M})$ for all landmarks $k \neq j$, and optimize for the j^{th} landmark via the Expectation Maximization (EM) algorithm. We can iterate this alternating process multiple times until convergence.

3.4 Occlusion Inference

Compared to fully visible facial images, occluded faces are with one or several facial parts that are sheltered by obstacles. As we know, occlusion of landmarks is highly pose-dependent.

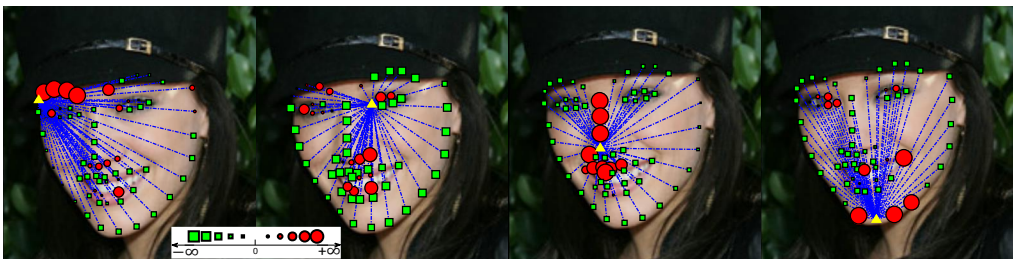


Figure 3.5: Visualization of weights for label propagation. The size of a landmark is proportional to its weight. Yellow triangle is the central landmark being processed. Red landmarks are with positive weights which are similar to the central landmark while green landmarks are with negative weights which are dissimilar to the central landmark.

The same landmark with different head poses may have different appearance. The head pose can be inferred by Procrustes Analysis over the predicted landmarks and the 3D reference face shape [206]. Then our inference work starts with classifying each landmark as occluded or non-occluded under different poses. By extracting pyramid SIFT descriptor $h(x)$, a standard linear SVM framework is applied to provide the detection score, $f(h(x)) = \omega^T h(x) + \beta$. In the training part, well-aligned landmark appearance and occluded appearance are collected with respect to three head poses, left head pose $(-45°, -15°)$, near-frontal head pose $(-15°, 15°)$ and right head pose $(15°, 45°)$. The testing examines the head pose first and apply the pose-dependent occlusion classifier.

Usually the classification might be sensitive and not consistent among landmarks. But we can obtain some detections with high confidence. These highly confident detections are labeled with occlusion state labels. We use a graph-based method to jointly infer the occlusion status for all the landmarks. Motivated by the work from Zhu et al. [225], assuming there are m labeled points $x_1, ..., x_m$, and $n - m$ unlabeled points $x_{m+1}, ..., x_n$, which constitutes the node set V. All those points are fully connected, which forms the edge set E. The weights between edges are defined by (3.12).

$$w_{ij} = exp\left(-\|x_i - x_j\|^2_{\Sigma^{-1}_{d,i,j}} - \lambda\|h(x_i) - h(x_j)\|^2_{\Sigma^{-1}_h}\right) \tag{3.12}$$

The first term in the exponential represents the spatial distance, Σ_d is the covariance matrix among different nodes. The second term measures the similarity of feature vectors, h denotes the feature extractor and Σ_h is the covariance matrix among all the features. λ is a balancing

factor between the two terms. The similarity between different landmarks is visualized in Figure 3.5.

Given such graph $G = (V, E)$, with the edges defined by the weights w_{ij}, the task becomes a label propagation problem on Graph G. The classifying function f maps features into category label $(0, 1)$. Such function f is not good enough to depict all the points' category. We search another function $\tilde{f}$ to minimize the weighted score error. The boundary conditions must satisfy f at the m given labels which assume the joint probability of the graph nodes a Gaussian distribution. The objective function is depicted in (3.13).

$$\begin{aligned} &\arg\min_{\tilde{f}} \sum_{i\sim j} w_{ij} \left(\tilde{f}(i) - \tilde{f}(j)\right)^2 \\ &s.t. \qquad \tilde{f}|_{i=1,\ldots,m} = f \\ &\qquad\qquad \Delta\tilde{f} = 0 \end{aligned} \tag{3.13}$$

Denoting $D = diag(d_i), d_i = \sum_j w_{ij}$, $W = [w_{ij}]$ and $\Delta = D - W$, separating W into four blocks, $W = [W_{m,m}, W_{m,n-m}; W_{n-m,m}, W_{n-m,n-m}]$, the top left of which corresponds to the labeled weights $W_{m,m}$, the unlabeled landmarks are computed as in (3.14). More theoretical representations and explanations can be found in [225].

$$\tilde{f}_{n-m} = (D_{n-m,n-m} - W_{n-m,n-m})^{-1} W_{n-m,m} \tilde{f}_m \tag{3.14}$$

3.5 Robust Initialization with Max-margin Learning

Regression based methods are quite sensitive to initial positions because initialization deviated from the search space of regression leads to undesirable predictions. For example, face detection may generate drifted face bounding boxes, faces are in-plane rotated, etc. Usually, faces in the unconstrained environments are not up-frontal, which increases the complexity of the shape searching space. If we could detect the face shape variation ahead, i.e. the in-plane rotation of the face, the initialization of landmarks could be consistent across all the test cases.

Re-initialization by re-projecting the initial landmarks [25] alleviates the problem but is very restricted. Because the re-projection may not locate initial landmarks close to the regression search space with any guarantee. We present a simple but very effective algorithm to generate good initializations. Our goal is to overcome the scaling, translation and roll variation

Algorithm 2 Robust initialization using max-margin learning on dense sift feature.

Input: Facial Image I given a rough bounding box $B = [x_l, y_l, x_r, y_r]$, reference shape $\bar{s}$.

Output: Initialized shape s_0, affine parameters, scaling $\mathcal{A}$, translation $\mathcal{T}$ and in-plane rotation $\mathcal{R}$.

Extract dense SIFT descriptors, their coordinates $d = [x_1, y_1, ..., x_r, y_r]$ on image I restricted within B.

Calculate centroid of d as $C = Centroid(d)$, $\mathcal{T} = C - Centroid(B)$.

Denote bounding box operation as $\mathcal{B}$, $\mathcal{A} = \mathcal{B}(d)/\mathcal{B}(\bar{s})$.

Training of roll angle θ detection:

SIFT feature extraction: centered at C, patch scale as $\mathcal{B}(d)$.

Positive samples $\mathcal{C}_+$: SIFT feature with orientation of one of the angle intervals: $[-45°, -30°]$, $[-30°, -15°]$, $[-15°, 0°]$, $[0°, 15°]$, $[15°, 30°]$, $[30°, 45°]$.

Negative samples $\mathcal{C}_-$: SIFT feature with orientations other than the selected angle interval in the positive examples.

$\arg\min_{\psi,\epsilon} \left(\sum_{u \in \mathcal{C}} \epsilon_u + \|\psi\|_2^2 \right)$, $s.t. \forall u \in \mathcal{C}_+, \langle \psi, u \rangle \geq 1 - \epsilon_u, \forall u \in \mathcal{C}_-, \langle \psi, u \rangle \leq -1 + \epsilon_u$

Roll angle θ detection:

SIFT feature extraction $(u_1, ..., u_7)$: centered at C, patch scale as $\mathcal{B}(d)$, orientation varies every $15°$ from $-45°$ to $45°$.

$\arg\max_{u_i} \langle \psi, u_i \rangle \rightarrow \theta$, $\mathcal{R} = [\cos\theta, \sin\theta; -\sin\theta, \cos\theta]$.

$s_0 = \mathcal{A}\mathcal{R}\bar{s} + \mathcal{T}$

(in-plane rotation) problems. As described in Algorithm 2, the detection training of roll angle θ is an off-line stage, in which we use a max-margin learning framework to discriminate the positive samples $\mathcal{C}_+$ from the negative samples $\mathcal{C}_-$. ϵ and ψ are the margin and weights for training. $\mathcal{C} = \mathcal{C}_+ \bigcup \mathcal{C}_-$ is the training set. $\bar{s}$ is a 3D pre-calculated frontal reference shape. For notation consistency, we list the training part in the algorithm after translation and scaling calculation.

There are two other factors, pitch and yaw angles, to encode arbitrary head pose. These factors are represented in the training data sufficiently and the multi-step regressions are able to search for the variations of pitch and yaw angles.

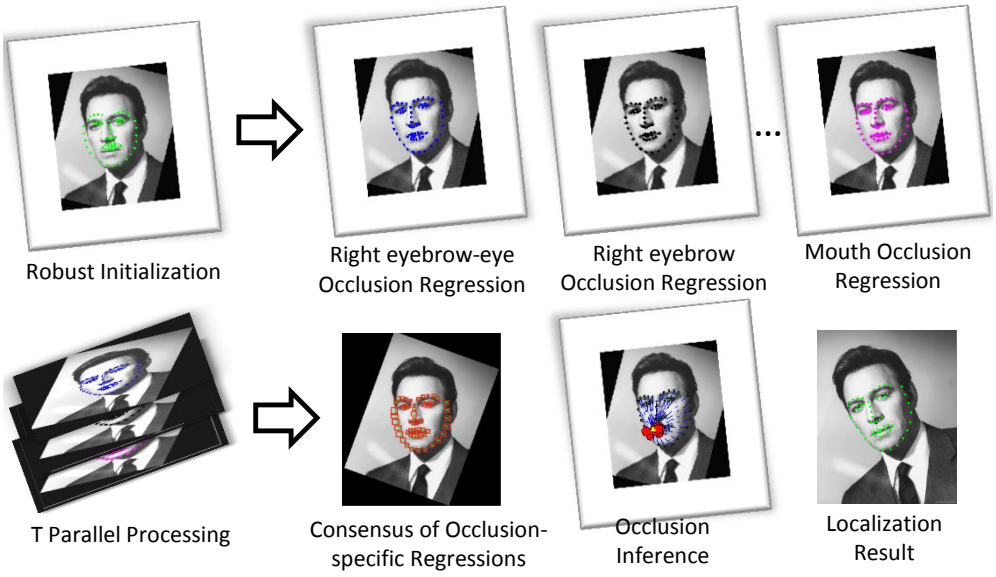


Figure 3.6: The flowchart of the proposed framework consisting of robust initialization, parallel occlusion-specific regressions, consensus of regressions and explicit occlusion detection.

3.6 Proposed Framework and Computational Complexity

In summary, our proposed method consists of robust initialization, occlusion-specific regressors, a robust consensus method and an explicit occlusion detection strategy. The overall flowchart is shown in Figure 3.6.

The robust initialization automatically detects the affine transformation parameters and provide a near up-frontal facial image for further processing. Then the T occlusion-specific regressors align the face independently. Given all the alignment results, we form local patch for each landmark and apply our proposed consensus of regressions method to predict the final landmarks. Based on the localization result, we build a graph connecting all the points with our defined similarity measure. Assuming Gaussian MRF on the graph, a joint label propagation scheme is conducted over all the landmarks.

Complexity of each component of our method is as follows: 1) Occlusion-specific regressions, takes $O(Kn^2)$ which is the same as SDM when parallelized, where n is number of landmarks and K is the number of cascaded regression iterations. 2) Assuming the local patch size q and the number of iteration C, the complexity for consensus of regression for all the

n landmarks is $O(Cnq^2)$. In our implementation, value for C and q are 3 and 20, respectively. 3) Occlusion detection, SVM detection takes $O(n)$, close-form label propagation takes $O(n^2)$. The overall time complexity is $O(n^2)$. Merging all the steps, we achieve the same time complexity as SDM.

3.7 Experiments

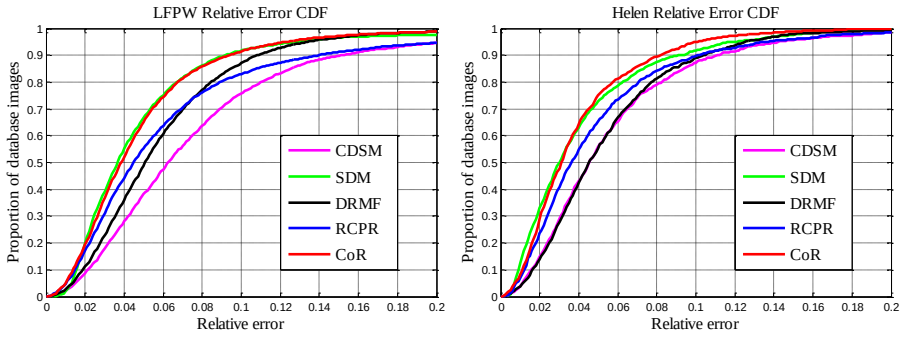
Our method is mainly focused on facial landmark localization under both non-occluded and occluded conditions. We evaluate our method on two challenging benchmarks, Labeled Facial Parts in the Wild (LFPW) [14] and Helen facial feature database [105]. Moreover, we select occluded images from both LFPW and Helen databases, denoted as LFPW-O and Helen-O. Together with Caltech Occluded Faces in the Wild (COFW) [25], we evaluate our method on the three occlusion datasets and compare with several state-of-the-art algorithms. We also evaluate the occlusion detection performance on COFW and compare it to [25].

3.7.1 Experimental Setup

In the experiments, we use the 66 points annotation from 300 Faces in-the-Wild challenge [153] for both training and testing, omitting two inner mouth corner points. The annotation is consistent across different databases, e.g. LFPW and Helen. Since COFW uses the 29 points annotation same as the original annotation of LFPW, when evaluating on COFW, we use the overlapped 19 points which are defined by both 66 points annotation and 29 points annotation.

LFPW consists of face images under wild conditions. The images vary significantly in pose, illumination and occlusion. There are 811 training images and 224 testing images in this database. We selected all occluded images, which is 112 out of 224 testing images to form LFPW-O. Helen is another wild face database, consisting of faces under all kinds of natural conditions, both indoor and outdoor. Most of the images are of high resolution. The training set contains 2000 images and testing set contains 330 images. We randomly selected 290 occluded face images out of 2330 images to form Helen-O. In the training of our regressors, we select 402 Helen training images which are not included in Helen-O and 468 LFPW training images.

We compare our method Consensus of Regressors (*CoR*) with 4 state-of-the-art methods,



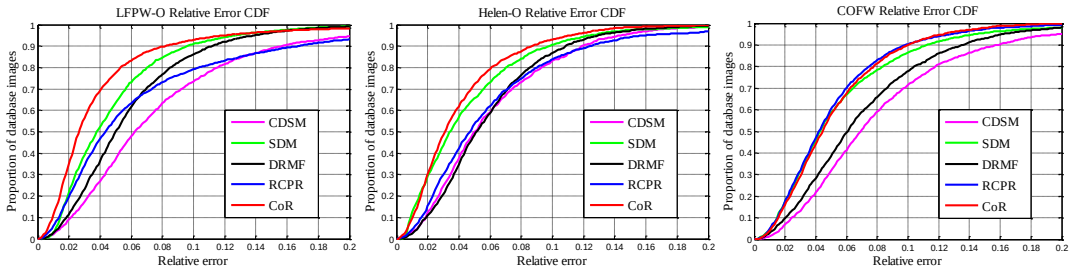
(a) LFPW (b) Helen

Figure 3.7: Relative error Cumulative Distribution Function curves for landmark localization on LFPW and Helen, comparing the proposed method *CoR* in Red curve with other state-of-the-art methods. (a) Error cumulative distribution tested on LFPW database. (b) Error cumulative distribution tested on Helen database.

Supervised Descent Method (SDM) [198], Robust Cascaded Pose Regression (RCPR) [25], Discriminative Response Map Fitting (DRMF) [5] and Optimized Part Mixture with Cascaded Deformable Shape Model (CDSM) [206]. These methods report the top performance among the literature. SDM and RCPR are non-parametric methods while DRMF and CDSM are parametric methods. The codes used for this experiments are downloaded from internet provided by the authors. The DRMF and CDSM are 66 points annotation. RCPR's annotation is flexible since it provides the training code in which the annotation can be defined by users. To compare on Helen and LFPW, we re-trained RCPR model with the same training set which we used to train our occlusion-specific regressors. SDM only provides 49 points annotation, omitting 17 profile and jawline fiducial points. To make the comparison consistent, on LFPW and Helen, we adopt 49 points evaluation over all the methods. On COFW, we adopt the intersected 19 fiducial points which are defined by all the methods.

3.7.2 Evaluation of Facial Feature Localization

Non-occluded face datasets: We compare the alignment accuracy on non-occluded faces, the images complementary to LFPW-O and Helen-O in LFPW and Helen with 4 state-of-the-art methods as shown in Figure 3.7. The measurement is Cumulative Distribution Function



(a) LFPW-O (b) Helen-O (c) COFW

Figure 3.8: Relative error Cumulative Distribution Function curves for landmark localization on LFPW-O, Helen-O and COFW, comparing the proposed method *CoR* in Red curve with other state-of-the-art methods. (a) Error cumulative distribution tested on all occluded images from LFPW database. (b) Error cumulative distribution tested on occluded images selected from Helen database. (c) Error cumulative distribution tested on COFW database.

(CDF). Almost all methods encounter failure[1] during testing. It may be from the failure of face detection, improper initialization and the algorithm itself. For fairness, we compare on the images that encounter no failure by all the methods. In Figure 3.7 (a), SDM and *CoR* (the proposed method) perform almost the same, which significantly outperform other methods with at least 10% proportion gap. In Figure 3.7 (b), the proposed *CoR* method achieves better results than all other methods. Nevertheless, considering the failure cases, besides the face detection failure, our method achieves 9.7% and 33.3% failure rate on LFPW and Helen while SDM achieves 10.2% and 36.6% respectively. The general evaluation over LFPW and Helen demonstrates that *CoR* is among the top level while marginally better than those methods.

Occluded face datasets: When evaluating on occluded faces, traditional methods may have problems, i.e. SDM extracts every landmark's local appearance information for regression. The occluded landmarks' appearance which brings in error degrades the regression results significantly. We compare all the methods on the LFPW-O, Helen-O and COFW in Figure 3.8.

From all the plots, our method accomplishes significantly better accuracy than the rest of the

[1]failure is referred to no face detected by face detector or landmark alignment result with significant deviation, i.e. average localization error larger than or equal to a proportion of inter-ocular distance.

Table 3.1: Average Root Mean Square Error (in pixels) of CDSM, DRMF, RCPR, SDM and proposed method *CoR* on LFPW, Helen, LFPW-O, Helen-O and COFW databases.

Method	LFPW	Helen	LFPW-O	Helen-O	COFW
CDSM	6.33	9.57	5.81	10.28	5.17
DRMF	4.90	9.59	5.40	10.23	4.50
RCPR	5.49	8.75	6.32	10.62	**3.38**
SDM	**3.84**	8.16	4.62	8.93	3.80
CoR	3.96	**7.23**	**3.49**	**7.18**	3.51

methods especially on LFPW-O and Helen-O. For the COFW dataset, our method approaches the performance of RCPR and is significantly better than other methods. The RCPR result on COFW is trained based on COFW. But our method is trained on part of LFPW and Helen images. When RCPR is trained on the same training set part of LFPW and Helen, the performance of RCPR on Helen, LFPW as well as Helen-O and LFPW-O is not as good as our method. Compared to general image alignment, the margin between the proposed method and SDM is larger when evaluating on LFPW-O and Helen-O. It is because our method is particularly designed with occlusion-specific regressors which shows the effectiveness in handling occlusion.

Quantitative results are evaluated in terms of Average RMSE in Table 3.1. *CoR* provides the most consistent and accurate performance against other methods on Helen, Helen-O and LFPW-O. It is very competitive to the state-of-the-arts on LFPW and COFW. As we know, the profile and jawline parts suffer the largest variance in face shape. The 49-point annotation in SDM omits the profile and jawline, which imports less variance. While in our method, we consider the profile and jawline and simultaneously optimize all the facial components, which needs to overcome more regression variance than SDM. Even so, *CoR* achieves the same while sometimes better performance than SDM.

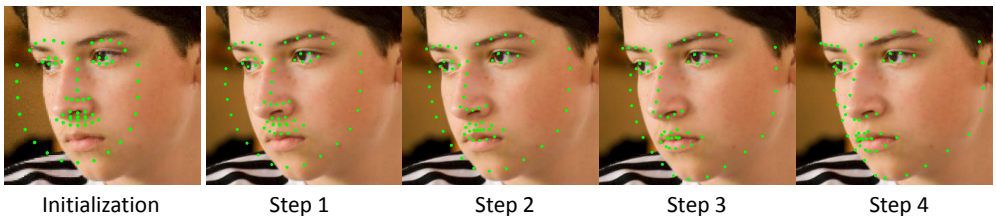
Figure 3.12 and Figure 3.13 show the visual comparison of the proposed method with SDM

and RCPR. From both the results of LFPW and Helen, SDM and RCPR encounter certain distortion on the occluded regions while our method robustly overcome the occlusion and predicts both the occluded and non-occluded regions consistently. More specifically, Figure 3.13 shows the cases with partially component occlusion, i.e. the first column shows partial mouth and partial face profile are occluded, which is a more natural case in the wild conditions. The results support that our method can successfully deal with the partially component occlusion problem. It is because from the assumption that the occlusion is a small portion of the whole face area, using all the other non-occluded areas to predict the overall face shape achieves promising results. Theoretically inside the partially occluded component, those partially non-occluded component contributes to the regression results. Since our occlusion regressor setup is component-wise, the regression cannot be further divided into sub-component update. Thus, either consider the whole component in which the partial occlusion degrades the performance or not consider the component in which the non-occluded partial component does not contribute to the shape update which should be. The visual results from Figure 3.12 consistently suggests that not using the whole component achieves better results than the other method. Figure 3.14 presents more qualitative results showing the localization precision of our method and the robust initialization results in blue rectangle.

3.7.3 Evaluation of Robust Initialization and *CoR* framework

In training, the pitch and yaw angle variations are mostly incorporated. In other words, our T cascaded regression steps could gradually converge in the presence of different pitch and yaw variance. Figure 3.9 provides an example of face with large yaw angle, at about -60°.

However, if the face image is with certain roll angle variation, since our training does not incorporate in the roll variation, the alignment procedure may fail as shown in Figure 3.9. As a matter of fact, we would introduce a robust initialization framework as proposed in our main submission to overcome such variation. Another way to deal with the roll angle is to incorporate such variation into training steps as the pitch and yaw angles. However, we think it enlarges the training space which need more cascaded steps to converge while at the same time increases complexity for the training procedure.



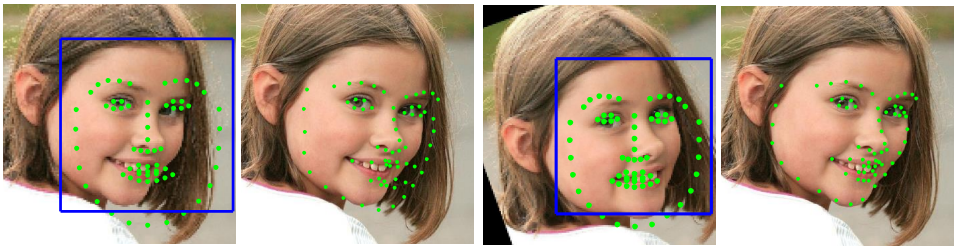
(a) Succeeded example of cascaded regression

(b) Failed example of cascaded regression with initial roll variation

Figure 3.9: Illustration of cascaded linear regressors overcoming large yaw variation. The procedure starts with initialized frontal landmarks put on the facial area. Step 1 until Step 4 are 4 cascaded regression steps for one occlusion-specific regressor. (a) The succeeded example of cascaded regression. (b) a failure case with initial roll angle variation 20°.

In this section, we investigate the effectiveness of components in our proposed method, including the robust initialization and *CoR* framework. First, the face detection drift and in-plane rotation may influence the final fitting results. In Figure 3.10 (a), since the face is clock-wise in-plane rotated and also with some yaw angle variation, the face detection result is not as tight as it should be compared to the detection result in Figure 3.10 (b). Because the initialization is too loose, the *CoR* cannot search through the feature space and thus obtain poor results in the second column of Figure 3.10 (a). We further evaluate the alignment accuracy on *CoR* with and without the new initialization method in Table 3.2 first two rows. We denote robust initialized *CoR* as *CoR* in all the experiments. The results demonstrate that robust initialization boosts the accuracy with certain margin.

Finally, we compare proposed *CoR* with N-CoR, wm-agg and gm-agg. In Table 3.2, N-CoR is the proposed framework without robust initialization, wm-agg represents the weighted mean aggregation over all T regressors and gm-agg represents geometric mean over all regressors. The table shows that *CoR* consistently outperforms wm-agg and gm-agg with a significant margin, which indicates that the Bayesian consensus of regression scheme is a more robust and



(a) Traditional method and result (b) Robust initialization and result

Figure 3.10: Visual results on traditional initialization, its fitting result, robust initialization and its fitting results. The first and third columns are the initializations while the second and the fourth columns are the fitting results. (a) Traditional initialization on a face with roll variation, the detected face bounding box in blue rectangle and the fitting result with *CoR*. (b) Proposed robust initialization on the same face in (a) with roll rectification, the detected face bounding box in blue rectangle and the fitting result with *CoR*.

effective way in optimizing the positions.

3.7.4 Evaluation of Occlusion Detection

Among the previous methods, only RCPR detects occlusion. Thus, we compare the performance of occlusion detection with RCPR. Since other databases do not provide occlusion ground truth, we only focus on COFW for evaluation. For RCPR, as the code published by the authors, we do not tune any parameter and simply use the default settings. In our method, we also fix the parameters for testing. The parameters are tuned via 3-fold cross validation. Figure 3.11 shows some visual results on occlusion detection. Compared to ground truth, the RCPR results seem to miss out many occluded landmarks while our method hit more occluded ones. Quantitatively, by holding the false alarm at the same level, our method achieves 41.44% accuracy while RCPR is with 34.16%, which reveals that our method improves the detection precision by 7%.

Table 3.2: Average Root Mean Square Error (in pixels) of Robust initialized CoR (*CoR*), Non robust initialization CoR (N-CoR), weighted mean aggregation (wm-agg) and geometric mean aggregation (gm-agg) methods on LFPW, Helen, LFPW-O, Helen-O and COFW databases.

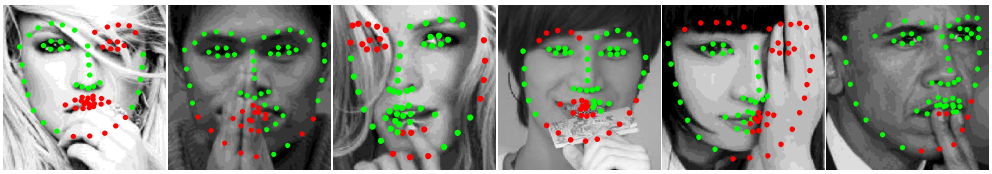
Method	LFPW	Helen	LFPW-O	Helen-O	COFW
N-CoR	4.17	7.39	3.53	7.20	3.63
CoR	**3.96**	**7.23**	**3.49**	**7.18**	**3.51**
wm-agg	4.32	7.34	3.61	7.41	3.63
gm-agg	4.43	7.66	3.65	7.53	3.66

3.8 Summary

We proposed a new consensus of regression based approach which trains an ensemble of occlusion-specific regressors to handle occluded faces in the wild. Due to the lack of occlusion priors, we conduct the consensus of the occlusion-specific regressors under a Bayesian framework to optimize the inference. In addition, to overcome the initialization problem in regression based methods, we propose a max-margin learning strategy to detect the affine transformation. A graph-based semi-supervised learning is also utilized to explicitly detect the occlusion. Our method shows consistent improvement on facial feature localization on both non-occluded and occluded face databases. Additionally, our method demonstrates improvement on occlusion detection compared to the state-of-the-art.

(a) Ground truth landmarks of COFW

(b) Localization and occlusion detection result by RCPR

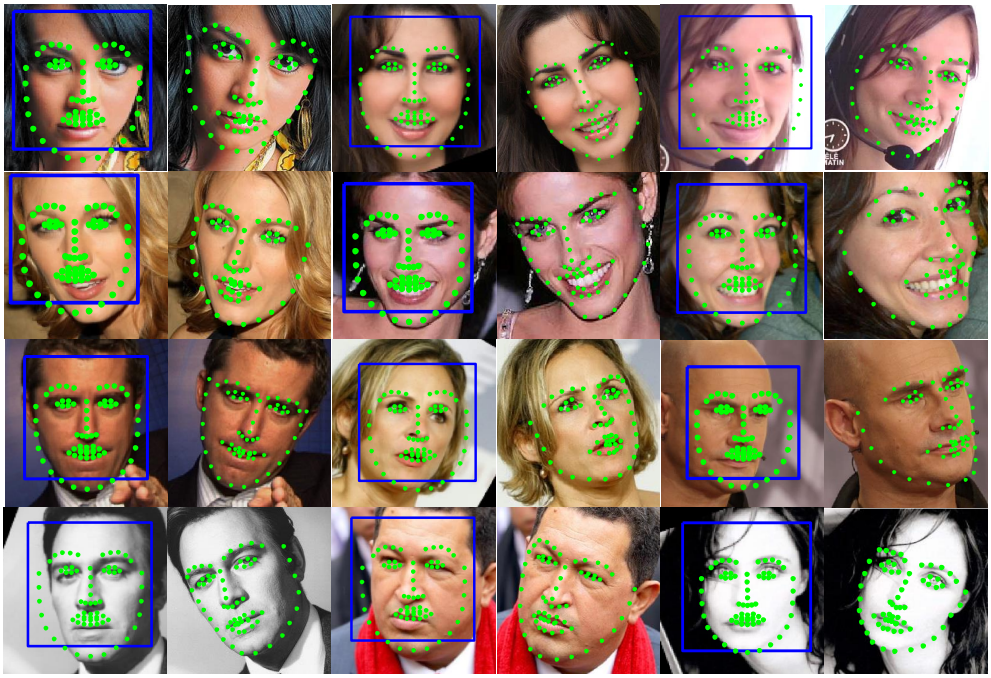


(c) Localization and occlusion detection result by proposed *CoR*

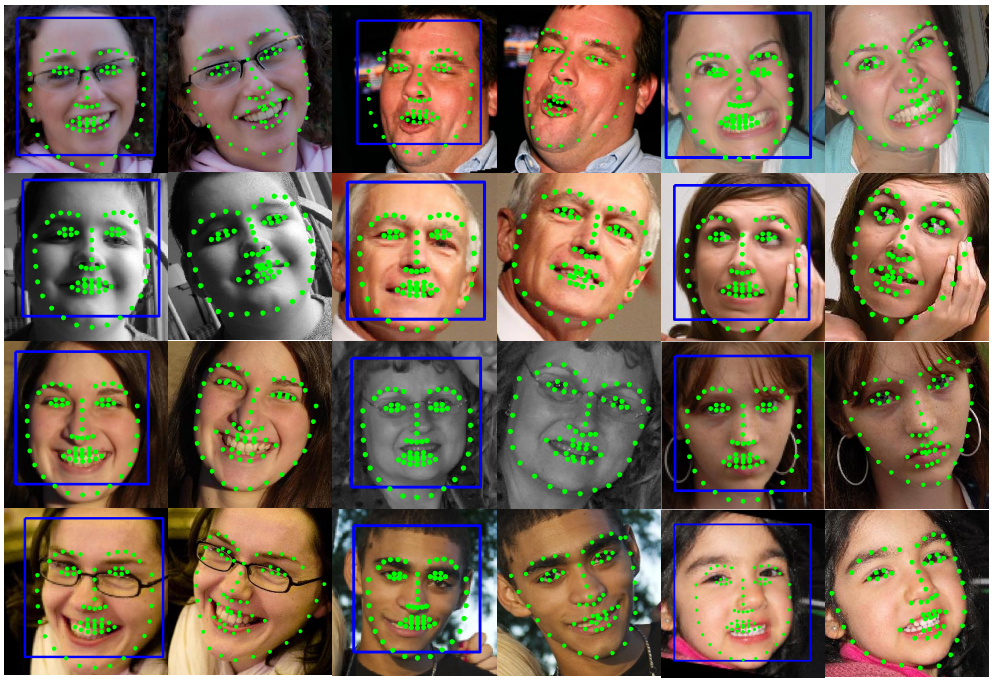
Figure 3.11: Occlusion detection comparison of *CoR* and RCPR on COFW database (Red dots: occlusion, green dots: non-occlusion). (a) The first row shows ground truth from COFW. (b) The second row shows the results of RCPR with default parameters. (c) The third row shows the results of proposed *CoR* method.

Figure 3.12: Visual comparison of SDM [198], RCPR [25] and proposed *CoR* on facial images from LFPW [14] database.

Figure 3.13: Visual comparison of SDM [198], RCPR [25] and proposed *CoR* on facial images from Helen [105] database, in which partial-component occlusions are shown.



(a) Robust initialization and Localization result on LFPW



(b) Robust initialization and Localization result on Helen

Figure 3.14: Proposed robust initialization with roll rectification, the detected face bounding box in blue rectangle and the fitting result with *CoR* in green dots. The odd columns are initialization results and the even columns are localization results.

Chapter 4

Pose- and Occlusion-robust Unified Framework

We have proposed two independent methods to handle the pose variation and partial occlusion problems separately in the previous chapters. In this chapter, we aim to conceive a unified framework which simultaneously deals with the two problems. Under the regression-based framework, we combine the pose conditioning inspired from [46] with the cascaded regression. Moreover, in order to depict the subtle local shape variance, we further employ the part-based regressors to achieve the local refinement. To connect the part-based regressors with the holistic cascaded regressors, we propose a hierarchical structure to seamlessly unify the whole process.

4.1 Introduction

Many face alignment algorithms have been proposed and have shown promising results both in accuracy and speed [27, 198, 14, 226, 37, 153]. They aim towards not only the near-frontal faces but also faces in the wild. However, due to large head pose variation, various types of occlusions, unpredictable illumination and some other factors, the landmark localization task still remains challenging.

Early representative work, such as the Active Shape Model (ASM) [40], uses a parametric model to represent a face shape and proposed an iterative framework for optimizing the landmark positions. Following these efforts, researchers have attempted to build more robust and sophisticated models which can be robust to different types of interfering conditions, such as pose and expression variations exemplified in Figure 4.1. The multi-view deformable part model [226, 69] alleviates the pose problem. However, discrete pose intervals and rigid shape modeling make it difficult to capture all possible facial variations. Recently, regression based methods [27, 198, 144] report accuracy approaching the human labeling level and their typical runtime can be within several milliseconds.

Figure 4.1: Results of our method on unconstrained face images with pose variations and occlusion. Detected occlusion landmarks are denoted in red dots and non-occluded landmarks are denoted in green dots.

However, the regression based methods may significantly suffer from uncommon part appearance due to extreme poses, lighting or expression. Noisy appearance results in bad features and the mapped landmark displacement is disturbed. On the other hand, faces in unconstrained environments are often affected by occlusion as shown in Figure 4.1. Hence, occlusion handling becomes crucial for improving these methods. An ensemble of a set of occlusion-resistant regressors relying on the probabilistic inferring is proposed to implicitly overcome occlusion [207]. Explicitly detecting the occlusion and incorporating the occlusion information into landmark detection also show some satisfactory [69]. But the mutual effect between the occlusion inference and landmark localization is still unclear.

Another important aspect for the regression-based methods is the number of iterations. Usually there is no sophisticated rule to set the "ideal" number of iterations. An empirical choice for the number of iterations may cause the regressor to under-fit the data or drift away from the correct solution. Establishing an online evaluation strategy would help to indicate how well the landmarks are localized by the current stage. Furthermore, if the occlusion information can be simultaneously inferred with the localization process, the framework would be more unified and efficient.

We propose a two-stage framework consisting of a pose-dependent holistic regression model and a hierarchical part-based regression model to robustly localize facial features. Faces with different poses are fed to different sets of pose-dependent regressors. Consequently the shape variation inside each set is largely reduced. From the holistic regressors, the hierarchical part regressors are automatically learned by our proposed projection optimization algorithm. Based upon the hierarchical part-based structure, the alignment likelihood is firstly evaluated to determine whether further local regressions are needed and also to estimate the occlusion information simultaneously; then the hierarchical part-based regression models are applied to corresponding parts to further refine the landmarks. Occlusion status are propagated to all the landmarks from the previous occlusion information during the part-based regression.

4.2 Related Work

Numerous methods have been proposed in the facial feature localization literature, e.g. deformable part model, regression, convolutional neural network, etc. Based on the types of the underlying models, we can categorize these methods into two groups, the parametric and non-parametric methods.

The seminal work of ASM [40] and AAM [35, 131] are both classical parametric methodologies, which inspired many follow-on works. The component-wise ASM [85] breaks the holistic shape into components and reduce the alignment error caused by the global constraint. A nonlinear discriminative ensemble learning for the shape parameters' update is also proposed for AAM [154]. The rotation-invariant kernel in feature space makes the AAM parameters linearly predictable [80]. Recently a fast AAM [176] was proposed for real-time alignment, an ensemble of AAM [29] was presented to align the landmarks for a image sequence and a probabilistic AAM [132] was introduced to reformulate the classic AAM problem. The constrained local model (CLM) [43, 157] investigates each landmark's local appearance and aggregates all the local patches by the conditional independence assumption.

In the non-parametric algorithms, a data-driven method [14] achieved top accuracy evaluated on Labeled Facial Parts in the Wild (LFPW). Zhu et al. [226] introduced a multi-view tree-structure part model to jointly detect faces and localize landmarks. Deep learning [215, 212] is

applied to localize the facial features as well. Recently, regression based methods receive high interests in facial feature detection due to its high accuracy evaluated on wild face databases and fast performance. The supervised descent method [198] achieves so far the best accuracy on LFPW and Ren et al. [144] claims 3000 frame per second processing speed. An early contribution from Liang et al. [112] trains component-wise classifiers to update the landmarks. A number of regression constructions have since been proposed, such as boosted regressions [44, 178, 129], regression forests [46, 201, 98], linear regressions [198, 53, 7], regression ferns [27, 25], etc.

To overcome pose variation, multi-view shape models [38, 226] were proposed either by local search to estimate the head pose or by combining models from different view-points. The regression-based methods can also handle certain pose variations, which are incorporated into the training data. However, too much pose variation increases the training complexity. Cascaded pose regression [53, 25] and conditional regression forests [46] are the most similar works to ours. The former ones take pose as an explicit factor to regress, while ours treats the pose as a conditional hidden state. The latter one partitions the poses into subspaces before the regression fitting. Within each subspace, they aggregate many regression trees to predict landmarks, while in our method, we allow the pose state to change during each step of regressions.

Regression-based methods are fast but are sensitive to occlusion. There have been several works introduced for handling occlusion, for example, Artizzu et al. [25] proposed a block-wise statistical model to approximate the occlusion. Yu et al. [207] introduced multiple regressors which are specially designed to infer the particular occlusions. A similar work [69] also used a hierarchical deformable part model to localize landmarks. This method is similar to [226] in which it sets up multi-view shape models and adopts detection based strategies to vote for the positions. To infer the occlusion status, both [69] and ours use part-based models. But for alignment, instead of detection of facial features in a pictorial structure in [69], we model both holistic and local landmark update as a regression based strategy and jointly learn the part regressors from holistic regressors in a projection optimization framework.

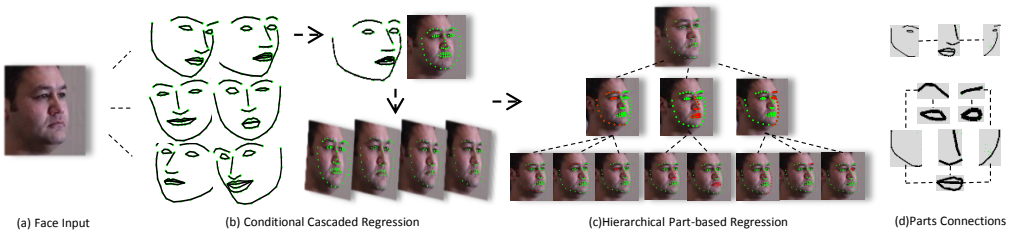


Figure 4.2: Graphical structure illustration of the proposed framework. (a) The input face image. (b) Conditioned by head poses, the facial landmarks are initialized with different priors and a cascaded regression is applied as global shape fitting. (c) The holistic shape is split into parts hierarchically to effectively overcome the local shape variance, e.g. the shape is firstly divided into left part (left profile, left eyebrow and left eye), middle part (nose and mouth) and right part (right profile, right eyebrow and right eye). The second layer is derived from the first layer by further dividing the components. (d) The geometric connections of the two layer parts defined in (c).

4.3 Preliminary

Holistic regression based methods have shown promising results in facial landmark localization [198]. However, the common challenges still exist, for example, head pose variation makes the regression harder to train, holistic fitting cannot capture local shape variation and occlusion severely degrades the alignment accuracy.

In order to overcome these challenges, we propose a hierarchical regression method with pose-dependent and part based modeling. As shown in Figure 4.2, we first propose a conditional cascaded regression model to separate the regression manifold into several subspaces. Then a hierarchical part-based model is proposed to decompose the holistic structure into a more flexible part-based hierarchical structure. Each part represents a facial component, e.g. mouth. By inferring the occlusion conditions for the parts, the landmarks' update model is regularized.

Given the definition of N facial feature points, denoted as $\mathbf{s} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N]$, and their starting position $\mathbf{s}_0$, the goal is to minimize the squared error in coordinates $\|(\mathbf{s}_0 + \Delta\mathbf{s}) - \mathbf{s}^*\|_2$, where $\mathbf{s}^*$ is the ground truth. The evidence we could observe is only the appearance feature.

Thus, (4.1) minimizes the error in feature space instead of the coordinate space.

$$\arg\min_{\Delta\mathbf{s}} \|\mathbf{h}(I(\mathbf{s}_0 + \Delta\mathbf{s})) - \mathbf{h}(I(\mathbf{s}^*))\|_2^2 \tag{4.1}$$

$\mathbf{h}$ is the feature descriptor, i.e. SIFT feature [121]. $I(s)$ are the facial image patches surrounding each fiducial point of $\mathbf{s}$. $\mathbf{h}(I(s))$ is the concatenated feature descriptor applied on each of the image patches $I(s)$. The cascaded regression based framework [198] has the shape update form as follows:

$$\mathbf{s}_{t+1} = \mathbf{s}_t + \mathbf{R}_t\phi_t + \mathbf{b}_t \tag{4.2}$$

where $\mathbf{R}_t$ is the regression matrix and $\mathbf{b}_t$ is the intercept. $\phi_t = \mathbf{h}(I(\mathbf{s}_t))$ denotes a local feature descriptor all through the work. Typically the number of iterations is fixed to 4 or 5. The cascaded regression attempts to apply a set of linear regressions sequentially to predict landmark positions. Given ground truth $\mathbf{s}^*$, the training process is to minimize the prediction error over all training samples $\mathcal{T}$ as follows:

$$\arg\min_{\mathbf{R}_t,\mathbf{b}_t} \sum_{z\in\mathcal{T}} \|\mathbf{s}^* - (\mathbf{s}_t + \mathbf{R}_t\phi_{z,\mathbf{s}_t} + \mathbf{b}_t)\|_2^2 \tag{4.3}$$

4.4 Conditional Cascaded Regression

To reduce the complexity of the face shape manifold, we divide the manifold into several subspaces, as shown in Figure 4.2 (b). Shapes are mainly clustered into three groups, the frontal view, the left view and the right view. We set the threshold angles to be -22.5° and 22.5°. Then given image I, by introducing head pose parameter θ, the regression problem becomes equivalent to solving (5.1).

$$\arg\max_{\Delta\mathbf{s},\theta\in\Theta} \; p(\Delta\mathbf{s}|I) = \frac{1}{\Xi} p(\Delta\mathbf{s}|\theta, I)p(\theta|I) \tag{4.4}$$

where Θ is the set of discrete head pose intervals, Ξ is the distribution normalizer and the pose likelihood term is learned based on the logistic regression framework:

$$p(\theta|I) = \frac{1}{\Phi} \frac{\exp(w_\theta\psi + c_\theta)}{1 + \exp(w_\theta\psi + c_\theta)} \tag{4.5}$$

where $\boldsymbol{\psi}$ is a holistic appearance feature, i.e. HoG and Φ is a normalization factor to make $p(\theta|I)$ a distribution.

The conditional alignment likelihood $p(\Delta\mathbf{s}|\theta, I)$ is modeled by the coordinate displacement in (4.6),

$$p(\Delta\mathbf{s}|\theta, I) = \frac{1}{\Gamma}\exp(-\beta\|\mathbf{R}_t(\theta)\phi + \mathbf{b}_t(\theta)\|_2) \tag{4.6}$$

where Γ is again a normalization factor. Notice that $\Delta\mathbf{s} = \mathbf{R}_t(\theta)\phi + \mathbf{b}_t(\theta)$. We assume the alignment likelihood $p(\Delta\mathbf{s}|\theta, I)$ follows the exponential distribution. At each regression iteration, we maximize the alignment likelihood in (5.1) by conditioning on different head poses. The corresponding holistic regressors are applied to update the landmark positions. Such procedure largely reduces the shape complexity caused by head poses and are more likely to converge.

4.5 Hierarchical Part-based Regression

Holistic regression is effective in aligning the face as a whole, but it may not produce perfect fitting results at local parts due to appearance and shape deformation. For instance, assuming the same face with eyes fully open and half open, holistic regressions localize landmarks as a whole but may fail in the eye region due to the lack of local constraint from the holistic regression. A part-based regression step could alleviate this problem more by deformable local fitting.

4.5.1 Part-based Local Regression

From the holistic regression, each landmark's update utilizes exactly one row of R. By dividing the facial area into parts, we partition the regression matrix $\mathbf{R}$ into row-wise blocks. Recall from Figure 4.2 (c), for the first layer, we divide the shape into left, middle and right parts. As shown in (4.7), the partition of $\mathbf{R}$ is denoted as $\mathbf{R} = [\mathbf{R}_l, \mathbf{R}_m, \mathbf{R}_r]^T$, where $\mathbf{R}_l, \mathbf{R}_m$ and $\mathbf{R}_r$ correspond to left, middle and right parts, respectively. Such division is recursively applied by further partitioning the previous layer's blocks into smaller units. Figure 4.2 (c) shows the second layer of facial components, i.e. left profile, mouth, left eye, etc.

Notice that the partition still uses the holistic feature ϕ for update $\Delta\mathbf{s} = [\mathbf{R}_l, \mathbf{R}_m, \mathbf{R}_r]^T\phi$. In other words, inside the regressors $\mathbf{R}_l, \mathbf{R}_m, \mathbf{R}_r$ themselves, the correlation in between should

be diminished. We aim to obtain local regressors from the holistic regressor $\mathbf{R}$ as shown in (4.7).

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_l \\ \mathbf{R}_m \\ \mathbf{R}_r \end{bmatrix} \rightarrow \hat{\mathbf{R}} = \begin{bmatrix} \hat{\mathbf{R}}_l & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{R}}_m & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \hat{\mathbf{R}}_r \end{bmatrix} \tag{4.7}$$

where each part regressor $\hat{\mathbf{R}}_i, i = l, m, r$ is a block-wise sub-matrix. The transformation optimization from $\mathbf{R}_i$ to $\hat{\mathbf{R}}_\mathbf{i}$ is introduced in section 4.6. After generating the local part regressors directly from the holistic ones, the part-based regression confirms to the same update rule, $\Delta\mathbf{s}_i = \hat{\mathbf{R}}_i\phi + \hat{\mathbf{b}}_i, i = l, r, m$.

4.5.2 Localization Evaluation

To determine when to halt the holistic and local regressions, we set up an evaluation function to validate the alignment. The function propagates each part's (a.k.a component's) alignment score to the upper layer and finally generate the overall alignment score. Given the k^{th} component $\mathcal{G}_k$ and its landmarks $\mathbf{s_i} \in \mathcal{G}_k$, the part score function can be defined as:

$$\mathbb{E}(I, \mathcal{G}_k) = \sum_i \mathbf{U}(I, \mathbf{s}_i) + \sum_{i,j} \mathbf{Q}(\mathbf{s_i}, \mathbf{s_j}), \mathbf{s_i}, \mathbf{s_j} \in \mathcal{G}_k \tag{4.8}$$

where $\mathbf{U}(I, \mathbf{s}_i)$ is the unary term defined in (4.9), i.e. the inner product of the feature and its corresponding weights. $\phi(I, \mathbf{s}_i)$ is the descriptor extracted at landmark $\mathbf{s}_i$. The relationship between ϕ appeared in the previous sections as descriptor and $\phi(I, \mathbf{s}_i)$ is $\phi = [\phi(I, \mathbf{s}_1), ..., \phi(I, \mathbf{s}_N)]$.

$$\begin{aligned} \mathbf{U}(I, s_i) &=< \boldsymbol{w}_i^u, \phi(I, s_i) > \\ \mathbf{Q}(\mathbf{s}_i, \mathbf{s}_j) &=< \boldsymbol{w}_{i,j}^b, \mathbf{q}(\mathbf{s}_i, \mathbf{s}_j) > \end{aligned} \tag{4.9}$$

The second pair-wise term $\mathbf{Q}(\mathbf{s}_i, \mathbf{s}_j)$ is defined as the geometric smoothness term of two landmarks in one component, i.e. $\mathbf{q}(\mathbf{s}_i, \mathbf{s}_j) = [|\mathbf{s}_i - \mathbf{s}_j|, \Delta|\mathbf{s}_i - \mathbf{s}_j|], \mathbf{s}_i, \mathbf{s}_j \in \mathcal{G}_k$, which is independent from the image I, the landmark's alignment likelihood and occlusion condition.

4.5.3 Occlusion Regularization

We then independently train a classifier $\mathcal{O}_i = \boldsymbol{w}_{o,i}\phi(I, \mathbf{s}_i) + c_{o,i}$ to provide the occlusion likelihood of each landmark at each regression step. This additional classification step produces

little overhead due to the sharing of features for both regression and occlusion detection.

Misalignment is not necessarily caused by occlusion, while occlusion can adversely affect the alignment. Suppose landmark $\mathbf{s}_i$ is occluded. The alignment score $\mathbf{U}(I, \mathbf{s}_i)$ is close to 0, which does not contribute to the overall score. By detecting occlusion of $\mathbf{s}_i$, we can equivalently set the feature at $\mathbf{s}_i$ as $\phi(I, \mathbf{s}_i) = 0$. During the process, the landmarks' occlusion condition is confidently predicted by both the occlusion detector and the alignment score.

All the landmarks' occlusion states are then modulated by the neighboring landmarks with a Markov Random Field. Landmarks with sufficiently small and large scores of $\mathcal{O}_i$ and alignment are selected as negative and positive boundary conditions respectively. By setting up the connection weights among the landmarks, a label propagation algorithm [225] is applied to assign the unlabeled landmarks.

To summarize our method, firstly the conditional cascaded regression is applied with fixed T_1 iterations and the hierarchical part score is calculated evaluating the alignment. If the score is less than confidence threshold d, the part-based regression is called. The top layer of the hierarchical part structure is applied to refine the landmarks and alignment score is refreshed. If the score is still less than d, the second layer regression is initialized. Such local two-layer regression is conducted for T_2 times or until the local refinement achieves confident score (larger than d). The proposed method is illustrated in Algorithm 3. d is a preset alignment likelihood threshold which is optimized by evaluating on randomly selected well-aligned training samples and manually disturbed samples for binary classification.

4.6 Holistic and Part Regression Training

In this section we describe the training of the holistic regressors and how to derive the hierarchical part regressors directly from holistic regressors. We also introduce the training of the graphical model for evaluating the alignment likelihood.

4.6.1 Holistic Regressor Training

We firstly introduce the training details for holistic regressors. In experiments, we tried the gaussian random perturbation of each landmark, even if the perturbation step is small, the

Algorithm 3 The two-stage regression algorithm.

Input: I, $\mathbf{s}_0$, threshold d
Output: $\mathbf{s}$, $\mathcal{O}$
repeat
  run (4.2), (4.6) and (4.5), optimize (5.1).
  evaluate $\mathcal{O}_i = \boldsymbol{w}_{o,i}\phi(I, s_i) + c_{o,i}, i = 1, ..., N$, set $\phi(I, s_{\mathcal{O}\leq 0}) = \mathbf{0}$
until T_1 times
fix θ, evaluate (4.8), if $\mathbb{E}(\mathcal{G}) > d$, halt.
repeat
  for layers in hierarchical structure
    run part-based (4.2)
    evaluate $\mathcal{O}_i = \boldsymbol{w}_{o,i}\phi(I, s_i) + c_{o,i}, i = 1, ..., N$, set $\phi(I, s_{\mathcal{O}\leq 0}) = \mathbf{0}$
    evaluate (4.8), if $\mathbb{E}(\mathcal{G}) > d$, halt.
  end
until T_2 times

regression result returns jittering shapes. When the training samples are not sufficient, we augment the initialization by rotation and random perturbation of global translation. Meanwhile, to prevent overfitting, denoting $\Delta\mathbf{s} = \mathbf{s}^* - \mathbf{s}_0$, we modify (4.3) by adding the regularization terms, which is (4.10). The problem can be solved by splitting R into row pieces and each piece-wise sub-problem is convex.

$$\min_{\mathbf{R},\mathbf{b}} \sum_{z\in\mathcal{T}} \sum_{\mathbf{s}_0} \|\Delta\mathbf{s} - \mathbf{R}\phi_z - \mathbf{b}\|_2^2 + \frac{\eta_1}{2} tr(\mathbf{R}\mathbf{R}^T) + \frac{\eta_2}{2}\mathbf{b}^T\mathbf{b} \tag{4.10}$$

4.6.2 Part-based Regressor Derivation

As we have introduced in (4.7), to convert holistic regressors $\mathbf{R}_i$ to part regressors $\hat{\mathbf{R}}_i$, we propose a projection matrix W to accomplish the transformation. The original partitioned regressor is projected onto a new subspace in which the correlation between parts is diminished, as shown in (4.11). Projection from holistic regressors is expected to preserve the global information for the local regression. Directly training part regressors only contains local information, which is sensitive to noise. Moreover, bounding the update error provides criterion for automatic update

halting.

$$\hat{\mathbf{R}}_i = \mathbf{R}_i \mathbf{W}_i, i = l, r, m \tag{4.11}$$

We expect that after transforming the original $\mathbf{R}$ into block-wise $\hat{\mathbf{R}}$, by bounding the part regression error, the holistic regression error becomes a supreme of the part regression error in (4.12).

$$\Delta \mathbf{s} = \mathbf{R}_i \phi + \mathbf{b}_i = \sup_{\mathbf{W}_i} \left\{ [\mathbf{R}_i, \mathbf{b}_i] \mathbf{W}_i \left[\phi^T, 1\right]^T \right\} \tag{4.12}$$

Thus, it leads to an optimization over $\mathbf{W}_i, i = l, r, m$ such that the local part regression further reduces the update error based on the holistic result. The above optimization problem can be formulated as:

$$\arg\min_{\mathbf{W}_i} \ \|\tilde{\mathbf{R}}_i \mathbf{W}_i \tilde{\phi}\|_2^2 + \|\mathbf{W}_i\|_F^2, \mathbf{W}_i^T \mathbf{W}_i = \mathbb{I} \tag{4.13}$$

We simplify the notation of $\mathbf{R}_i, \mathbf{b}_i$ as $\tilde{\mathbf{R}}_i = [\mathbf{R}_i, \mathbf{b}_i]$ and the raw feature is rephrased as $\tilde{\phi} = \left[\phi^T, 1\right]^T$. $\tilde{\mathbf{R}}_i \in \mathbb{R}^{(m,n)}, \mathbf{W}_i \in \mathbb{R}^{(n,m)}$, m is the number of landmarks in the corresponding part and n is the original feature dimension plus one dimension of $\mathbf{b}_i$. $\mathbf{W}_i^T \mathbf{W}_i = \mathbb{I}$ constraints that the projection of each part should be orthogonal.

By solving independently each part's transformation matrix $\mathbf{W}_i$, we obtain each local part's regressor $\hat{\mathbf{R}}_\mathbf{i}$ which is guaranteed to further shrink the localization error because the optimization of (4.13) is to find the optimal $\mathbf{W}_i$ such that the displacement from the ground truth is minimized from the holistic step. For each part's regressor, a second layer regressor can be achieved under the same construction.

4.6.3 Localization Evaluation Model Training

The weights for score calculation of landmark localization evaluation in (4.9) are learned in the following. We first concatenate bottom layer unary weights $\boldsymbol{w}_i^u$ as $\boldsymbol{w}^u$, bottom layer pair-wise weights $\boldsymbol{w}_{i,j}^b$ as $\boldsymbol{w}^b$ and upper layer pair-wise weight $\boldsymbol{w}_{i,j}^g$ as $\boldsymbol{w}^g$. We denote $\mathbf{q}(\mathbf{s}) = [\mathbf{q}(\mathbf{s}_i, \mathbf{s}_j)]$ for all pairs (i, j), in which the pair-wise smoothness features for all landmarks are concatenated. Similarly, $\mathbf{q}(\mathcal{G}) = [\mathbf{q}(\mathcal{G}_i, \mathcal{G}_j)]$ denotes the upper level pair-wise feature for all the parts. Re-arranging all the weights as $\boldsymbol{w} = [\boldsymbol{w}^u, \boldsymbol{w}^b, \boldsymbol{w}^g]$ and all the features as $\mathbf{f} = [\phi, \mathbf{q}(\mathbf{s}), \mathbf{q}(\mathcal{G})]$,

the evaluation score is $\boldsymbol{w}^T\mathbf{f}$. We set the loss function as hinge loss, which is the first term in (4.14). By regularizing $\boldsymbol{w}$ with l_2 norm, minimizing the loss function leads to solution of $\boldsymbol{w}$ for all the parts.

$$\arg\min_{\boldsymbol{w}} \sum_{\mathbf{f}_i \in \mathcal{C}} max(0, 1 - \alpha \cdot \boldsymbol{w}^T\mathbf{f}_i) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 \tag{4.14}$$

The training set $\mathcal{C}$ includes both positive and negative samples. The positive samples are the facial images with ground truth landmark positions while the negative samples are non-facial images with initialized landmarks or facial images with unaligned landmarks. α is the ground truth label taking 1 if it is positive sample and -1 if the sample is negative. The above problem can be efficiently solved by gradient descent approach.

4.7 Experiments

We evaluate our method on six benchmarks, i.e., Labeled Faces in the Wild (LFW) [84], Labeled Facial Parts in the Wild (LFPW) [14], Annotated Faces-in-the-Wild (AFW) [226], Helen [105], iBug [153] and Caltech Occluded Faces in the Wild (COFW) [25]. To evaluate the localization performance under occlusion, subsets of LFPW and Helen are selected, which are denoted as LFPW-O and Helen-O. During all the experiments, LFPW and Helen refer to the whole datasets. Our occlusion detection method is evaluated on LFPW-O, Helen-O and COFW, with comparison to the state-of-the-art.

4.7.1 Experimental Setting

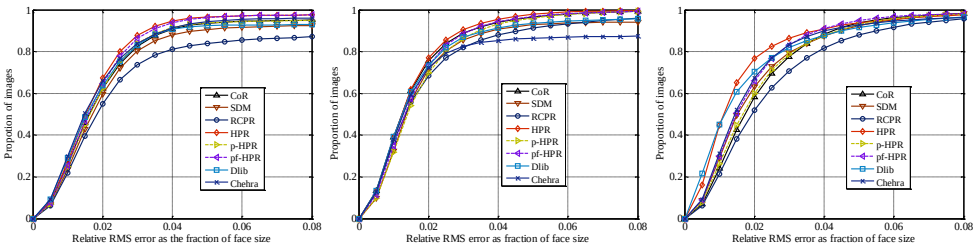
The landmark definitions vary significantly across different datasets. LFPW and COFW have a set of 29 fiducial points while Helen defines 194 landmarks and LFW includes 7 points. A re-annotation from 300 Faces in-the-wild Challenge (300-W) [153] consistently provides the LFPW, Helen, AFW and iBug with 68 key points annotation. In the experiments, we use the 66 points re-annotation from 300-W for training, omitting two inner mouth corner points for the consistency of the annotation. It is for the fair comparison with SDM [198], Chehra [7] and CoR [207], which do not include these two points. Since COFW uses 29 points annotation, when evaluating on COFW, we use the overlapped 19 points which are defined by both our

annotation and the COFW annotation. In the same way, LFW evaluation is conducted on the overlapped 7 points.

Datasets: The LFW dataset is widely used for face landmark localization, face detection and face recognition. We obtained 12007 out of 13233 images which have valid annotations, most of which are wild faces under natural conditions. In LFPW, the images vary significantly in pose, illumination and occlusion. From 300-W, 811 training images and 224 testing images are provided. We selected all occluded images, which is 112 out of 224 testing images to form LFPW-O. Helen is another wild face database containing images with large resolution from all kinds of natural conditions. It has 2000 training images and 330 testing images. The manually selected HELEN-O contains 290 occluded faces out of the overall 2330 images. The COFW testing set contains 507 images which show a wide variety of faces with different head poses and especially large portion of occlusion. AFW and iBug are the two most challenging wild face databases, since they include complex head poses, occlusion, illumination, focal length, etc.

Training: In the training of conditional cascaded regression, 2078 near frontal images, 428 left-view images and 516 right-view images are selected from LFPW, Helen and MultiPIE [74]. To increase the generalization, multiple sets of initial landmarks are generated for each training image. Firstly the facial bounding box is provided by the Viola-Jones face detector [90]. Then centering in the bounding box, the initial landmarks are parameterized by a mean shape and affine transforms. We sampled 5 times of the parameters under Gaussian assumption. Each iteration's initial shapes are from the last step's result. To train the occlusion detection model, 870 positive examples are chosen with annotations from LFPW and Helen and 870 negative samples are randomly selected from a natural scene database [87]. The landmarks for negative samples are put on with mean shape of which the position can be randomly selected within the image range.

Evaluation: We compare our method, "hierarchical part-based regression" (HPR), with five state-of-the-art methods, Supervised Descent Method (SDM) [198], Robust Cascaded Pose Regression (RCPR) [25], Consensus of Regression (CoR) [207], Dlib [98] and Chehra [7]. These methods report the top performance among the most recent regression-based methods. Methods with different experimental settings, e.g. Neural Network structures are currently not



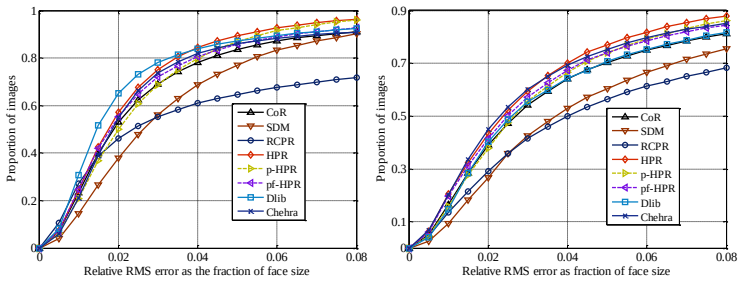
(a) LFW (b) LFPW (c) Helen

Figure 4.3: Cumulative distribution function curves of normalized error on LFW, LFPW and Helen, comparing the proposed method HPR with other state-of-the-art methods. The horizontal axis is the normalized error and the vertical axis is the image proportion of the volume of database. (a) Error CDF on LFW database. (b) Error CDF on LFPW database. (c) Error CDF on Helen database.

compared. The codes are provided by the authors from internet. RCPR provides its training code in which the annotation can be defined by input data. To make the comparison consistent, the training databases for HPR, RCPR and CoR are the same, which are LFPW and Helen. SDM is reported to be trained with MultiPIE and LFW. Since SDM and Chehra uses 49 points annotation for training and testing, we also select the overlapped 49 points from our 66 points training setup for testing on LFPW, Helen, AFW and iBug, by neglecting 17 points of face profile. For fair comparison, we use the images in which faces are successfully detected by a third-party face detector, i.e., Viola-Jones detector [90]. The success rate for detection is 97.05% for LFW, 93.02% for LFPW, 88.24% for Helen, 82.76% for Helen-O, 96.43% for LFPW-O and 59.17% for COFW. For AFW and iBug, since most of the images are failed on face detection, we provide face bounding boxes according to the ground truth landmarks for all the comparing methods for fair comparison, which largely alleviates the detection failure problem.

4.7.2 Localization on Wild Databases

As shown in Figure 4.3, our method HPR is competitive in the top performance while sometimes slightly better, e.g. the performance on Helen database. The error is calculated by dividing root mean squal pixel error over the face size. Face size is calculated as the tight bounding



(a) AFW (b) iBug

Figure 4.4: Cumulative distribution function curves of normalized error on AFW and iBug, comparing the proposed method HPR with other state-of-the-art methods. (a) Error CDF on AFW database. (b) Error CDF on iBug database.

box around the ground truth. The proposed method explicitly separates the landmarks into components which is more suitable for deforming the local shape variance. The images in Helen have large resolution, typically 2k by 2k, in which the local shape variance is enlarged compared to other databases. That is why our method's performance is comparatively better than at other databases. The average runtime on a 640 by 480 image is around 0.3s in Matlab with a dual core i7 3.4GHz CPU.

Furthermore, the AFW and iBug, two additional challenging databases, are evaluated in Figure 4.4. They contain faces with more extreme head poses, occlusion and illumination. Since large portion of the images fail in face detection, we provide each face in AFW and iBug with a bounding box according to the ground truth. Even though, the performance of all

Table 4.1: Absolute mean average pixel error of CoR, SDM, RCPR, and proposed method HPR on LFW, LFPW, Helen, LFPW-O, Helen-O and COFW databases.

	LFW	LFPW	Helen	LFPW-O	Helen-O	COFW
CoR	3.17	3.89	7.35	3.33	7.10	3.46
SDM	3.23	3.78	7.58	4.49	9.52	3.63
RCPR	5.51	4.65	8.64	5.73	9.37	**3.03**
HPR	**3.12**	**3.69**	**5.79**	**3.17**	**7.03**	3.56

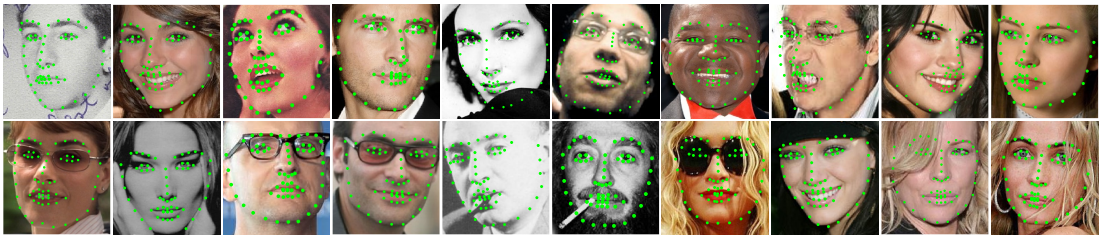


Figure 4.5: Qualitative localization results on some images from LFPW database. The first row mainly shows faces with pose variations and the second row shows faces with partial occlusion.

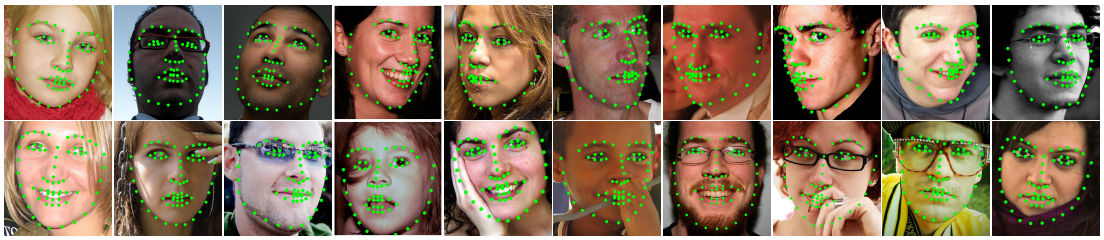
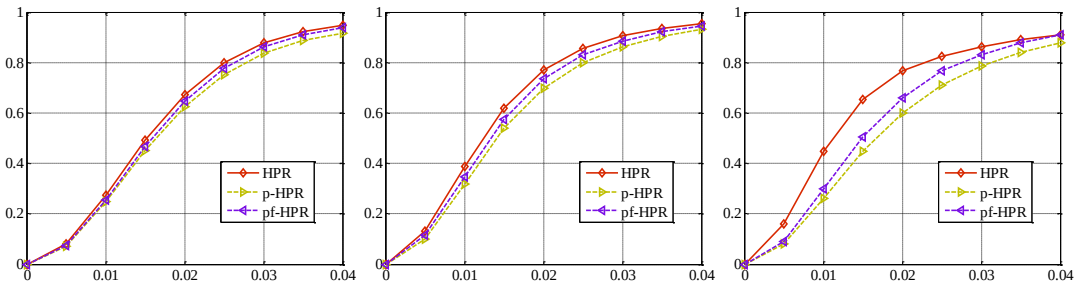


Figure 4.6: Qualitative localization results on some images from Helen database. The first row shows faces with pose variations and the second row shows faces with partial occlusion.

methods drop compared to the ones on other databases, which again suggests the challenge of the two databases. The proposed method still shows its advantage among all the comparing methods consistently.

Table. 4.1 shows the quantitative results of the absolute mean average pixel error over all the methods. The absolute mean average pixel error is defined as: for each aligned image in a database, all the n landmarks' pixel errors are calculated. Each image obtain an averaged pixel error, which is the average over the n absolute pixel errors. For all the images in the database, the mean of the average pixel errors is calculated. The table shows that the proposed HPR method outperforms the compared methods significantly on all the datasets except on COFW, where RCPR is the best. Note that RCPR is expected to perform well on COFW since it is trained on this dataset.

Qualitative results are visualized in Figure 4.5 and Figure 4.6. Figure 4.5 shows our results on LFPW. The landmarks are localized in green dots. The first row presents facial images with different pose variations. The second row shows examples with various occlusion, i.e. hat, sun glasses, moustache and beard, hands, hair, etc. Same set up is applied in Figure 4.6. The visual



(a) LFW (b) LFPW (c) Helen

Figure 4.7: Cumulative distribution function curves of normalized error on LFW, LFPW and Helen, comparing the proposed method HPR with its module-wise methods, p-HPR and pf-HPR. (a) Error CDF on LFW database. (b) Error CDF on LFPW database. (c) Error CDF on Helen database.

results reveal that our method can handle unconstrained real-world cases well.

4.7.3 Component Analysis

In our framework, a pose-dependent cascaded regression is proposed as the holistic shape deformation. It is essentially an integrated shape regressor. We denote the pose-dependent cascaded regressor as p-HPR. Furthermore, the hierarchical part-based regressors are designed to divide the searching space into subspaces, which is shown in the first layer of Figure 4.2 (c). We denote this stage method as pf-HPR. Finally, the second layer component-wise regressors are applied on top of the pf-HPR, which is our proposed HPR method.

To investigate the performance of each module, we run the three methods on LFW, LFPW and Helen. The results are shown in Figure 4.7. Across all the databases, the three methods show spaced CDF curves with HPR on top, pf-HPR in the middle and p-HPR at the bottom. Each is with a significant margin from each other. The results indicate that each module of our method is carefully designed and is effective to boost performance. We further notice that the margin is increasing from LFW to LFPW and from LFPW to Helen, which may indicate the difference of difficulty in face alignment. As we introduced, Helen contains more images with larger shape variance because of the large image resolution and more pose variation and occlusion.

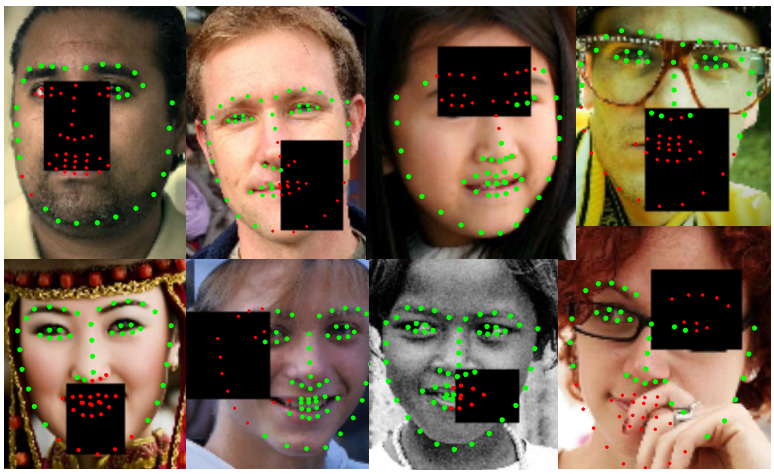


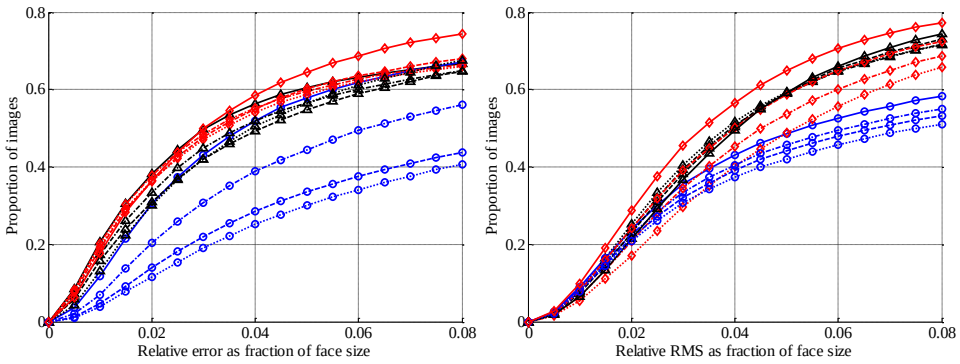
Figure 4.8: Synthesized occlusion samples from AFW and Helen. The occlusion is shown as the black boxes. Localization is presented as green dots for non-occluded points, and red dots for occluded ones.

4.7.4 Localization on Occluded Datasets

For validating accuracy, an evaluation is conducted on the selected occlusion datasets, LFPW-O and Helen-O and the specific occlusion database COFW. Quantitative results from Table. 4.1 show that HPR achieves consistently better results on the occlusion datasets especially comparing the two occlusion-robust methods, CoR and RCPR. Note that RCPR is trained based on the COFW itself while other methods including ours are not trained on this database.

To further evaluate the robustness of our method with respect to the occlusion, we synthesize occlusion data from Helen and AFW. We obtain the tight bounding boxes from the ground truth for each face. Then by randomly selecting a center point, we put on a black bounding box centered at the selected point, which is shown in Figure 4.8. The occlusion ratio is controlled by the black bounding box area over the facial area, i.e. 5%, 10%, 15% and 20% in the experiment. Samples in Figure 4.8 includes almost all types of occlusions, e.g. nose occlusion, mouth occlusion, eye occlusion, etc. The sample localization provides not only the accurate landmark positions but also the accurate occlusion detections.

We test our method against the other two occlusion-robust methods, RCPR and CoR on our synthesized 5%, 10%, 15% and 20% occlusion from Helen and AFW test sets. The CDFs is compared in Figure 4.9. Regarding the different levels of occlusion, all the methods show



(a) Helen (b) AFW

Figure 4.9: Cumulative distribution function curves of normalized error on Helen and AFW, comparing HPR with RCPR and CoR with occlusion portion 5%, 10%, 15% and 20%, in which the solid line, dash-dot line, dash-dash line and dot line correspond to each occlusion level respectively. Red lines stands for HPR, black ones represents CoR and blue lines are RCPR's results.(a) Error CDF on Helen database. (b) Error CDF on AFW database.

accuracy decrease from the small occlusion level to the large occlusion level. RCPR is relatively more sensitive to the occlusion conditions since the accuracy of different occlusion levels varies largely. Our method performs better on Helen and on AFW at lower occlusion level. But it shows sensitive trend on AFW when the occlusion portion increases. CoR is not so sensitive to the occlusion level because it applies plenty of independent occlusion-robust regressors. The large amount of regressors alleviates the degradation caused by the increase of occlusion portion.

Occlusion Detection: Since only COFW provides the occlusion annotations and only R-CPR and CoR predicts the occlusion landmarks, we compare the occlusion detection with R-CPR and CoR, in which the default settings for RCPR and CoR are applied. In our method, by tuning the graph connection weights, the occlusion label propagation is changed correspondingly. Then the hit rate and false alarm rate can be compromised manually. By keeping the false alarm rate at the same level, the proposed method achieves 41.7% hit rate, while CoR is 41.4% and RCPR is 34.2%.

Figure 4.10 shows some visual results of occlusion detection as well as landmark localization. The occlusion patterns are various, including hands, hair, sunglasses, objects, beard,

Figure 4.10: Visual results of localization and occlusion detection on some images from COFW. Green dots indicate the non-occluded landmarks and red dots show the occluded landmarks.

etc. From the results, our method precisely detects the occlusion regions although there may be false alarm due to the neighborhood gaussian constraint. At the same time, the results verify that our method accurately and robustly predict the landmark positions even when large occlusion is present.

4.8 Summary

We proposed an unified framework of conditional cascaded regression and hierarchical part-based regression to jointly localize the facial features. With conditioned regressions, head pose variation is controlled within each separated subspace and the global shape is fast localized. With hierarchical part-based regression, the alignment is evaluated and occlusion information is fed back to the local regressions. Meanwhile, local shape variance is compensated by the part-based regression and the occlusion information is propagated to other landmarks at the last step. The high localization accuracy and fast performance provide potentials for more applications such as face tracking. Our framework can be directly extended to the structured object localization as well.

Chapter 5

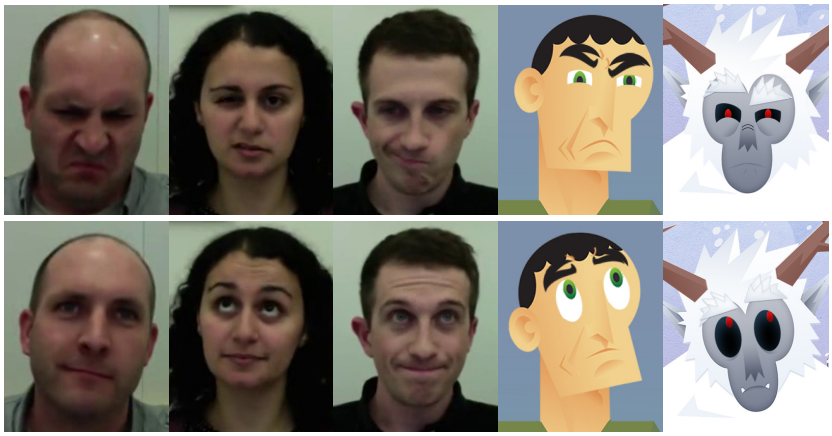
Application I: User-defined Expression Recognition for Cartoon Animation

In this chapter, we address the user-defined expression recognition problem. As a direct application of facial landmark localization, the expression recognition requires accurate landmark positions for geometric feature extraction and the facial region normalization for appearance feature extraction. The input to our algorithm is a set of 1-2 second customized expression frames recorded by a single user. We extract a combination of *handcrafted features* and regularized Deep Convolutional Neural Network (CNN) features for the expression classification. We apply a Hidden Markov Model (HMM) on top of the SVM prediction to online smooth the temporal sequence.

5.1 Background

Animating virtual characters has become a critical task in the production of movies, television shows, computer games, and many other types of digital media. Traditional character animation typically involves keyframing of animation parameters that define how the character moves. While keyframe-based animation gives the user fine-grained control, it requires a large amount of time, effort and skill to produce high quality results. More recently, advances in motion capture technology have enabled performance-driven workflows where users control characters by acting out the desired motions with their faces and/or bodies. This authoring modality allows users to quickly create expressive character animations without having to explicitly define how each individual animation parameter changes over time.

In most performance-driven systems, the continuous motion of the user is directly transferred to the virtual character. While this approach is suitable in some animation scenarios (e.g., creating realistic motion for virtual characters in live action movies), continuous motion



(a) Customized expressions (b) Cutout characters

Figure 5.1: *Performance-driven cutout character animation.* Actors perform customized expressions in (a) e.g. "disdainful" (top) and "daydreaming" (bottom) to animate the expressions of various cutout characters in (b). Note that the large inter-person expression variations even within the same expression category.

alone is not sufficient for all styles of animation. In particular, *cutout* animation is a popular style of 2.5D animation that combines continuous transformations of visual elements with discrete replacements of artwork. These replacements allow animators to drastically alter the appearance of certain visuals and are often used to change the expression of a character (see Figure 5.1 and Figure 5.5). Since most existing systems do not support performance-based triggering of artwork replacements, they cannot directly support the creation of cutout character animations.

In this work, we propose a customized facial expression recognition method that enables authoring of cutout character animations via facial performance . We focus on facial animation since it is a critical component of most character animation scenarios. Our approach addresses the following unique challenges of building a practical performance-driven cutout character animation system:

Wide range of expressions. Expressive cutout animation characters exhibit many different facial expressions that help define the unique personality of the character. It is thus important for the expression recognition algorithm to handle a wide range of expressions. Moreover, since

animators often use different expressions for different characters, the algorithm must be flexible enough to handle a customizable rather than predefined set of expressions.

Minimal training. One way to support customized expressions is to allow actors to train the system online to recognize specific expressions. Training frames are recorded in a short period and thus very few. Given the wide range of expressions used in a typical animation, it is important to minimize the required training effort.

Real-time recognition. A key benefit of performance-driven animation is that actors can quickly experiment with different timings and motions by acting out a few variations of a performance and evaluating the resulting animations. To realize this benefit, the animation system should be able to recognize expressions in real-time so that the user receives immediate feedback on the results.

Facial expression recognition is a widely explored topic in computer vision. Significant efforts have been made to boost recognition accuracy through better feature representations [216, 169, 159, 217, 18] and better strategies to discriminate expression categories [32, 13, 160]. However, most of these techniques are designed to recognize just the canonical expressions, i.e. angry, disgusted, scared, happy, sad and surprised. As explained above, a practical performance-driven cutout animation system must support a much wider range of expressions. Moreover, non-canonical expressions often exhibit far more inter-person variations, even within a single expression category, which indicates the need for customized recognition. For example, Figure 5.1 shows the non-canonical expressions “disdainful” and “daydreaming” performed by three different people who have very different interpretations of these sentiments.

5.2 Related Work

From approach point of view, facial expression recognition is generally divided into two mainstreams: emphasizing feature extraction and designing classifiers. Most of the features are handcrafted features, i.e. Gabor wavelets [216, 169], Haar feature [202, 194], Local Binary Patterns (LBP) [159, 217, 179], which are all extracted from patch appearance. Geometric handcrafted features are also proposed in the literature [216, 51]. In contrast to the handcrafted

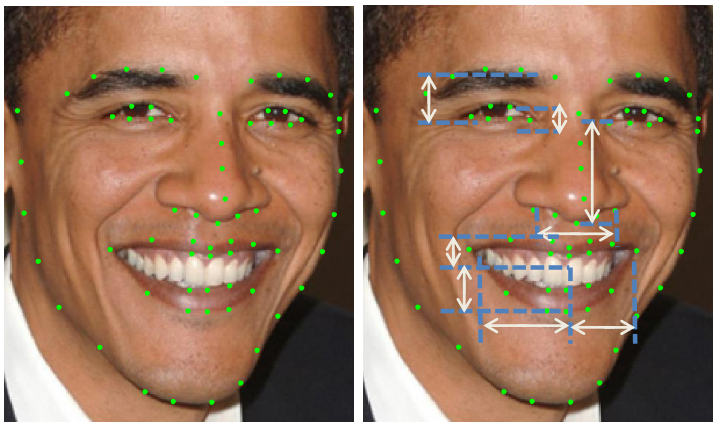
features, the learning based strategies are more and more developed recently, e.g. methods utilizing sparse representations [221, 205, 113, 219]. Some learning based features directly model the dynamic sequential information such as boosted coded dynamics [202]. With powerful representation ability, deep Neural Networks are also employed in the expression recognition task [149, 115, 116].

Fusion of features is an important branch of feature representation. Many researchers created a number of fusion algorithms to boost the recognition performance [168, 205, 210]. However, to the best of our knowledge, those fusion methods are based on the above mentioned appearance features, i.e. Gabor, LBP, etc. There may be an upper bound of the performance by combining the appearance features. If we explore the appearance feature fused with Neural Network features, due to the CNN's strong representability [101], an improved performance can be expected. Moreover, our method is an embedded structural fusion, not a simple concatenation, which provides a new channel for feature fusion.

With finely designed expression representations, Support Vector Machines (SVMs) [13, 159, 219] is the most common and effective method for recognition. Some variants are proposed to extend its applicability [30, 117]. There are some other classifier modelings, such as laplacian ordinal regression [152]. The static image based approaches are suspicious to perfectly solve the problem because of the lack of utilizing the dynamic temporal information. Thus, many dynamic models are proposed such as Hidden Markov Model (HMM) and its variants [32, 160], dynamic Bayesian network [92] and latent conditional random fields [86]. Some other methods model the spatial-temporal cube as a longitudinal atlas [78] or as expressionlets forming a spatial-temporal manifold [116].

5.3 Facial Expression Features

We consider three different types of features for representing facial expressions: geometric features, which describe the spatial deformations of facial landmarks; appearance features, which capture the appearance of the most discriminative facial regions for expression recognition; and CNN-based features that we extract from a deep neural network trained to recognize generic facial expressions. We also consider a concatenation of the geometric and appearance features,



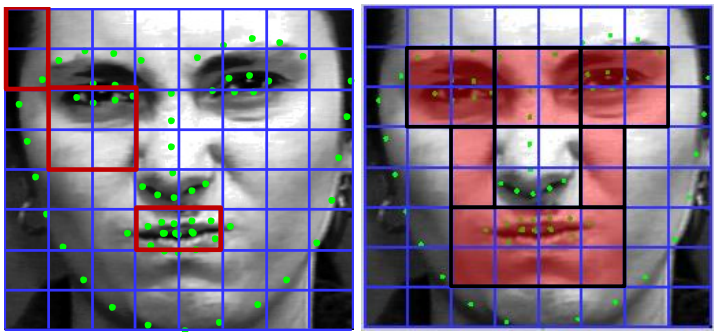
(a) (b)

Figure 5.2: Geometric feature definition. (a) Facial image with detected facial key points in green dots from a state-of-the-art face alignment method [198]. (b) The defined geometric parameters, left/right eyebrow height, left/right eyelid height, nose height, nose width, upper lip height, lower lip height, left mouth corner to mouth center distance and right mouth corner to mouth center distance.

which we refer to as our handcrafted feature vector.

5.3.1 Geometric Features

To capture deformations caused by the activation of facial muscles, we define geometric features that capture the 2D configuration of facial landmarks (Figure 5.2). Since expressions are mainly controlled by muscles around the mouth, eyes and eyebrows [54], we focus on features that characterize the shape and location of these parts of the face. Specifically, our features include the following measurements: the left/right eyebrow height (vertical distance between top of the eyebrow and center of the eye), left/right eyelid height (vertical distance between top of an eye and bottom of the eye), nose height (vertical distance between bottom of the nose and center of both eyes), nose width (horizontal distance between leftmost and rightmost nose landmarks), upper lip height (vertical distance between top and center of the mouth), lower lip height (vertical distance between bottom and center of the mouth), left mouth corner to mouth center distance, and right mouth corner to mouth center distance. To ensure that these measurements are consistent across different images, we transform each face into a frontal view (via an



(a) (b)

Figure 5.3: Selected region for appearance feature by the facial region selection. (a) The normalized facial image with detected facial key points in green dots, 8x8 patches in blue lines and blocks defined in red rectangles. Images are consistently normalized by aligning facial components, i.e. eyebrows and eyes are normalized into corresponding patches. (b) The selected regions in red. The regions are selected by evaluating on the frequency of each block's being selected based on multiple independent optimization processes.

affine deformation) and normalize the scale based on inter-ocular distance. Note that a similar set of geometric features has been validated in [51].

5.3.2 Appearance Features by Facial Region Selection

While geometric features capture spatial deformations of facial landmarks, they do not consider the appearance changes caused by such deformations. We define patch-based appearance features using a method inspired by Zhong et al. [219]. First, we partition the face image into a uniform grid of 8x8 image patches, and then we consider all 2×1, 2×2 and 1×2 blocks or regions of patches covering the entire image (Figure 5.3) allowing overlap. We then compute HoG features on each block and concatenate these features into an integrated feature vector. While we could potentially use this integrated feature vector directly to represent appearance, only a subset of the concatenated HOG features are actually meaningful for distinguishing between different expressions. We use the following data-driven approach to select the best set of features to include.

For the training data, we assume a set of face images, each of which is labeled with one

of T expression categories. For each expression category t, we create a set of tuples (x_i^t, y_i^t) where $x_i^t \in \mathbb{R}^M$ is the integrated feature vector for the i^{th} image and $y_i^t \in \{-1, 1\}$ indicates whether the image is a positive ($y_i^t = 1$) or negative ($y_i^t = -1$) example of category t. For each category t, we define a weight vector w^t that represents a separating hyperplane such that $y = (w^t)^T x_i^t + b^t$ is the classification prediction for x_i^t. We define an overall weight matrix $W \in \mathbb{R}^{T \times M}$ for all expression categories by setting its t-th row $W(t,:) = (w^t)^T$. Then, we decompose the matrix into a concatenation of sub matrices $W = [w_{C_1}, ..., w_{C_K}]$, where w_{C_j} corresponds to the weights for the j-th block across all T expression categories and C_j indicates the patches that belong to j-th block.

During training, we try to minimize the classification error over all the expression categories while requiring that W satisfies a structured group sparsity property. We can formulate this problem as multi-task sparse learning, where recognizing each of the T independent expression categories represents the individual tasks. Specifically, we define the problem as follows:

$$\underset{W \in \mathbb{R}^{T \times M}}{\arg\min} \sum_{t=1}^{T} \frac{1}{n} \sum_{i=1}^{n} L(W, X^t, Y^t) + \lambda R(W), \tag{5.1}$$

where n is the number of training face images, X^t is a matrix with $\{x_i^t\}$ as columns, and Y^t is the concatenated label vector for all examples for category t. $L(W, X^t, Y^t)$ is the loss evaluation over expression t classification and $R(W)$ is the regularization term selecting the block-wise patches. We choose the loss function as logistic loss as shown in (5.2).

$$L(W, X^t, Y^t) = \log(1 + \exp(-Y^t \odot (WX^t))) \tag{5.2}$$

where $\odot$ refers to element-wise product. For regularization, we use $l_{1,2}$ to enforce group sparsity as shown below.

$$R(W) = \sum_{j=1}^{K} \|w_{C_j}\|_2. \tag{5.3}$$

To solve this multi-task sparse learning problem, an accelerated algorithm can be referred to [196]. After solving the optimization, by thresholding $\|w_{C_j}\|_2$, the facial components are selected for the our classification. Independent such pursuits are conducted based on randomly chosen training sets from the canonical facial expression datasets multiple times. The robust selection is shown in Figure 5.3 (b) red regions. As expected, the selected regions are surrounding important facial areas, such as eyes, eye brows, and mouth. Once the facial regions

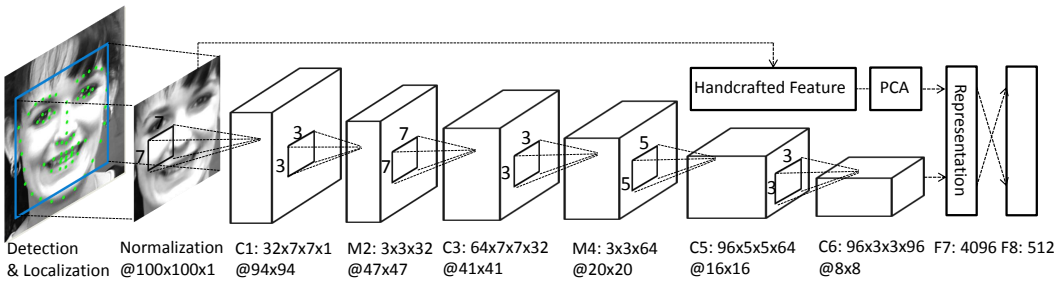


Figure 5.4: Outline of the Regularized CNN architecture. The input image is normalized as 100x100 pixels. The convolutional layers, max pooling layers and fully connected layers are denoted as C, M and F followed by the layer number. The number of channels are illustrated by the width of cuboid. They are also denoted as number of filters by horizontal filter size by vertical filter size by channels. Local receptive fields of neurons are illustrated by small squares in each layer.

are fixed, LBP and HoG features are extracted from each region and the final appearance feature is obtained by concatenating all of them.

5.3.3 Standard and Regularized CNN Features

By combining the geometric and appearance features into a single handcrafted feature vector, we can capture much of the relevant variation across different expressions. However, recent results have shown that the features extracted from deep CNN can also be useful for a variety of image understanding tasks. We experimented with two CNN structures for defining facial expression features.

CNN. This structure consists of multiple convolutional layers followed by max-pooling layers and several full-connected layers as in [101]. The network parameters are detailed in the bottom path of Figure 5.4 where “C” denotes convolutional layers, “M” denotes max pooling layers and “F” denotes fully connected layers. The softmax layer is not shown in the figure.

Regularized CNN (r-CNN). Since handcrafted features typically demonstrate good generalization behavior, we introduce the regularized CNN (r-CNN) structure in Figure 5.4. The proposed r-CNN has two paths: the top path extracts handcrafted features followed by PCA dimensionality reduction, and the bottom path is the standard CNN; the two paths are fused in

the fully connect layer F7. Trained from scratch, the r-CNN learns a deep model combines the best of both worlds; CNN-based features that perform well on constrained recognition tasks and handcrafted features that generalize well to more categories.

Network Training. To learn deep CNN models that generalize well across a wide range of expressions, ideally we need sufficient training data with a large number of expression categories. Unfortunately, all publicly available facial expression databases only include six canonical expressions — angry, disgusted, scared, happy, sad, and surprised. A CNN trained with these datasets would be tuned to classify these six expressions, which may hurt its ability to generalize to customized expressions. Moreover, creating ground truth datasets with additional expressions may be challenging, since non-canonical expressions tend to have larger inter-person variations that make accurate labeling a difficult task. Thus, our approach is to use existing canonical expression databases, including CK+ [93, 123], MMI [180], 3DFE [204] and 4DFE [203], for training both the standard and regularized CNN models. As shown in subsection 5.5.1, we indeed find that, while the standard CNN model perform extremely well on the six canonical expressions, exceeding state-of-the-art methods significantly, they do not generalize well to arbitrary customized expressions, especially when we use the last fully connected layer output as our features. Instead, we use the outputs of C6 for standard CNN, and the combination of C6 outputs and handcrafted features for r-CNN. Our experimental results show that r-CNN performs the best for both canonical expression recognition and customized expression recognition tasks.

To train our CNN models, we augment the canonical expression datasets by generating variations of each face via cropping, horizontal flipping, and perturbing aspect ratios. In the end, we obtain around 1 million data samples from the existing facial expression datasets mentioned above. We normalize the detected faces to 100x100 as the inputs to our network models. Considering the forward propagation, the output of each layer is the linear combination of the inputs non-linearly mapped by an activation function:

$$u^{k+1} = f((\mathcal{W}^{k+1})^T u^k) \tag{5.4}$$

where u^k indicates the k^{th} layer output, $\mathcal{W}^k$ indicates the weights that connect to each output node and $f(.)$ is the nonlinear activation function, for which we use rectified linear unit (ReLu)

as in [101]. To update the weights of each layer, back propagation is applied:

$$\delta^k = (\mathcal{W}^k)^T \delta^{k+1} \frac{\partial f}{\partial u^k}, \tag{5.5}$$

where δ^k is the increment of weights at layer k. For training the r-CNN, we split the weights connecting F7 into two parts: weights for the handcrafted features $\mathcal{W}_h^7$ and weights for C6 $\mathcal{W}_c^7$. We initialize $\mathcal{W}_c^7$ to 0 and only update the weights connecting F7 and F8 according to the handcrafted feature inputs. Upon convergence, we fix $\mathcal{W}_h^7$ and update the whole CNN network. In this way, the CNN generates features that are complementary to the handcrafted features and improve the overall classification accuracy. As mentioned before, we then combine the handcrafted features with the output of C6 as our r-CNN feature.

5.4 Performance-driven Cutout Character Animation

We describe a customized expression recognition framework that uses the features described above to support performance-driven cutout character animation. In our approach, an animator first demonstrates all the customized expressions that the system should recognize by recording a few seconds of video for each expression. These demonstrations act as training data for a set of SVM-based ensemble classifiers, one for each expression. To animate a character, the animator simply performs the desired motion. The ensemble classifiers recognize the current expression in real-time, and the system uses the detected expression to trigger the appropriate artwork replacements in the character. We also apply continuous deformations to the character based on the motion of the tracked facial landmarks on the actor.

5.4.1 Online Classifier Ensemble Learning

For each of the T expressions that the user demonstrates to the system, we train an ensemble classifier as follows. We take all the n_i training frames from the demonstration of expression i as positive samples and treat the recorded frames from all the other expressions as negative samples. Note that n_i is typically far less than $\sum_{j \neq i} n_j$. Thus, we randomly split all the negative samples into $N = \frac{\sum_{j \neq i} n_j}{n_i}$ piles, each of which has approximately n_i samples, and then train N independent SVM classifiers. We repeat this procedure independently t times to

produce tN classifiers, which we combine linearly to obtain the final ensemble classifier for expression i:

$$F_N(x) = \sum_{j=1}^{tN} \omega_j f_j(x), \tag{5.6}$$

where f_j is the j-th SVM classifier trained using the positive samples and the j-th pile of negative samples, and ω_j is its associated weight that is initialized as $\frac{1}{tN}$. During online testing, among the tN classifiers, some of the classifiers may produce results that conflict to the final classification output of F_N. To give our classifier a certain amount of online adaptation ability, we penalize those violating classifiers by decreasing their weights with a small amount of decay β,

$$\omega_j = (1 - \beta)\omega_j. \tag{5.7}$$

Then all the weights of tN classifiers are normalized to unit sum for next iterations, i.e.,

$$\omega_j = \frac{1}{\sum_k \omega_k}\omega_j. \tag{5.8}$$

By adjusting the contributions of the ensemble of classifiers, our algorithm can achieve robustness to slight mismatches between the few recorded training samples and the same expression demonstrated in a performance. Note that the proposed ensemble of classifiers can be regarded as a generalization of exemplar-based SVM [126] in order to gain some robustness in case of scarce training samples.

5.4.2 Temporal Smoothing with HMM

Without considering temporal information, frame-by-frame classification using the ensemble classifier could produce jittering artifacts (i.e., flipping rapidly between two or more expressions). To smooth the classification results, we apply an online sequential Hidden Markov Model (HMM). The HMM maximizes the joint probability of the current hidden state s_t and all the previous observations $x_{\{1,2,\ldots,t\}}$. Here, the hidden state s_t is the underlying expression category while the data observations are the captured facial expressions. We denote the joint probability as $\alpha(s_t) = p(s_t, x_{\{1,2,\ldots,t\}})$. By Bayesian inference, the recursion function of

updating the joint probability is shown bellow.

$$\alpha(s_t) = p(x_t|s_t) \sum_{s_{t-1}} p(s_t|s_{t-1})\alpha(s_{t-1}) \tag{5.9}$$

where $p(x_t|s_t)$ is the expression recognition posterior and $p(s_t|s_{t-1})$ is the state transition probability. In the transition matrix, for each non-neutral expression, the probability of a self-transition (i.e., remaining in the same expression) and a transition to the neutral expression are the same. In addition, transitions from the neutral expression to every other non-neutral expression are equally likely. The probability of a self-transition from the neutral expression is independent. Between one non-neutral expression and another non-neural expression, we always assume there are neutral frames. Thus, the transition matrix contains 4 independent variables. It can be obtained through cross validation with multi-dimensional line search. For the posterior $p(x_t|s_t)$, according to Bayes' rule, $p(z_t|s_t) \propto p(s_t|z_t)$ (uniform prior on all customized expressions), where the likelihood $p(s_t|z_t)$ can be approximated by converting our classifier outputs in (5.6) into probabilities with the softmax function.

5.5 Experiments

In this section, we evaluate our algorithm on both canonical expression recognition datasets including CK+ [93, 123] and MMI [180], as well as our customized expression recognition dataset for cutout character animation.

5.5.1 Evaluation on Canonical Expression Datasets

The CK+ dataset contains 327 labeled expression sequences, containing seven expressions, i.e., anger, contempt, disgust, fear, happiness, sadness and surprise. For single image based expression detection, we manually remove the first half of each sequence that appears to be neutral or less intensive than the normal expression. In this way, a total of 1988 images from CK+ are generated for training and testing. We randomly selected 300 images for testing and leave the rest for training. The MMI dataset includes expressions from 30 subjects with different ages, gender and ethnicity. Six canonical expressions are defined as anger, disgust, fear, happiness, sadness and surprise. The entire image volume for MMI is 534. We randomly select 30 images

Table 5.1: Expression recognition average accuracy on geometric feature (Geo), appearance feature (App), the geometric and appearance combined handcrafted feature (HC), CNN and regularized CNN (r-CNN) feature testing on CK+ and MMI datasets. Some state-of-the-art methods, i.e. ITBN [192], CSPL [219] and LFEA [78] are also listed for comparison.

Method	CK+							MMI						
	Angry	Disgust	Fear	Happy	Sad	Surprise	Ave.	Angry	Disgust	Fear	Happy	Sad	Surprise	Ave.
Geo	0.84	0.76	0.58	0.88	0.66	0.75	0.81	0.35	0.75	0.45	0.92	0.85	0.94	0.71
App	0.87	0.96	0.97	0.87	0.93	0.87	0.91	0.62	0.80	0.48	0.95	0.84	0.97	0.78
HC	0.96	0.97	0.95	0.96	0.99	0.90	0.96	0.62	0.97	0.67	1.00	0.96	1.00	0.87
ITBN	0.91	0.94	0.83	0.89	0.76	0.91	0.87	0.47	0.55	0.57	0.71	0.66	0.63	0.60
CSPL	0.71	0.95	0.81	0.95	0.88	0.98	0.88	0.50	0.79	0.67	0.83	0.60	0.89	0.71
LFEA	0.95	0.98	0.95	0.99	0.97	0.99	0.97	0.92	0.95	0.94	0.97	0.92	0.94	0.94
CNN	1.00	0.99	0.99	1.00	1.00	1.00	0.99	1.00	1.00	**1.00**	0.99	0.99	**0.99**	0.99
r-CNN	**1.00**	**0.99**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.99	**1.00**	**0.99**	0.98	**0.99**

per category as testing and the rest for training. To make all comparisons consistent, we remove the contempt category from CK+ to form the same 6 canonical expression categories.

We evaluate each component in our combined feature set on these two datasets and summarize the results in Table 5.1. The combined geometric feature and appearance feature produces significant performance boosted from either geometric or appearance feature alone. Even compared to some state-of-the-art methods listed in the table [192, 219, 78], the handcrafted features already achieve competitive performance. Using our CNN and r-CNN features, the performance on both CK+ and MMI approaches 100%, which demonstrates CNN's capability in the canonical expression recognition task. The regularized r-CNN features improve the result of CNN marginally since both are working extremely well. However, the advantages of r-CNN will become clearer in the following section when being evaluated on the customized expression dataset.

Table 5.2: Precision/Recall, F1 score and correction ratio (C-Ratio) comparison on geometric feature (Geo), appearance feature (App), handcrafted feature (HC) combining Geo and App, CNN feature of C6 layer (CNN-c6), CNN feature of F7 layer(CNN-f7), simple combination HC + CNN-fc7 and HC + CNN-c6, and our regularized CNN (r-CNN) feature. The classifier for all features is HMM.

Feature	Precision	Recall	F1 Score	C-Ratio
Geo	0.66±0.14	0.63±0.13	0.65	0.19±0.16
App	0.85±0.08	0.85±0.11	0.85	0.13±0.10
HC	0.86±0.08	0.89±0.10	0.87	0.12±0.10
CNN-f7	0.79±0.11	0.78±0.13	0.79	0.25±0.20
CNN-c6	0.82±0.08	0.79±0.17	0.80	0.15±0.15
HC+CNN-f7	0.87±0.06	0.84±0.13	0.85	0.14±0.14
HC+CNN-c6	0.89±0.05	0.85±0.11	0.87	0.12±0.11
r-CNN	**0.90±0.06**	**0.89±0.09**	**0.89**	**0.10±0.09**

5.5.2 Evaluation on Customized Expression Dataset

In order to evaluate our framework on customized facial expression task, we collect a customized expression dataset using the following procedure. We first ask the subject to define 5 to 10 expressions they would like to act, where the expressions could be arbitrary; they do not necessarily belong to the six canonical expressions and the expressions can be intensive or subtle. For each customized expression, we record a short video (1 to 2 seconds) for training. Then the subject is asked to record a 2-minute test video by performing their customized expressions in arbitrary order repeatedly for 3 to 5 times. In total, we collected ten testing videos.

Comparing different features.

We first evaluate different subsets and variants of the proposed feature set on the collected dataset with the same classification method in Section 5.4.1. We summarize the comparison results in Table. 5.2 in terms of precision, recall, F1 score and correction ratio (C-Ratio). The

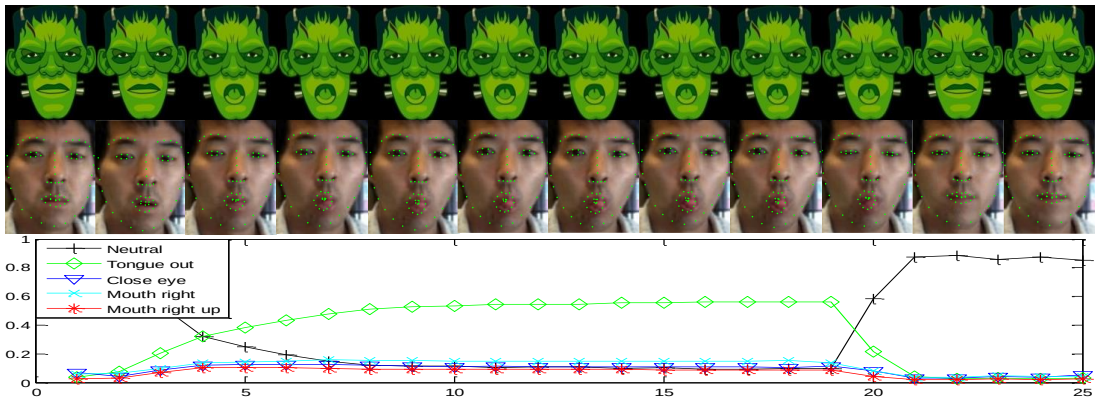
Table 5.3: Precision/Recall, F1 score and correction ratio (C-Ratio) comparison on kNN, ensemble of SVMs (eSVM), HMM with observation from kNN (HMM-kNN) and HMM with observation from ensemble of SVMs (HMM-eSVM). The features for all classifiers are the r-CNN feature.

Classifier	Precision	Recall	F1 Score	C-Ratio
kNN	0.85±0.08	0.81±0.19	0.83	0.17±0.14
eSVM	0.86±0.13	0.81±0.15	0.83	0.11±0.09
HMM-kNN	0.86±0.08	0.89±0.10	0.87	0.13±0.11
HMM-eSVM	**0.90±0.06**	**0.89±0.09**	**0.89**	**0.10±0.09**

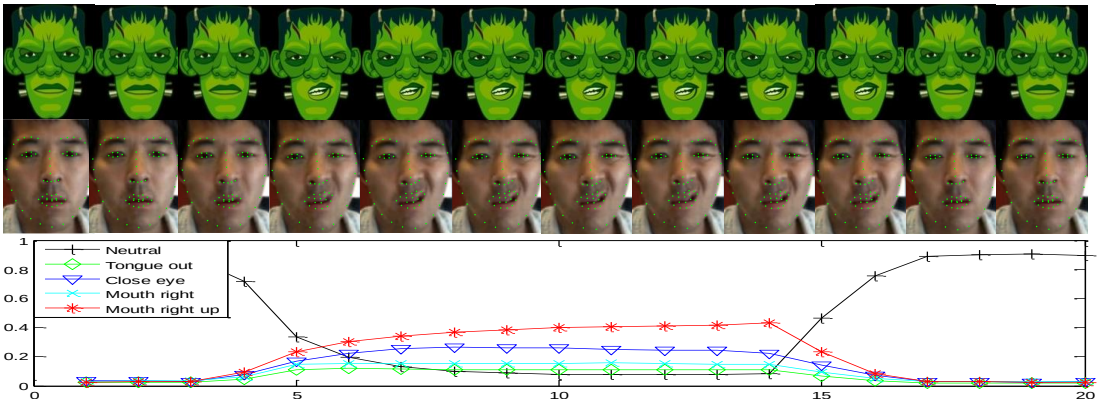
correction ratio is defined as the number of incorrect detected expression intervals over the number of groundtruth intervals that fail to yield a higher-than-threshold overlap with a groundtruth expression interval. For each metric, we show the mean and standard deviation across the test dataset. Note that our handcrafted feature achieves higher precision and recall compared to the CNN feature, which achieves almost perfect results on CK+ and MMI. This suggests that CNN training is overfitted to the canonical facial expressions and thus generalizes poorly to the other customized expressions. This is evident as c6 layer feature is much better than f7 layer feature, where the latter is more tuned for recognizing the six canonical expressions. Simply combining handcrafted feature with CNN feature, however, does not improve the performance (similar F1 scores and correction ratios). In contrast, with our r-CNN structure, we can learn features that are complementary to handcrafted features and the fused feature outperforms both.

Comparing different classifiers.

To justify our classifier choice, we compare it to ensemble of SVMs (denoted as eSVM) as well as a baseline classifier k-nearest neighbor (kNN). We list the comparison results in Table 5.3 for all different combinations of these techniques (eSVM, kNN, HMM-kNN and HMM-eSVM) using r-CNN features. While kNN and eSVM show little difference in Precision/Recall, eSVM is better in the correction ratio, which indicates that e-SVM predicts better expression occurrences than kNN does. After combining with HMM, significant improvements are achieved



(a)



(b)

Figure 5.5: A cutout character animation generated by our system based on a test facial performance. The output probabilities of the trained expressions (neutral, mouth right, close eye, tongue out and mouth right up) for each selected frame are plotted in colored lines.

both from kNN to HMM-kNN and from eSVM to HMM-eSVM (more than 4% in terms of F1 score). These prove that the HMM play a positive role in boosting the performance by incorporating the temporal coherence prior. A sample sequence of customized expression recognition for animation is shown in Figure 5.5.

5.6 Discussions

While the canonical expressions among different people exhibit large degrees of consistencies, customized expressions can have very large inter-personal variations (Figure 5.1). This presents

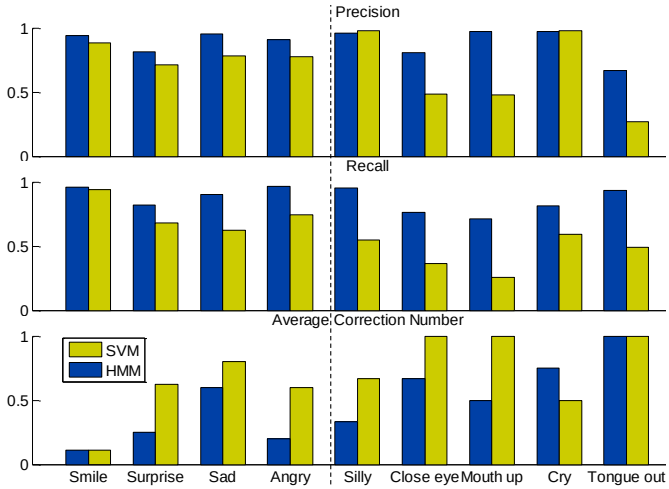


Figure 5.6: Single expression comparison of canonical expressions and customized expressions. The precision, recall and average correction number are listed among smile, surprise, sad, angry, silly, close eye, mouth up, cry and tongue out 9 expressions. The first 4 are canonical expressions and the last 5 after the vertical dash line are the customized ones.

a big challenge for collecting labeled data for a large number of expression categories in order to learn features that are more effective to arbitrary expression recognition, even for targeting user-specific customized expression recognition. In our collected dataset, there are user expressions that belong to both the canonical categories, as we did not constrain the user on what expressions to perform, and the customized or even unnameable expressions. We select the most common canonical expressions from the video dataset, including smile, surprise, sad and angry, and some recognizable user-specific expressions, including silly, close eye, mount lift up, cry and tongue out. We summarize the results on these nine expressions in Figure 5.6, where the left four are canonical expressions and the right five are user defined expressions. Comparing the performance between the two groups, the correction numbers in the left group are notably smaller than the right group, meaning that the canonical expressions can be recognized with better performance even when our training is totally adaptive to specific users. This is not surprising, as our r-CNN features are trained using the canonical expression face dataset. But it indeed suggests that coming up with facial expression dataset with more variety or a better method that has better generalization ability is important for real-life expression recognition and thus in general for HCI.

In this chapter, we provide an initial investigation over the customized expression recognition for performance-driven cutout character animation. We propose several types of features including handcrafted features by combining geometric features derived from facial landmarks and region-based appearance features selected with sparse learning, and Deep CNN features learned with regularization from the handcrafted features. We demonstrate that the proposed features can achieve state-of-the-art performance on conventional canonical expression recognition benchmarks. Then an online ensemble SVM classifiers is proposed to recognize customized expressions from few samples. A sequential HMM online smoothing is applied to further boost the recognition performance by incorporating temporal coherency. Experiments on our collected customized expression dataset demonstrate promising results.

Chapter 6

Application II: Visual-cue Deception Detection

A higher level application, the deception detection, is investigated in this chapter. From the previous chapter, we know that face landmark localization boosts the performance of expression recognition. Actually it helps many other visual applications such as face tracking. The higher level analysis is built on the visual cues extracted by the face localizer including head movement, head posture and facial expression. The significance of different visual cues and their combinations are explored. Higher level synchrony feature is proposed to analyze the interactive response from the interviewer and the interviewee. The synchrony pattern is served to the SVM classifiers for the deception recognition.

6.1 Background

Implicit in all interpersonal interactions is the need to gauge whether an interlocutor is truthful and authentic in his or her presentation of self. The expectation of truthfulness, in fact, is one of the foundations of human discourse [72]. Yet, notwithstanding the importance of this largely outside-of-consciousness assessment process, voluminous research has shown that humans, unaided by technology, are very poor at detecting deception [2, 20, 187]. Average detection accuracy is estimated at 54%, or only slightly above chance, and detection of deception specifically, as opposed to detection of truthfulness, is approximately 47% [20]. Those accuracy estimates have included both lay and professional judges, although some recent evidence points to experts achieving higher accuracy rates under interviewing conditions more characteristic of their usual professional setting and task [23].

One reason cited for humans' poor detection in interpersonal dialogue is that deceivers take advantage of the give-and-take of interaction to adapt to any signs of skepticism in the

interviewer's verbal and nonverbal feedback. Deceivers adjust their messages to make their responses more plausible and their demeanor more credible [21, 193]. That same give-and-take, however, has the potential to offer subtle clues to deception through the disruption of interactional synchrony. Interactional synchrony refers to interaction that is non-random, patterned, and aligned in both timing and configuration of kinesic behavior (i.e., head, face, body and limb movement) with the rhythms and forms of expression in the vocal-verbal stream. It is considered a key marker of interaction involvement, rapport, and mutuality. Synchrony may take the form of simultaneous synchrony, in which two or more people's behaviors mimic or match one another (e.g., similar postures and facial expressions) in the same time frame and behavioral changes occur at the same junctures. This is speaker-listener synchrony. Synchrony may also be concatenous, in which one person's behavior is followed by similar behavior from the next speaker (e.g., each using rapid nodding while speaking). This serial form of synchrony captures speaker-speaker and listener-listener coordination.

Hypothesis: The current investigation explores both simultaneous and concatenous synchrony. It is premised on the possibility that engaging in deception disrupts interactional synchrony and may therefore be a clue to its presence. Practitioners have suggested using rapport-building techniques or interactional synchrony as an effective method for detecting deception: in FBI interviews with terrorists and in police investigations [96, 138, 174]. However, few systematic studies of rapport, coordination, synchrony or reciprocity have examined the effects of synchrony on deception or vice versa [24, 56]. The emphasis typically has been on interviewers using interactional synchrony to promote more verbal disclosures and confessions by interviewees.

Our approach is a novel perspective on the role of synchrony in revealing deception in that we are focusing on the interaction between the interviewee and interviewer rather than only the **interviewee** side of the equation. Deception has been shown to be a cognitively and emotionally taxing activity, especially when the stakes are high and the consequences of being discovered are serious [88, 186]. Interactional synchrony entails a very close linkage among behavioral, physiological and emotional manifestations such that synchrony is positively correlated with rapport and empathy between interlocutors; conversely, incongruent feeling states and behavioral states can disrupt coordination, synchrony and perceived rapport [107]. Mimicry, named

the chameleon effect [8, 28, 52], is a non-conscious tendency to imitate others' verbal and nonverbal behaviors whereas we reserve the term "mirroring" for visually based static behaviors and not dynamic ones. In contrast, interactional synchrony is the smooth meshing in time of the rhythmic, patterned activity of two interlocutors and thus is a better fit for the behaviors we are interested in here because of the temporal component. If the behaviors involved are visual ones and are identical in form (e.g. one person's posture is just like the partner's), the pattern is *mirroring*. If he behaviors instead reflect some temporal, rhythmic and smoothly meshed coordination between interactants, the pattern is called *interactional synchrony*[1] Because deceivers may experience various negative emotional states (or at least emotional states that diverge from those of an interlocutor) and because deceivers may be too preoccupied with constructing plausible verbal responses to attend to or coordinate their nonverbal behaviors with another, we expect interactional synchrony to be attenuated and disrupted when interviewees are deceptive as compared to when they are truthful. Even skilled deceivers may be unable to counter this decrement in interactional enmeshment because conscious efforts to produce synchronous behavior patterns through mirroring another's posture or matching another's degree of animated gesturing and facial expressions may appear "inauthentic" and "off" [71]. Deception thus may be one cause of poor interactional synchrony and dissynchrony may be one sign that deception is taking place.

Our hypothesis tested this possibility. Specifically, we expected that interviews with deceivers are less synchronous than interviews with truth tellers. Initial research using manual coding to evaluate synchrony suggested that deception alters the level of synchrony between deceivers and receivers. The modality used for the questioning and the type of questioning also affected the outcome [209]. Our goal is to determine whether computer vision techniques can expand on previous research by automatically detecting synchrony behavior, which can then be used to distinguish truthful from deceptive individuals. Testing this hypothesis required developing the computer vision methods to assess simultaneous and concatenous synchrony. Those methods are a central focus of the current work.

---

[1]This definition of synchrony is differentiated from mimicry, mirroring, and other forms of behavioral adaptation described in [22].

Moderators: Little is known about whether moderator variables alter the patterns of synchrony. Two possible factors were investigated here: the modality of interaction (face-to-face or video-conferencing) and sanctioning of the deception. Video-conferencing is becoming a widespread medium for communication and sanctioning may alter how deception is behaviorally expressed [57]. Few experiments have examined video-conferencing and instead compare face-to-face interactions to those in text-only modalities [70].

In addition, few experiments directly compare the situation where the experimenter has sanctioned the deception to unsanctioned deception [61] and instead tend to focus on one or the other. In many experiments, participants are told by an experimenter to deceive their partner which may result in less nervousness, guilt, and dissynchrony. In other experiments, participants are allowed to choose whether or not to lie, which results in a lack of random assignment, such that only confident or skilled deceivers may choose to deceive.

Our experiment examines both of these moderators, modality and sanctioning. We asked the following research questions: (1): Is the synchrony between interviewer and interviewee affected by the modality they are using (face-to-face or video-conferencing)? (2): Does the sanctioning of the deception (sanctioned or unsanctioned by experimenter) affect the synchrony between interviewer and interviewee?

Method: In overview, testing data were derived from a cheating experiment in which some subjects cheated during a trivia game with a confederate and some did not, but all were encouraged to appear as credible as possible when interviewed about the game by expert interviewers from the Department of Defense [57]. Thus cheaters were expected to be deceptive and non-cheaters, to be truthful. This kind of deception has been considered high stakes by other researchers [108] and the subjects in our study were reminded that they were in violation of the University's honor code during their interviews. They knew they could face disciplinary action for this act and thus, were under pressure to evade detection. All interviews were videotaped. Separate cameras captured the interviewer and the interviewee and the time-aligned videos were rendered in split screen form. Modality and sanctioning were experimentally manipulated such that some participants were interviewed face-to-face and others were interviewed via Skype. Some were told that the experimenters were aware of their cheating but that they were to deny it to the interviewer (sanctioned version) whereas others received no such explicit

approval of their cheating (unsanctioned cheating).

The videotaped interviews were analyzed using computer vision methods for automated analysis. The autonomous tracking of interactional synchrony cues are proposed to deal with the cases where manual codings are not available. Also, as video is increasingly available (e.g., video data through the internet), manual coding would not be a suitable method for synchrony tracking because coding is time- and labor-intensive. Autonomous nonverbal cues extraction significantly improves such situations. Moreover, it is difficult for a person to track synchrony across multiple cues (e.g., shakes, nods, smiles, and gaze) in real time. Thus, the main considerations of proposing autonomous strategies to code the synchrony feature are to (1) save the coding process time and labor, especially when the applications are large-scale, e.g., web based video dataset; (2) improve the deception detection accuracy by investigating different nonverbal cues, its fusion and feature selection methods; (3) investigate the factors which may influence the synchrony feature. For example, the modality difference, Computer-Mediated Communication (CMC) vs. Face to Face (FtF), deceiver sanctioning, the type of nonverbal cues, the confessor groups, etc. Considering the computer vision algorithms, Constrained Local Models are used for detailed face tracking and head gestures [42, 157]. Our focus in this article is on the automated tracking of head gestures and expressions of both the subject and the interviewer, extracting normalized meaningful synchrony features and learning classification models for deception recognition.

6.2 Relevant Work

6.2.1 Deception Detection

Deception is defined as a message knowingly transmitted to create a false impression or knowledge on the part of the receiver [21, 185]. Early research focused on cardiorespiratory measures to detect evidence of lies in the form of the polygraph [104, 127]. In recent years, scholars have also added other physiological data such as neurological activity to identify liars [103].

The other main approach of detecting deception is to investigate nonverbal and verbal behavioral indicators of deception. For instance, there are many speech cues, such as pauses,

voice pitch, interruptions, delay of response and response length that are associated with deceit [184, 188, 189]. Verbal content indicators have included such features as the amount of detail, logical inconsistencies, spontaneous corrections, or other cues [50, 189]. Nonverbal behaviors that have been examined for indications of honesty or mendacity, have included eye contact, blinking, head movement, posture changes, gestures and leg movements [15, 26, 81, 145, 148, 173, 182]. Two ways in which these indicators reveal veracity is through signaling arousal and emotional states: "the emotional arousal hypothesis suggests that deception produces various emotional states which may influence nonverbal signals" [185]. Indicators may also reflect other processes such as cognitive load. Understudied is the extent to which observed behaviors reflect the social interaction between interlocutors rather than internal states of senders. This latter aspect motivated the current study.

Measurement of nonverbal behavior can be manual, non-computational methods or automatic computational methods [49]. Early research relied on trained observers' rating, counting or timing behaviors, which is the behavioral coding method [100]. This type of method is tedious and requires significant investment of labor and time. Moreover, there is no fixed rule for those methods to segment, annotate and label observations, which may lead to confusion. Hence, researchers have turned to automatic techniques to measure individual behaviors and to assess the degree of similarity between the dynamic nonverbal behaviors of dyadic partners. These methods rely on motion-tracking devices, image processing and video-based computer algorithms [26, 148, 182].

6.2.2 Face Tracking

Face tracking is a fundamental problem in computer vision. The face is a non-rigid and pose-variant object, which increases the difficulty in tracking. Moreover, illumination, facial expression and occlusion are other factors that make the problem even harder.

There are two types of methods dealing with face tracking problems. One is to extract local features and use standard trackers to trace the variation of the features. The other is to directly approximate the shift between the consecutive frames of face images where the local features are the same. In extracting features, the Active Shape Model [40] is one of the most successful methods so far. It sets up a series of landmarks which capture the face profile. Those

landmarks extract gradient or pixel value and do a local search to find the proper locations along the face profile. A linear shape space is trained to constrain those landmarks in a face shape. The Active Appearance Model [35] is another well-known algorithm to capture face profile. Besides the shape landmarks, it imports face appearance and attempts to minimize the error between the reference appearance and the searched appearance. Nevertheless, the global shape or appearance may encounter problems of local variations. The Constrained Local Model developed by Cristinacce and Cootes [42] represents a face as a combination of shape and local feature templates. It is fitted by optimizing the shape parameters to match the image's local appearances to the templates. Improved methods are proposed based on those three above models [39, 41, 136, 206].

6.2.3 Head Movement Detection and Facial Expression Recognition

Detection of head gestures has a long tradition including work by Kapoor and Picard [94] who proposed a method to recognize head nods and head shakes based on two Hidden Markov Models trained and tested on a two dimensional dataset from an eye gaze tracker. Kawato et. al developed a method using "between eyes" templates [97]. Recently, sequential analysis tools have become more and more important in gesture recognition. For instance, a Conditional Random Field is imported in Quattoni et. al's work [142]. They model the head motion as a temporal sequence and establish a graphical structure to analyze the behavior. Moreover, due to the complexity of real problems, Hidden-state CRFs and Dynamic CRFs are gradually employed in analyzing time series data [137, 166].

Facial expression recognition is another important topic in the communities of computer vision. The previous work can be categorized into two classes, image based methods and video based methods [60, 140]. Image based methods neglect the dynamic characteristics during an expression sequence. However, video-based methods deal with the dynamics to classify expression. Many experiments have demonstrated the importance of the facial dynamics [16, 33]. In video based expression recognition, temporal segmentation of expression action events and the representation of the dynamics are two major problems. Torre [170] used condensation to trace the local appearance dynamics and Cohn [34] applied key point tracker to represent the dynamics.

Figure 6.1: Sample snapshots from tracked facial data showing a subject (left) and an interviewer (right). Red dots represent tracked facial landmarks (eyes, eyebrows, etc.), while ellipse in top left corner depicts the estimated 3D head pose of the subject; top right corners show the detected expressions and head gestures for subject and interviewer.

6.2.4 Interactional Synchrony

Synchrony is the dynamic and reciprocal adaptation of behaviors between interactive partners [49]. It is reflected by the relevant features of the interactive motion, i.e. head motion, facial expression, etc. Although using synchrony to evaluate the deceptiveness of statements is a rather new technique, an initial investigation using human coding has suggested that disruptions in synchrony can distinguish between deceivers and truthtellers. It is perhaps due to their cognitive load or their violation of conversational norms [56].

To evaluate synchrony, correlation is one of the mainstream methods [3, 12, 19]. After extracting the motion or expression features, a time-lagged cross correlation is applied over the two sequential single-channel features with certain time slot window and thus the response indicates the degree of synchrony. Another strategy is to use recurrence analysis [146, 162]. For a given two time series, every vector with delay t is compared with every vector in the second time series. A recurrence matrix is created. By thresholding the distance between the two vectors, the degree of synchrony is provided. The third type of methods are spectral methods. Some methods focus on dealing with the two time series' relative phase [139, 147]. Others focus on measuring the overlap between the movement frequencies of the two interactants [48,

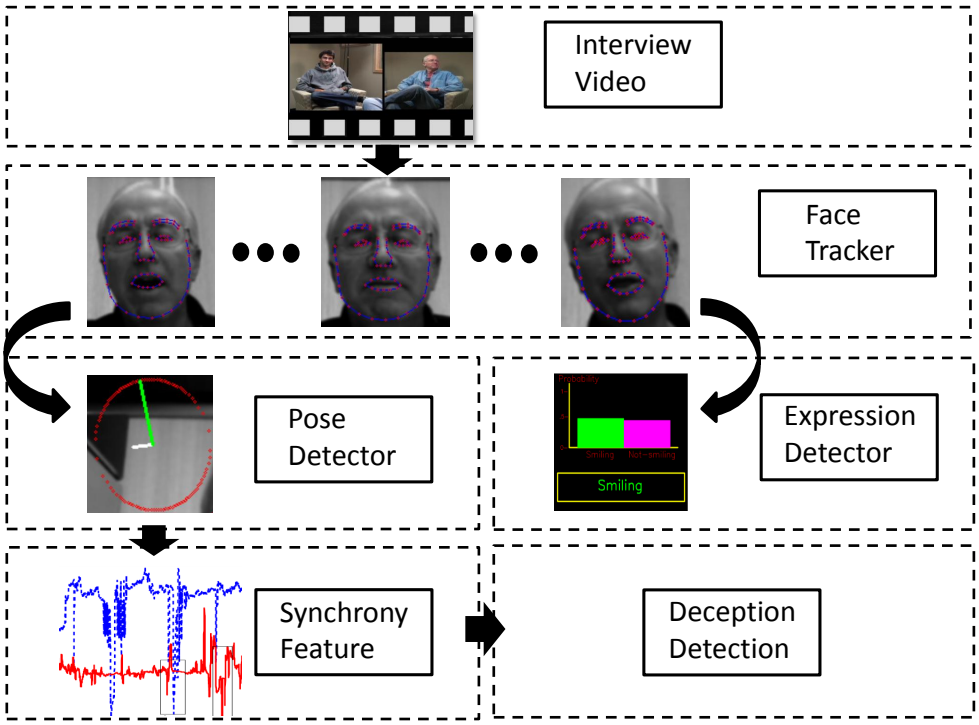


Figure 6.2: The workflow of deception detection framework consisting face tracker, head pose detector, expression detector, synchrony feature extractor and deception detection classifier.

145].

6.3 System Overview

We have developed a framework that is capable of analyzing synchrony and detecting deception. The framework includes tracking facial movements module, facial expression recognition module and head movement detection module. The sample interface is shown in Figure 6.1. Based on the single-channel features extracted by those modules, i.e., head nodding, head shaking, smile, etc, we designed the temporal causality like strategy to generate synchrony features. Using the higher level features, a data-dependent learning based classifier is designed to differentiate deceptive groups from truthful groups. The whole flowchart is shown in Figure 6.2. Each of the modules is illustrated in detail in the following subsections.

6.4 Multi-pose Face Tracking

Face tracking is a challenging problem. The shapes of faces change dramatically with various identities, poses and expressions. Furthermore, poor lighting conditions may cause a low contrast image or cast shadows on faces, which adversely affect the performance of the tracking system. We have developed a robust face tracker [206, 207] based on the pose-robust OPM initialization and hierarchical part-based regression together.

The local profile models capture the local appearances around each landmark point and are used for selecting the best candidate landmark positions. We adopt a logistic regression based learning method [208] to obtain weights w, b for template detectors.

$$p(v_i = 1|s_i, I) = \frac{1}{1 + exp(wf + b)}, f = \Phi(I, s_i) \tag{6.1}$$

In (6.1), we formulate the possibility of locating a candidate landmark position as $p(v_i = 1|s_i, I)$ knowing the facial image I, the i^{th} landmark candidate position s_i. Here v_i is a random variable indicating whether the candidate position is in the positive location. f is feature vector extracted by Φ, which is the feature extraction strategy. In general, Histogram of Oriented Gradient (HOG) [45] and Local Binary Pattern (LBP) [190] are widely used feature extractors in appearance detection. To locate the facial features in varying poses, as stated in Chapter 2, the multi-view optimized part mixtures initialize the facial key points under many different pose assumptions.

Fitting shapes in consecutive frames is conveniently considered as given the previous alignment result as initialization, fitting the face shape in the current frame, which could directly utilize the algorithms proposed in the previous chapters, i.e. a hierarchical cascaded regression stated in Chapter 4 fast and accurately localize the key points given the well-aligned previous tracking result. Moreover, in chapter 4 we have proposed a graphical model based evaluation strategy (4.8). This scheme allows us to have a measure of tracking success (confidence) for frame, so we can have early detection and correction when drifting from the target.

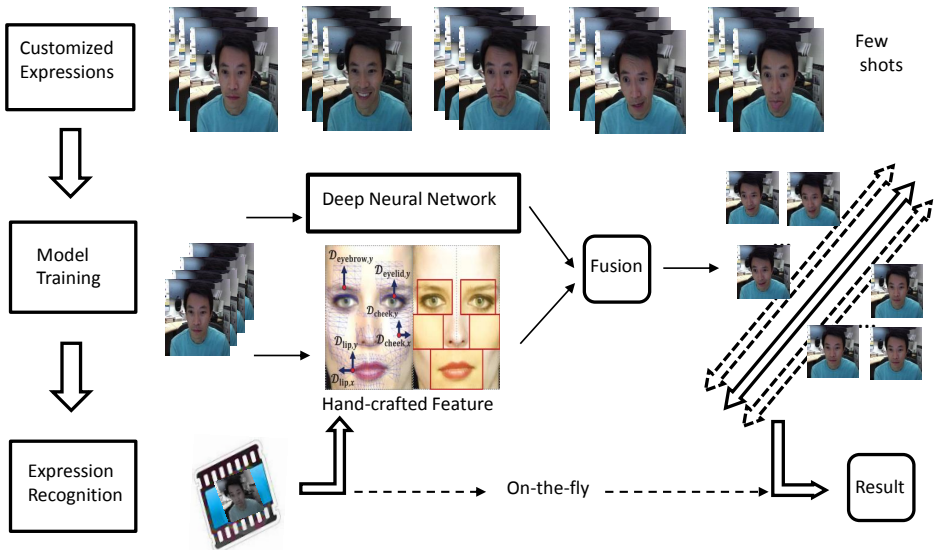


Figure 6.3: The flowchart of the user-defined expression recognition.

6.5 Head Movement and Facial Expression

From the landmark positions in each frame, we are able to estimate the 3D poses (pitch, yaw, and tilt) and detect the relevant head gestures (head shaking, nodding, head towards front and head turning away). To estimate the face pose, a linear Partial Least Squares (PLS) regression model [79] is built for all linear regions in the shape manifold. This regression model takes the x and y coordinates of the 79 landmarks as input, and predicts the pitch, yaw and tilt angles.

$$A = XB + F^{*} \tag{6.2}$$

Matrix A represents the head pose parameters pitch, yaw and tilt. Matrix X is concatenated by all the landmarks' coordinates. B is the mapping matrix which we pursue and F^{*} is the residual matrix indicating the variation. The head nod is a gesture in which the head is tilted in alternating up and down, and head shake means that the head is turned left and right, repeatedly in quick succession. Therefore, by differentiating the pitch value and yaw angles in each frame, we are able to detect the head nod and shake, respectively.

A facial expression classifier is also built to detect facial expressions such as smile in chapter 5. The framework is defined under user-defined expressions, the range of which is much more than the traditional six expressions. Figure 6.3 shows the flowchart of our user-specific

expression recognition. From the structure, our framework not only handles user-defined expressions but could also naturally facilitate to the traditional expression recognition task. For the offline training stage, the customized images are prepared. Then the Convolutional Neural Network features as well as handcrafted features are extracted and fused. An ensemble of one-vs.-all SVM classifiers is applied for the frame-based expression recognition. An HMM filter is further applied to smooth the temporal sequence output. The online testing stage is much more straight forward. The input test frame is firstly processed for the feature extraction. Then the recognition result is provided by the SVM-HMM predictor.

6.6 Interactional Synchrony

The interaction of people in a dialogue is directly indicated by head gestures and facial expression. However, the inner property of such interaction should be depicted by more profound features. The subtle and significant way people influence each other can be seen through their nonverbal synchrony. Synchrony refers to similarity in rhythmic qualities and enmeshing or coordination of the behavioral patterns of both parties in an interaction [24]. Such synchrony can either be simultaneous or concatenous. In Dunbar et al.'s work [56], synchrony can be indicated by nodding or shaking, facial mirroring, etc. Providing pairs of interview videos, we can capture head nodding or shaking and facial expressions (especially smiling) in videos by our proposed facial tracking and facial expression detection methods. Based on such single-channel features, we intend to check the simultaneous response from both two people in an interview.

Individual feature vectors of two interview videos from one interviewer and one interviewee can be viewed as two corresponding data sequences. We can get large response while doing correlation over two sequences if the two sequences have similar magnitude at the same position, which may measure the simultaneous response. If two sequences have similar magnitude at different position, we can take a time sliding window to compensate the time delay and then calculate their correlation. Cross correlation is a standard method of estimating the degree to which two sequences are correlated. The cross correlation of two signals with a latency d is

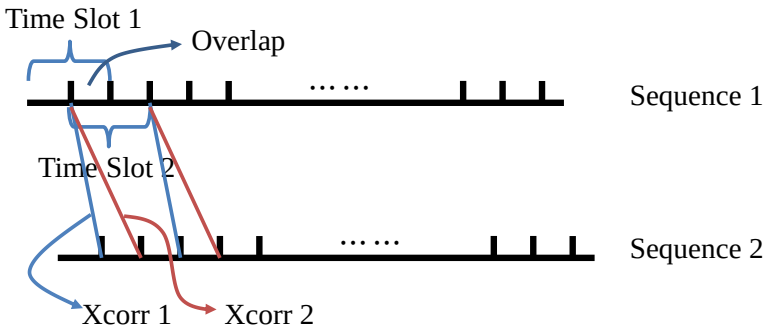


Figure 6.4: sequences cross correlation scheme

defined as:

$$C(d) = \frac{\sum_i (z_i - \bar{z})(y_{i-d} - \bar{y})}{\sqrt{\sum_i (z_i - \bar{z})^2} \sqrt{\sum_i (y_{i-d} - \bar{y})^2}} \tag{6.3}$$

where z_i and y_i are the ith element of sequence z and sequence y, $\bar{z}$ and $\bar{y}$ are the mean value of sequence z and sequence y.

In order to accommodate concatenous synchrony, we divide two sequences into overlapped time slots, as shown in Figure 6.4. The two sequences are required to have the same length. Then we equally divide each sequence into m time slots. Starting from either of the sequences at current time slot, we take $[-t, t]$ time slots to calculate their correlation with the current time slot. The largest cross correlation response is chosen as the current time slot's feature value. We repeat such procedure for every time slot in a sequence. As a result we obtain a cross correlation based synchrony feature vector with length m.

6.7 Feature Selection and Deception Detection

Once we obtained effective single-channel feature, e.g. head-nodding, smiling, looking forward, a synchrony feature is formed by combining those single-channel features. The single-channel features are normalized, and then a weighted vector concatenation is applied to generalize a uniform feature vector. The weights for different single-channel features can either be tuned by k-fold cross validation or empirical setting. Such combination may not lead to optimal feature representation. Moreover, in the single-channel feature extraction step, noise may be introduced. Inside the combined feature, different feature elements may be correlated.

Algorithm 4 Genetic Algorithm for feature selection.

Input: Initialized random feature selectors
Output: Optimized feature selectors.
repeat
    Crossover on parent selectors
    Mutate on parent selectors
    Test performance on whole population, choose the top N candidates
    set the top N candidates as parent selectors
until accuracy can not increase or iteration number exceeds
set top parent candidates as optimized feature selectors

Therefore, we chose the most effective feature elements out of the original synchrony feature vector to remove noise and redundancy.

This feature selection was achieved by Genetic Algorithm (GA). In this process, we randomly set the initial feature selectors consisting of $0/1$ elements, in which 0 indicates not selecting and 1 means selecting. In each generation, such random feature selectors would crossover and mutate to generate new descendants. Then among the whole population, the algorithm will choose the top candidates which achieves the best performance in the classification task. Such iteration repeats until the accuracy cannot be increased or the maximum iteration limit is reached. Algorithm 4 illustrates this procedure in detail.

Obtaining the effective and precise selected synchrony features, we formulate the deception detection problem as a classification problem. We intend to differentiate the truthful group from the deceptive group, which is a two-class classification problem. Since the training volume is not large and the aim is to minimize the misclassification rate, we maximize the margin of those two groups and thus choose support vector machine (SVM) [181] as our first layer classifier. In addition, inside the deceptive group, it can be further classified as the sanctioned cheating group in which people are allowed to lie before they take the examination dialogue, and the unsanctioned cheating group in which people are not provided with any information about deceiving. Similarly, we adopt SVM as our second layer classifier. In such way, we designed a multi-layer SVM classifier for the deception detection task.

As a summary, in this section we have introduced a face tracking module to trace the head movement. Based on the head movement, we further set up the head pose detection module and facial expression detection module to obtain gesture and emotion features. Such single-channel features are not efficient enough for deception detection. We built a causal relationship based synchrony feature as higher level feature. After feature selection, we designed a two-layer SVM classifier to achieve the deception detection task.

6.8 Experiments

In this section, we first introduce the experimental protocols, and then show the tracking, gesture and expression detection results as the input for the deception detection. Further, different single-channel features are examined and the feature selection algorithm is investigated to improve the recognition accuracy. Finally two-class and three-class classification output is illustrated, which reveals the advantages of our framework for analyzing synchrony and hence discriminating truth.

6.8.1 Experimental Settings

The analysis began with creating a database of 242 videotaped interviews of 121 interviewer-interviewee pairs. Interviewees were students who participated in a trivia game and in some cases were induced by a confederate to cheat. All participants were then interviewed by expert interviewers about the game interaction and whether they cheated during the game. Approximately half of the interviews were conducted over Skype and the other half were conducted face-to-face to produce two modality conditions, CMC and FtF. Since a few of the pairs are incompletely recorded, we selected 100 out of 121 pairs of videos as our training and testing data. These video pairs vary from 4500 frames to 15000 frames. Although video pairs' lengths are different, we ensure inside each video pair, the interviewer sequence and interviewee sequence keep the same length, which allows using a fixed number of time slots to analyze the video sequences.

To generate the synchrony feature, we set up certain width of time window. Each window has the same size, which are 180 frames comprised of 6 seconds prior to the current frame

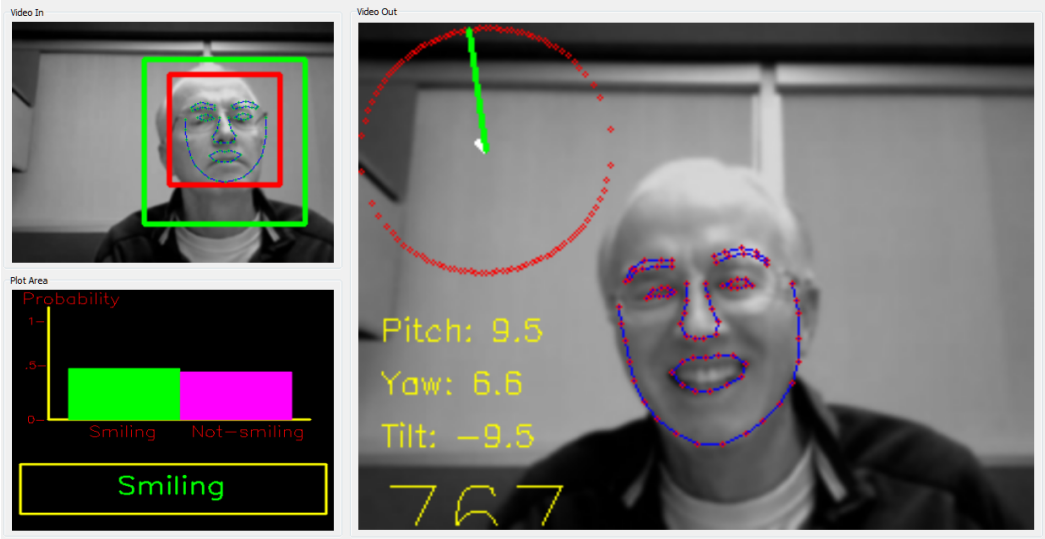


Figure 6.5: Face tracking and expression, head pose estimation visual result. Left Top: Initial face detection and landmark initialization. Left Bottom: Score plot of expression (smile or not smile) recognition. Right: Facial landmark tracking result and head pose estimation(depicted by pitch, yaw and tilt).

and 6 seconds following the current frame. Such window is tunable in practice. However, since too large window size may mediate the synchrony pattern and too small window size cannot capture significant pattern in synchrony, such window size is tuned according to the experimental performance. In our case, we modified the window size, generated the synchrony feature, and tested it on randomly chosen video segments. If the performance improved, we modified the the window size again. This process was repeated until the window size reached it optimum value. As our video pairs have length above 4500 frames, we set the window size to overlap a half in consecutive manner so that the procedure lasts for the entire sequence and generate a holistic synchrony feature.

6.8.2 Tracking, Gesture and Expression Detection

In the synchrony detection step, we extracted head nodding, head shaking, smiling and head direction(looking forward or looking away). The visual result is shown in Figure 6.5. More tracking results are shown in Figure 6.6. Head motion can be detected by analyzing pitch, yaw

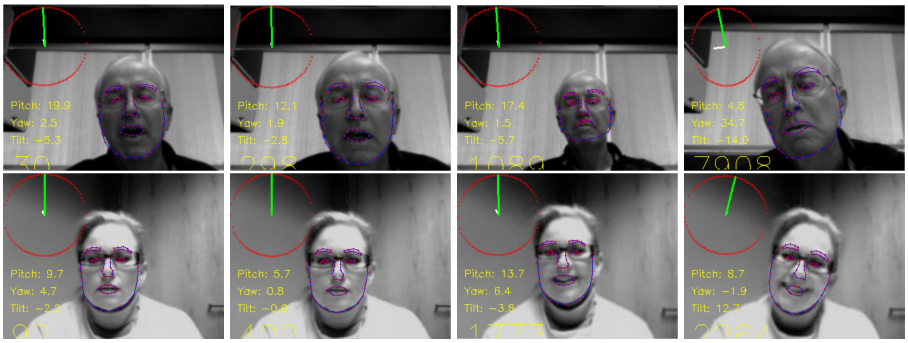


Figure 6.6: More visual results of the multi-pose tracking system. The first row are results from one interviewer. The second row are results from the corresponding interviewee.

and tilt as demonstrated in section 2.2. Pitch depicts nodding action and yaw reveals shaking action. The nodding and shaking synchrony patterns are shown in Figure 6.7. Based on such single-channel features, we further combine the interviewer's feature vector with interviewee's feature vector to form higher level features. A correlation-based method is adopted to identify synchrony. Then a two-layer classification scheme is designed to separate 3 classes (truthful group, sanctioned cheating group and unsanctioned cheating group). We first classify the truthful group from the cheating group using non-linear SVM classifier, which is a two class classification task. Then based on the result of the first step, we continue to classify the cheating group into sanctioned cheating group and unsanctioned cheating group by another non-linear SVM classifier. During the feature selection part, at each step we separately train a feature selector using Genetic Algorithms. The feature selector is an efficient way to promote the performance in recognition task because the raw features may have noise or redundancy.

6.8.3 Evaluation of Synchrony Features

Before using all features, different types of features should be investigated to find the effective ones for classification. Our strategy is to leave each single feature out of the whole feature vector and then test the recognition accuracy. We also identify the single feature's recognition accuracy and visualize the feature vector in plots to see the separability of each of the four types of features (head nodding, head shaking, smiling or not smiling and look forward or look away). During this step, we examine the average precision of classifying three classes, i.e., truthful, sanctioned cheating and unsanctioned cheating classes, to evaluate each feature. Table 1 shows

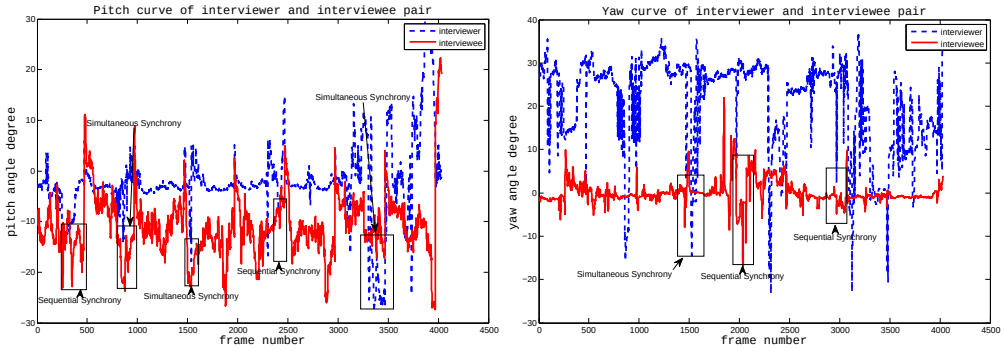


Figure 6.7: Synchrony pattern illustration in pitch and yaw angle curves. Left: Pitch curve of a pair of interviewer and interviewee; Right: Yaw curve of a pair of interviewer and interviewee. X axis stands for frame number and y axis represents angle degree (pitch or yaw).

the average precision of different feature combinations over the three-class classification.

Table 6.1 shows that in CMC, when the feature "Nod" or "Shake" is excluded from the whole feature vector, the performance is higher than the rest. When the feature "Smile" or "Look forward" is excluded, the performance drops. For FtF, the trend is opposite: "Nod" and "Shake" are more significant in classification. When testing each single feature's accuracy, it appears that "Look forward" and "Smile" are more accurate than "Nod" and "Shake" for CMC. And again, for FtF modality, "Nod" and "Shake" achieves higher accuracy than "Smile", where "Look forward" is not applicable in the FtF dataset. The reason is for FtF data, interviewer and interviewee sit face to face. The look-forward feature should be defined by their local head coordinates. But only one camera was capturing the scene, only allowing the global camera

Table 6.1: Detection accuracy evaluation of four features, "Nod", "Shake", "Smile" and "Look forward". "All but one" means that all features are used except the one of that column. "Single" means using only the feature of that column.

		Nod	Shake	Smile	Look forward
CMC	All but one	**0.422**	**0.437**	0.356	0.311
	Single	0.35	0.33	**0.38**	**0.43**
FtF	All but one	0.442	0.473	**0.554**	-
	Single	**0.513**	**0.464**	0.435	-

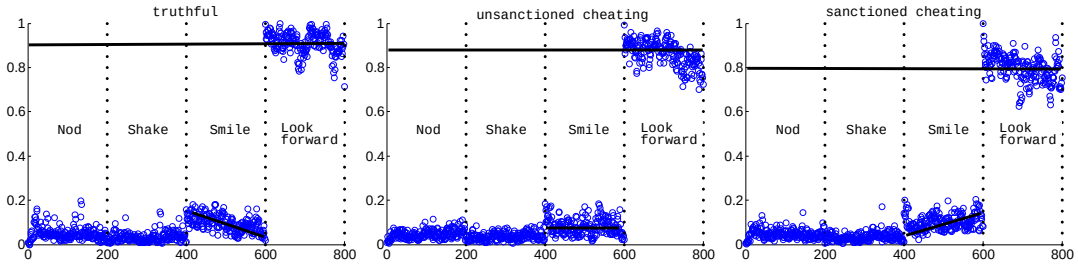


Figure 6.8: Mean feature vector patterns of three groups. Left: truthful group's mean feature vector pattern. Middle: unsanctioned cheating group's mean feature vector. Right: sanctioned cheating group's mean feature vector.

coordinate. Thus the frontal face cannot be obtained by the camera coordinate.

In Figure 6.8, the vertical dotted lines separate the plot into 4 regions representing the four separate features. The first column indicates the feature "Nod", the second one is the feature "Shake", the third is the feature "Smile" and the last one is the feature "Look forward". We plot the average feature vector of each group in the subplots. The feature vector is 800 numbers long, of which each region is with length 200 numbers. With the black line showing the trends in the figure, we see that in region three, the pattern of the feature vector is obviously different. In the subplot for the truthful condition, it is going down; in the subplot for unsanctioned cheating, it is flat; in subplot of sanctioned cheating, it is going up. In region four, the average value of those numbers is going down from above 0.9 to less than 0.9 until around 0.8.

In practice, we have applied nod, shake, smile, looking forward and hit-miss rate proposed in [134] for the single-channel features. Besides the "all-but-one" evaluation, we would like to investigate each single-channel feature comparing to features proposed in [134] and our fused synchrony feature. The comparison may reveal whether the proposed synchrony feature is a valid one for deception detection and whether it is more advantageous other than features proposed in [134] or single-channel features.

The feature proposed in [134] is mainly the hit-miss rate. The problem in that paper is to detect deception from truth, which is a two-class classification problem. The proposed method deals with three-class categorization problem, which is to classify truthful, sanctioned cheating and unsanctioned cheating groups.

In Table 6.2, we listed the comparison of true positive rate, false positive rate and precision.

Table 6.2: The accuracy of proposed synchrony feature, feature in [134] and each single channel feature. “TP” and “FP” stand for true positive and false positive.

		TP	FP	Precision	Recall
CMC	Proposed	**0.667**	**0.167**	**0.668**	**0.667**
	[134]	0.420	0.289	0.429	0.420
	Nod	0.370	0.314	0.374	0.370
	Shake	0.370	0.316	0.360	0.370
	Smile	0.437	0.282	0.432	0.437
	Look forward	0.462	0.261	0.478	0.462
FtF	Proposed	**0.651**	**0.187**	**0.668**	**0.651**
	[134]	0.442	0.280	0.442	0.442
	Nod	0.372	0.338	0.394	0.372
	Shake	0.395	0.319	0.395	0.395
	Smile	0.349	0.349	0.339	0.349

For the CMC database, the fusion feature achieves largest true positive and precision rate, with smallest false positive rate. In contrast, the feature proposed in [134] shows limited advantages over other single-channel features. The reason may be that our proposed fusion feature includes the feature proposed in [134] and we applied efficient feature selection method to improve the accuracy. The same observation appears to the FtF database, which consistently reveals that the synchrony feature is a more efficient higher level feature in detecting deception.

6.8.4 Evaluation of Feature Selection

We applied Genetic Algorithm (GA) for feature selection in our framework. In GAs, there are several parameters to influence the recognition rate. Basically, they are the length of selected elements, crossover segment number, mutation ratio, the amount of population for each generation, etc. The length of selected feature elements are decided by other factors such as crossover segment number, mutation ratio, the amount of population, etc. Since all other factors are finally reflected to the length of elements, for simplicity, we investigate how the recognition rate

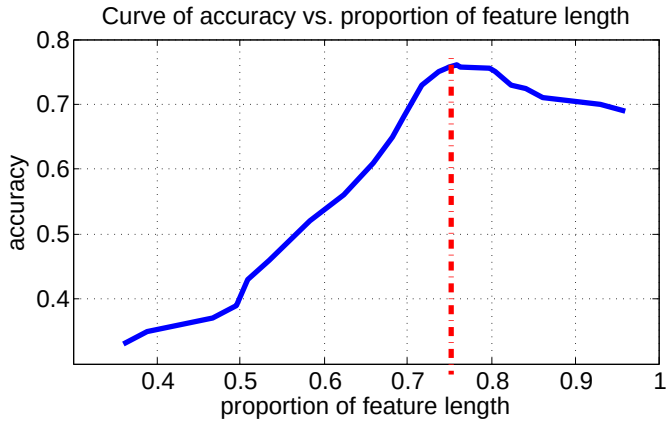


Figure 6.9: The relationship between proportion of feature length and classification accuracy.

varies with the selected feature length.

To experiment with the parameters in the Genetic Algorithm, we set the crossover segment number to vary from 2 to 100, the crossover time varied from 1 to 3 and the mutation ratio varied from 0.005 to 0.05. For each set of parameters, we independently ran the Genetic Algorithm 5 times. Each time we got the recognition rate together with the selected feature length. When all the sets of parameters were listed, we analyzed the relationship between recognition rate and the feature length. Then the curve is plotted in Figure 6.9.

In the plot, the horizontal axis means the proportion of the original feature length and the vertical axis stands for the recognition rate. From Figure 6.9, we observe that as the proportion decreases from 1 to 0.8, the recognition rate is increasing until it reaches 0.75, when the accuracy reaches maximum. And as the proportion goes down from 0.7 to 0.4, the accuracy decreases almost monotonically. We think that the feature vector with full length may contain redundancy and noise. After feature selection, the useful feature elements are selected, by which the redundancy and noise are removed. As a result, the accuracy is expected to increase. When the feature length is continuously shortened, some of the useful information in the feature vector may be eliminated. In this case, the accuracy may decrease as the figure shows. As a consequence, in this experiment, 0.75 is considered optimal proportion of feature length to achieve the best accuracy.

With optimal chosen parameters, comparing the accuracy of features with and without feature selection, for two-class classification, the original accuracy is 54.2% and the accuracy with

Table 6.3: The accuracy of classifying the truthful and deceptive cases. "TP" and "FP" stand for true positive and false positive.

		TP	FP	Precision	Recall
CMC	Truthful	0.667	0.200	**0.625**	0.667
	Deceptive	0.800	0.333	**0.828**	0.800
	Average	0.734	0.267	**0.727**	0.734
FtF	Truthful	0.750	0.259	**0.632**	0.750
	Deceptive	0.741	0.250	**0.833**	0.741
	Average	0.744	0.253	**0.758**	0.744

feature selection is 74.2%; for three-class classification, the original accuracy is 46.9% while the accuracy after feature selection is 66.8%. The accuracy with feature selection is more than 10% higher than that without feature selection, which indicates that the feature selection is a key step of accuracy improvement.

6.8.5 Evaluation of the Classification Accuracy

Two-Class Classification: Even with feature selection's promotion, it is still possible to improve the accuracy since proper classifier design could enhance performance. The initial three-class classification using non-linear SVM scheme may not be perfect because it contains at least three intersections of misclassification, which are intersections of truthful and unsanctioned cheating groups, truthful and sanctioned cheating groups, unsanctioned cheating and sanctioned cheating groups. Although the problem is to divide the data into truthful, unsanctioned cheating and sanctioned cheating groups, it is at least a two-class' classification problem, which is truthful and cheating groups' classification. We could continue to solve a two-class' classification problem on the unsanctioned cheating and sanctioned cheating groups in the same way. Hence, we get only 2 intersections of misclassification, misclassification of truthful and cheating groups and misclassification of unsanctioned cheating and sanctioned cheating groups, which is expected to decrease the error recognition rate. We set both 15 test samples for truthful group and cheating group. Thus 70 samples are the training samples, 16 in the truthful

Table 6.4: The confusion matrix of classifying truthful and deceptive cases of CMC and FtF modalities.

		Truthful	Deceptive
CMC	Truthful	**10**	5
	Deceptive	6	**24**
FtF	Truthful	**12**	4
	Deceptive	7	**20**

Table 6.5: The accuracy of classifying the truthful, unsanctioned and sanctioned cheating cases. "TP" and "FP" stand for true positive and false positive.

		TP	FP	Precision	Recall
CMC	Truthful	0.667	0.200	**0.625**	0.667
	Unsanctioned	0.600	0.133	**0.692**	0.600
	Sanctioned	0.733	0.167	**0.688**	0.733
	Average	0.667	0.167	**0.668**	0.667
FtF	Truthful	0.750	0.259	**0.632**	0.750
	Unsanctioned	0.538	0.067	**0.778**	0.538
	Sanctioned	0.714	0.172	**0.667**	0.714
	Average	0.651	0.187	**0.668**	0.651

group and 54 in the cheating group. The performance is shown in Table 6.3.

The confusion matrix in Table 6.4 show that for the CMC dataset in the truthful group, 10 samples are correctly classified while 5 are not; in the cheating group, which is the combination of unsanctioned cheating and sanctioned cheating groups, 24 samples are correctly classified and only 6 are not. Table 3 shows the classification accuracy details. In CMC, "truth" precision is 62.5% and "deception" precision is 82.8%, for an overall average of 72.7%. In FtF, the precision values are 63.2% for "truth" and 83.3% for "deception" for an overall precision of 75.8%, which is roughly at the same level as CMC.

Three-Class Classification: After the classification of truthful and cheating groups, based

Table 6.6: The confusion matrix of classifying truthful, unsanctioned and sanctioned cheating cases of CMC and FtF.

		Truthful	Unsanctioned	Sanctioned
CMC	Truthful	**10**	5	3
	Unsanctioned	4	**9**	2
	Sanctioned	2	2	**11**
FtF	Truthful	**11**	1	4
	Unsanctioned	5	**7**	1
	Sanctioned	2	2	**10**

on the cheating categorization result, we continue to classify the cheating group into unsanctioned cheating and sanctioned cheating groups. The classification scheme is the same as first step. However, the training and classification is data-dependent, especially in feature selection and non-linear SVM classifier training.

Table 6.6 shows our final confusion matrices over all the three categories. In each category, the number of correctly recognized samples dominates misclassified numbers. Further Table 6.5 illustrates that the precisions of all classes are above 60%, two of which are approaching 70%. The average accuracy is 66.8%, which is clearly a significant improvement over 47% [20].

6.8.6 Evaluation of confessors in deception detection

In the experiment, some of the interviewees confessed to deception during the interview. Before they confess to the interviewer, the pattern may appear the same as cheating mode. After the confession, they may have felt relieved and then performed in a similar fashion to truth-tellers. The confessor group inside the cheating group may have influenced detection. We aim to determine if the confessors were more synchronous than the non-confessors by evaluating the dataset excluding the confessor group. Comparing to the results in section 3.3, we expect to find the degree of synchrony by including and excluding such confession group.

Table 6.7 reports the confusion matrices of the three-class classification result on both CMC and FtF databases. The diagonal elements of the two matrices dominate all the other elements

Table 6.7: The confusion matrix of classifying truthful, unsanctioned and sanctioned cheating cases of CMC and FtF in confession group excluded condition.

		Truthful	Unsanctioned	Sanctioned
CMC	Truthful	**27**	3	1
	Unsanctioned	10	**18**	2
	Sanctioned	4	6	**7**
FtF	Truthful	**14**	2	0
	Unsanctioned	2	2	1
	Sanctioned	2	1	**3**

which reflect that our classification scheme groups most of the samples correctly. Further, comparing Table 6.8 with Table 6.5, the excluding confessor classification achieves at least as good as the including confessor scheme. Moreover, it shows that in "Truthful" group of both CMC and FtF, the excluding scheme achieves 80.6% accuracy for CMC and 87.5% for FtF, while the including scheme achieves 66.7% for CMC and 68.8% for FtF. Nevertheless, the average precision of excluding confessor cases is slightly higher than including cases in both CMC and FtF datasets.

To test the differences between the confessor group and non-confessor group, we set up Hypothesis-Test experiments both for CMC and FtF databases. For the CMC database, we have 109 valid video pairs and consequently 109 valid synchrony feature vectors. Among the whole dataset, there are 78 non-confessors and 31 confessors. We propose the following hypotheses:

H0: the confessor group has no difference with the non-confessor group.

H1: the confessor group has difference with certain significance level, which is to reject H0.

Our experiment is set up under the Monte-Carlo framework. The first experiment is to randomly divide the 109 feature vectors into two sets, one for training and the other for testing. Generally the training and testing volume is equal. Then we record the test accuracy of a one-time experiment and repeat this experiment 100 times. Each time the training and testing

Table 6.8: The accuracy of classifying the truthful, unsanctioned and sanctioned cheating cases in confession group excluded condition. "TP" and "FP" stand for true positive and false positive.

		TP	FP	Precision	Recall
CMC	Truthful	0.871	0.298	**0.659**	0.871
	Unsanctioned	0.600	0.188	**0.667**	0.600
	Sanctioned	0.412	0.049	**0.700**	0.412
	Average	0.667	0.201	**0.671**	0.667
FtF	Truthful	0.875	0.364	**0.778**	0.875
	Unsanctioned	0.400	0.136	**0.400**	0.400
	Sanctioned	0.500	0.048	**0.750**	0.500
	Average	0.704	0.251	**0.702**	0.704

datasets are partitioned again. Thus, we independently conduct the experiment 100 times and obtain 100 classification accuracy records for the 109 feature vectors. The second experiment was to remove the 31 confessors from the whole 109 feature vector group. With only 78 non-confessors, we conduct the same experiment as before. We independently repeat the training and testing process 100 times. Each time the training and testing datasets are re-partitioned. The final classification accuracy is recorded for comparison.

After the two separate experiments, we obtain two accuracy vectors, each of which is with 100 elements. Then we apply t-test over the two vectors under the null hypothesis with significance level 0.05, which is acceptable for most hypothesis-test experiments. For CMC experiment, the t value is 5.87 and the threshold from t-table is 1.66. Since 5.87 is larger than 1.66, we reject the null hypothesis with probability 0.95. For FtF experiment, the t value is 8.55 and the threshold from t-table is 1.66. Since 8.55 is larger than 1.66, we again reject the null hypothesis with probability 0.95.

The accuracy histogram is visualized for comparison in Figure 6.10. The blue columns represent the overall group accuracy and the red columns stand for non-confessor group accuracy. The vertical axis is the accuracy interval. Each interval calculates the accuracy below that

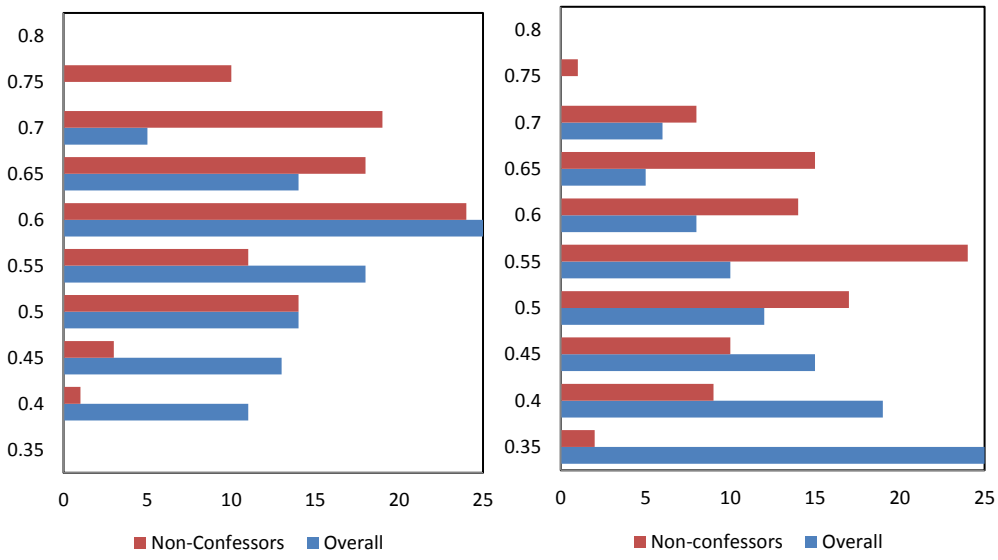


Figure 6.10: Accuracy histogram of overall and non-confessor groups. Left: Accuracy histogram of CMC database. Right: Accuracy histogram of FtF database.

threshold. The horizontal axis means how many times the accuracy appears inside the interval of vertical axis. Clearly, we observe that the distribution for the two groups is different. This supports the idea that the confessor groups undermine classification accuracy.

6.9 Discussions

In this investigation, we hypothesized that the introduction of deception into an interview would disrupt the synchrony of a dyad. We also examined whether the modality of the interview (CMC vs FtF) and the sanctioning of deception by the interviewee would affect the diagnosticity of synchrony. The analysis of the CMC and FtF conditions were proposed in parallel fashion, but the four features (head nod, head shake, smile and look forward) had different significance in the two modality conditions, possibly due to the physical location of the camera or individuals in each. Both in two-class and three-class classification, the performance of CMC and FtF datasets achieved the same degree of accuracy, which suggests that the degree of synchrony was not influenced much by the modality of communication. This finding is consistent with other synchrony research based on manual coding [55]. Nevertheless, from the three-class classification result, the sanctioned cheating group is well separated from the unsanctioned cheating group, which indicates sanctioning is a key factor to influence synchrony

and as a result discriminates unsanctioned cheating from sanctioned cheating. This finding does not contradict findings from synchrony research using manual coding which found the unsanctioned deceivers most distinguishable. We state that sanctioning is a key factor for deception detection but no judgment is made whether sanctioning group or unsanctioning group is easier to detect. Moreover, the manual assessment of synchrony was at a gestalt level, not at the level of detail presented here. Those deceivers who confessed during the interview also influenced the classification process. Once the confessor group is removed, the truth-tellers are much better separated from the deception groups than before.

Automatic methods can often detect events of synchrony which are missed by the human coders for whatever reason. In particular, we found that while the human coders in the Dunbar et al. study [56] would label a given video as having no synchrony in it, our software did detect a number of synchrony events, producing disagreement between the results of the manual analysis and the results of the automatic analysis. Despite a small percentage of false negatives in detecting the events of interest (i.e., nodding, shaking, smiling), the results of the automatic analysis are supportive of the initial hypothesis of synchrony being detectable and discriminating among conditions. This means that monitoring synchrony events, while establishing implicit models of deception, may be useful for automatic deception detection.

False negatives (for shaking and nodding) are attributed to the poor resolution of the input video and to the fact that the camera was not frontal to the faces. In particular, the face was quite small, and although it was correctly tracked, the displacement of the facial landmarks was sometimes not large enough to register as a nodding or shaking event. We believe that using videos of better quality, with facial close-ups, will improve our results and confirm our findings.

In this chapter, we investigated how the degree of interactional synchrony can signal whether deceit is present or absent. An automated framework has been introduced to analyze videos effectively, and a new group of features has been proposed that not only register synchrony but also can detect deception at a reasonable accuracy. Future analysis will consider if the trend discovered thus far by our computerized methods generalizes to the greater sample population and also to other scenarios in which deception may be present. Furthermore, we will improve our face tracking system by incorporating 3D deformable models [133, 47, 200] and sparsity-based shape priors [213, 214].

Chapter 7

Conclusions

In this thesis, we analyzed two most challenging situations, head pose variation and partial occlusion, in the wild face landmark localization problem. Two specific algorithms towards robustness in the wild conditions are proposed to deal with each of the problems. Moreover, we proposed a unified framework to simultaneously deal with the two problems while preserving the real-time efficiency. The extended user-specific expression recognition benefits from our robust face alignment methods. Moreover, a visual deception detection framework is designed based on all the previous proposed modules, i.e. expression recognition and robust landmark localization, which is further extended to face tracking. The main contributions of this thesis are summarized as follows:

- We proposed a fast multi-view cascaded face alignment algorithm to deal with the arbitrary head pose problem in the unconstrained face environment. An optimized part mixture model is novelly designed to both simplify the landmark structure and increase the localization accuracy. The bi-stage local search refines the local shape variance. Nevertheless, the simplified initial structure with the bi-stage local search approaches real-time performance, which extends the applicability of the method.

- A new regression-based facial feature localization method using a consensus of occlusion-specific regressors is presented. Those specific regressors are carefully designed to effectively resist occlusion, which can be trained on standard image datasets without occlusion labels. The consensus of sufficiently many independent regressors largely increases both the robustness and the accuracy of the method. A semi-supervised label propagation method is further introduced to provide the occlusion status of each landmark. In addition, to overcome the sensitivity problem of regression-based methods, we propose a robust initialization strategy, a max-margin learning on the variations of the head poses.

- A unified regression framework is proposed: A conditional holistic regression model is proposed to address severe pose variation and a hierarchical parts regression model is introduced to simultaneously search for optimal landmark configurations locally and infer occlusion. A projection optimization method is proposed to directly derive the hierarchical part regressors from the holistic regressors, which further improves the alignment accuracy while saving the training complexity. A graphical model is applied on the hierarchical structure to evaluate the alignment likelihood, which provides potential for early halting.

- As an immediate application of landmark localization, we proposed a user-specific expression recognition, which is a more general task other than the traditional six typical expressions' task. Under such framework, we explored a novel set of effective features to enable accurate and robust recognition of a variety of customized expressions. To solve the few training sample problem, an online few-shot SVM ensemble with an HMM-based temporal filtering algorithm is introduced, which improves the temporal coherence of expression prediction.

- Combining our developed face alignment and tracking, head gesture detection and expression recognition modules, a fully automatic visual cue extraction system is introduced and synchrony information is modeled to detect deception. Insights are put into investigating how synchrony influences the interaction of the two participants. Single-channel visual cue features are examined to show how they contribute to the classification of truth or deception. In-depth analysis are conducted for designing deception detection systems. The deception detection problem is formulated as a classification problem. Feature selection towards synchrony features and two-layer classifier design significantly improve the detection accuracy from 54%, which is what a typical unaided individual can expect, to 74%.

Our work also reveals the opportunity for some possible improvements and new directions to expand the research in computer vision, computer graphics and machine learning. There are some interesting points listed:

1) 3D face landmark localization and tracking

Traditional 2D face landmark localization has encountered a set of bottlenecks. For example, the head pose calculated by mapping 3D reference shape to 2D localization result is not always accurate. It is because, the 3D reference shape or the 2D localization may be sparse, which cannot guarantee the best match using the optimization methods. 2D dense landmarks are hard to define, e.g. the points in the textureless area cannot be uniquely identified. However, if the face landmarks are directly localized in 3D, many problems are easily solved, i.e. the head pose is uniquely determined as the 3D dense shape is fixed. There are several ways to directly localize the 3D face landmarks.

- 3D localization from face videos. The monocular view face videos must contain the faces under different view points. Firstly applying 2D localization methods to provide the 2D landmarks, a shape from motion scheme is applied to recover the 3D dense shape. As the video plays on, the 3D shape could be refined from more and more frames' 2D localizations. The challenge with monocular 3D reconstruction is the lack of sufficient 3D information. By obtaining the 2D information from different view points, such information gap is gradually filled up.

- 3D localization from RGBD images or videos. With the development of hardware devices, many instruments are convenient to provide the depth information, i.e. kinect. If faces are set within proper distance, the devices could provide the depth of faces with good granularity, e.g. the depth of checks and nose can be significantly differentiated. Though noise exists in the capturing, there are many good smoothing algorithms to retrieve the 3D face shapes. In this direction, especially in the computer graphics, there are already several pioneering work with quite good performance.

- Cooperative 3D localization. This direction is a more traditional one assuming there are multiple calibrated cameras capturing the same face. The 3D shape can be well retrieved by the canonical stereo matching. Some variation is that there is only one camera, which is similar to the 3D localization from face videos. The images for different head poses can be equivalently mapped to the images from different view-point cameras. In this way, we could still apply the stereo matching to obtain the 3D face shape.

2) General rigid object localization and tracking

The localization problem is essentially a detection problem. Face is approximately a rigid object. Thus, our method could seamlessly applied to many rigid object localizations, i.e. in-door scene desks, chairs, sofas and out-door scene cars, ships, etc. These applications are also able to be extended to 3D situations, such as retrieving 3D objects in a room. These applications lead to a higher level scene understanding, which is critical in robotics vision, autonomous driving, video surveillance, etc. The 3D vision has shown its significance in approaching the artificial intelligence.

3) Extended applications based on face landmark localization

As we have shown in chapter 5 and chapter 6, there are many extended applications which are directly built on top of the face landmark localization. They are as listed below but not limited to them.

- **Face recognition**. The preprocessing or registration of faces is a prerequisite for face recognition. Faces without registration bring in large spatial noise and significantly influences the recognition rate. Recently, the 3D face alignment has shown the significance in the top performance face recognition [167].

- **Expression recognition**. We have proposed a user-defined expression recognition framework in chapter 5. Due to the data limitation, the CNN is overfitted to the traditional six typical expressions, which lacks the generalization ability to other user-defined expressions. The unsupervised or semi-supervised distribution transfer may also alleviate the problem other than increasing the training data.

- **Face animation**. In chapter 5, with the expression recognition, we achieve the performance-driven cutout character animation. More complex application could be directly achieving face animation by rigging. The rigging information is obtained from the 2D or 3D face landmark localization or tracking.

- **Psychological analysis**. In chapter 6, we conducted a video-based deception detection analysis. The modules such as face tracking and expression recognition can be applied to other similar applications, such as American Sign Language (ASL) and autism analysis.

References

[1] http://www-prima.inrialpes.fr/FGnet/data/01-TalkingFace/talking_face.html.

[2] M. G. Aamodt and H. Custer. Who can best catch a liar?: A meta-analysis of individual differences in detecting deception. *The Foresnsic Examiner*, 15(1):6–11, 2006.

[3] K. Ashenfelter, S. Boker, J. Waddell, and N. Vitanov. Spatiotemporal symmetry and multifractal structure of head movements during dyadic conversation. *Journal of Experimental Psychology: Human Perception and Performance*, 35(4):1072–1091, 2009.

[4] A. Ashraf and S. L. ad T. Chen. Fast image alignment in the fourier domain. In *CVPR*, 2010.

[5] A. Asthana, S. Cheng, S. Zafeiriou, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *CVPR*, 2013.

[6] A. Asthana, S. Lucey, and R. Goecke. Regression based automatic face annotation for deformable model building. *Pattern Recognition*, 44(10-11):2598–2613, 2011.

[7] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Incremental face alignment in the wild. In *CVPR*, 2014.

[8] J. Bailenson and N. Yee. Digital chameleons automatic assimilation of nonverbal gestures in immersive virtual environments. *Psychological science*, 16(10):814–819, 2005.

[9] S. Baker, R. Gross, and I. Matthews. Lucas-kanade 20 years on: a unifying framework: part 3. *Tech. rep. Carnegie Mellon University*, 2003.

[10] T. Baltrusaitis, P. Robinson, and L. Morency. 3d constrained local model for rigid and non-rigid facial tracking. In *CVPR*, 2012.

[11] T. Baltrusaitis, P. Robinson, and L. Morency. constrained local neural fields for robust facial landmark detection in the wild. In *ICCVW*, 2013.

[12] A. Barbosa, E. Bateson, M. Oberg, and R. Dechaine. An instantaneous correlation algorithm for assessing intra and inter subject coordination during communicative behavior. 2010.

[13] M. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing facial expression: Machine learning and application to spontaneous behavior. In *CVPR*, pages 568–573, 2005.

[14] P. Belhumeur, D. Jacobs, D. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *CVPR*, 2011.

[15] N. Bhaskaran, I. Nwogu, M. Frank, and V. Govindaraju. Lie to me-deceit detection via online behavioral learning. In *Automatic Face and Gesture Recognition*, 2011.

[16] M. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *IJCV*, 25(1):23–48, 1997.

[17] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, pages 187 – 194, 1999.

[18] I. Bociu and I. Pitas. A new sparse image representation algorithm applied to facial expression recognition. In *MLSP*, pages 539–548, 2004.

[19] S. Boker, M. Xu, J. Rotondo, and K. King. Windowed cross-correlation and peak picking for the analysis of variability in the association between behavioral time series. *Psychological Methods*, 7(3):338–355, 2002.

[20] C. F. Bond, Jr., and B. M. DePaulo. Accuracy of deception judgements. *Personality and Social Psychology Review*, 10(3):214–234, 2006.

[21] D. B. Buller and J. K. Burgoon. Interpersonal deception theory. *Communication Theory*, 6:203–242, 1996.

[22] J. Burgoon, N. Dunbar, and C. White. *Interpersonal Adaptation*. Interpersonal Communication, in press.

[23] J. K. Burgoon, J. F. Nunamaker, Jr., and D. N. Metaxas. *Noninvasive measurement of multimodal indicators of deception and credibility*. Final Report to the Defense Academy for Credibility Assessment, July 10, 2010.

[24] J. K. Burgoon, L. A. Stern, and L. Dillman. Interpersonal adaptation: Dyadic interaction patterns. *New York: Cambridge University Press*.

[25] X. Burgos-Artizzu, P. Perona, and P. Dollar. Robust face landmark estimation under occlusion. In *ICCV*, 2013.

[26] N. Campbell. Multimodal processing of discourse information: the effect of synchrony. pages 12–15, 2008.

[27] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *CVPR*, 2012.

[28] T. Chartrand and J. Bargh. The chameleon effect: The perception behavior link and social interaction. *Journal of personality and social psychology*, 76:893–910, 1999.

[29] X. Cheng, S. Sridharan, J. Saragih, and S. Lucey. Rank minimization across appearance and shape for AAM ensemble fitting. In *ICCV*, 2013.

[30] S. Chew, S. Lucey, P. Lucey, S. Sridharan, and J. Cohn. Improved facial expression recognition via uni-hyperplane classification. In *CVPR*, pages 2554–2561, 2012.

[31] E. M. Chutorian and M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Trans. on PAMI*, 2009.

[32] I. Cohen, N. Sebe, L. Chen, A. Garg, and T. Huang. Facial expression recognition from video sequences: Temporal and static modeling. *Computer Vision and Image Understanding*, 91(1-2):160–187, 2003.

[33] I. Cohen, N. Sebe, L. Chen, A. Garg, and T. Huang. Facial expression recognition from video sequences: Temporal and static modeling. *Computer Vision and Image Understanding*, 91(1-2):160–187, 2003.

[34] J. Cohn. Automated analysis of the configuration and timing of facial expression. In *What the face reveals(2nd edition):Basic and applied studies of spontaneous expression using the Facial Action Coding System*, pages 388–392, 2005.

[35] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *ECCV*, 1998.

[36] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *PAMI*, 23(6):681–685, 2001.

[37] T. Cootes, M. Ionita, C. Lindner, and P. Sauer. Robust and accurate shape model fitting using random forest regression voting. In *ECCV*, 2012.

[38] T. Cootes and C. Taylor. A mixture model for representing shape variation. *Image and Vision Computing*, 17(8):567–573, 1999.

[39] T. Cootes and C. Taylor. Constrained active appearance models. In *ICCV*, pages 748–754, 2001.

[40] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models-their training and application. *CVIU*, 1995.

[41] T. Cootes, K. Walker, and C. Taylor. View-based active appearance models. In *Automatic Face and Gesture Recognition*, pages 227–232, 2000.

[42] D. Cristinacce and T. Cootes. Feature detection and tracking with constrained local models. In *BMVC*, pages 929–938, 2006.

[43] D. Cristinacce and T. Cootes. Automatic feature localization with constrained local models. *PR*, 2007.

[44] D. Cristinacce and T. Cootes. Boosted regression active shape models. In *BMVC*, 2007.

[45] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[46] M. Dantone, J. Gall, G. Fanelli, and L. Gool. Real-time facial feature detection using conditional regression forests. In *CVPR*, 2012.

[47] D. Decarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, 38(2):99–127, 2000.

[48] E. Delaherche and M. Chetouani. Multimodal coordination: Exploring relevant features and measures. 2010.

[49] E. Delaherche, M. Chetouani, A. Mahdhaoui, C. Georges, S. Viaux, and D. Cohen. Interpersonal synchrony: A survey of evaluation methods across disciplines. *IEEE Trans. on Affective Computing*, 3(3):349–365, 2012.

[50] B. DePaulo, J. Lindsay, B. Malone, L. Muhlenbruck, K. Charlton, and H. Cooper. Cues to deception. *Psychological Bulletin*, 129(1):74–118, 2003.

[51] H. Dibeklioglu, A. Salah, and T. Gevers. Like father, like son: Facial expression dynamics for kinship verification. In *ICCV*, pages 1497–1504, 2013.

[52] A. Dijksterhuis. Automatic social influence: The perception-behavior link as an explanatory mechanism for behavior matching. *Psychology Press*, 2001.

[53] P. Dollar, P. Welinder, and P. Perona. Cascaded pose regression. In *CVPR*, 2010.

[54] G. Duchenne. Mecanisme de la physionomie humaine, 1862.

[55] N. Dunbar, M. Jensen, D. Tower, and J. Burgoon. Synchronization of nonverbal behaviors in detecting mediated and non-mediated deception. *Journal of Nonverbal Behavior*, 2013.

[56] N. E. Dunbar, M. L. Jensen, J. K. Burgoon, B. Adame, K. J. Robertson, L. Harvell, and A. Allums. A dyadic approach to the detection of deception. In *The 44th annual Hawaii International Conference on Systems Sciences*, 2011.

[57] N. E. Dunbar, M. L. Jensen, K. M. Kelley, K. J. Robertson, D. R. Bemard, B. Adame, and J. K. Burgoon. Effects of veracity, modality and sanctioning on credibility assessment during mediated and unmediated interviews. *Communication Research*, 2013.

[58] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy"-automatic naming of characters in tv video. In *BMVC*, 2006.

[59] G. Fanelli, M. Dantone, and L. Gool. Real time 3d face alignment with random forests-based active appearance models. In *Automatic Face and Gesture Recognition*, 2013.

[60] B. Fasel and J. Luettin. Automatic facial expression analysis: A survey. *Pattern Recognition*, 36:259–275, 2003.

[61] T. H. Feeley and M. A. deTurck. The behavioral correlates of sanctioned and unsanctioned deceptive communication. *Journal of Nonverbal Behavior*, 22(3).

[62] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Trans. on PAMI*, 2009.

[63] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61, 2003.

[64] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 6(24):381 – 395, 1981.

[65] J. Friedman. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

[66] J. Friedman, T. Hastie, and R. Tibshiani. Additive logistic regression. *The Annals of Statistics*, 38(2):337–374, 2000.

[67] H. Gao, H. Ekenel, and R. Stiefelhagen. Face alignment using a ranking model based on regression trees. In *BMVC*, 2012.

[68] X. Gao, Y. Su, X. Li, and D. Tao. A review of active appearance models. *IEEE Trans. on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 40(2):145–158, 2010.

[69] G. Ghiasi and C. C. Fowlkes. Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. In *CVPR*, 2014.

[70] G. A. Giordano, J. S. Stoner, R. L. Brouer, and J. F. George. The influences of deception and computer-mediation on dyadic negotiations. *Journal of Computer-Mediated Communication*, 12(2).

[71] D. Goleman. Social intelligence: The new science of human relationships. *New York: Random House*, 2006.

[72] H. P. Grice. Studies in the ways of words. *Cambridge: Harvard University Press*, 1989.

[73] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(12):1080–1093, 2005.

[74] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 2010.

[75] N. Grujic, S. Ilic, V. Lepetit, and P. Fua. 3d facial pose estimation by image retireval. *Tech. rep., Deutsche Telekom Laboratories*, 2008.

[76] L. Gu and T. Kanade. 3d alignment of face in a single image. In *CVPR*, 2006.

[77] L. Gu and T. Kanade. A generative shape regularization model for robust face alignment. In *ECCV*, pages 413–426, 2008.

[78] Y. Guo, G. Zhao, and M. Pietikainen. Dynamic facial expression recognition using longitudinal facial expression atlases. In *ECCV*, pages 631–644, 2012.

[79] M. Haj, J. Gonzalez, and L. Davis. On partial least squares in head pose estimation: How to simultaneously deal with misalignment. In *CVPR*, 2012.

[80] O. Hamsici and A. Martinez. Active appearance models with rotation invariant kernels. In *ICCV*, 2009.

[81] C. Hart, L. Hudson, D. Fillmore, and J. Griffith. Managerial beliefs about the behavioral cues of deception. *Individual Differences Research*, 4:176–184, 2006.

[82] G. Hinton, S. Osindero, and Y. Teh. A fast learning algorithm for dep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[83] X. Hou, S. Li, H. Zhang, and Q. Cheng. Direct appearance models. In *CVPR*, 2001.

[84] G. Huang, M. Ramesh, T. Berg, and E. Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Technical Report*, 2007.

[85] Y. Huang, Q. Liu, and D. Metaxas. A component based deformable model for generalized face alignment. In *ICCV*, 2007.

[86] S. Jain, C. Hu, and J. Aggarwal. Facial expression recognition with temporal modeling of shapes. In *ICCVW*, pages 1642–1649, 2011.

[87] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, pages 304–317, 2008.

[88] M. L. Jensen, P. Lowry, and J. Jenkins. Effects of automated and participative decision support in computer-aided credibility assessment. *Journal of Management of Information Systems*, 28(1):201–233, 2011.

[89] M. Jones and T. Poggio. Multidimensional morphable models: a framework for representing and matching object classes. In *ICCV*, pages 683 – 688, 1998.

[90] M. Jones and P. Viola. Fast multi-view face detection. In *CVPR*, 2003.

[91] Z. Kalal, J. Matas, and K. Mikolajczyk. Weighted sampling for large-scale boosting. In *BMVC*, 2008.

[92] R. Kaliouby and P. Robinson. Real-time inference of complex mental states from facial expressions and head gestures. In *CVPRW*, 2004.

[93] T. Kanade, J. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *FG*, pages 46–53, 2000.

[94] A. Kapoor and R. Picard. A real-time head nod and shake detector. In *PUI*, 2001.

[95] L. Karlinsky and S. Ullman. Using linking features in learning non-parametric part models. In *ECCV*, 2012.

[96] S. M. Kassin, R. A. Leo, C. A. Meissner, K. D. Richman, L. H. Colwell, A. M. Leach, and D. La Fon. Police interviewing and interrogation: A self-report survey of police practices and beliefs. *Law and Human Behavior*, 31(4):381–400, 2007.

[97] S. Kawato and J. Ohya. Real-time detection of nodding and head-shaking by directly detecting and tracking the between-eyes. In *Automatic Face and Gesture Recognition*, pages 40–45, 2000.

[98] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *CVPR*, 2014.

[99] D. Kendall. A survey of the statistical theory of shape. *Statistical Science*, 2(4):87 – 99, 1989.

[100] M. Kipp. Spatiotemporal coding in anvil. 2008.

[101] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[102] F. D. la Torre and M. Nguyen. Parameterized kernel principal component analysis: theory and applications to supervised and unsupervised image alignment. In *CVPR*, 2008.

[103] D. Langleben. Detection of deception with fmri: Are we there yet? *Legal and Criminological Psychology*, 13:1–9, 2008.

[104] J. Larson. The polygraph and deception. *Welfare magazine*, 18:646–669, 1927.

[105] V. Le, J. Brandt, and Z. Lin. Interactive facial feature localization. In *ECCV*, 2012.

[106] D. Lee, H. Park, and C. Yoo. Face alignment using cascade gaussian process regression trees. In *CVPR*, 2015.

[107] R. W. Levenson and A. Ruef. Physiological aspects of emotional knowledge and rapport. *In W. Ickes, empathic accuracy*, pages 44–72, 1997.

[108] T. Levine, A. Shaw, and H. Shulman. Increasing deception detection accuracy with strategic questioning. *Human Communication Research*, 36:216–231, 2010.

[109] H. Li, Z. Lin, J. Brandt, X. Shen, and G. Hua. Efficient boosted exemplar-based face detection. In *CVPR*, 2014.

[110] Y. Li, L. Gu, and T. Kanade. A robust shape model for multi-view car alignment. In *CVPR*, 2009.

[111] Y. Li, L. Gu, and T. Kanade. Robustly aligning a shape model and its application to car alignment of unknown pose. *PAMI*, 33(9):1860–1876, 2011.

[112] L. Liang, R. Xiao, F. Wen, and J. Sun. Face alignment via component-based discriminative search. In *ECCV*, 2008.

[113] Y. Lin, M. Song, D. Quynh, Y. He, and C. Chen. Sparse coding for flexible robust 3d facial expression synthesis. 32(2):76–88, 2012.

[114] J. Liu, S. Ji, and J. Ye. *SLEP: Sparse Learning with Efficient Projections*. Arizona State University, 2009.

[115] M. Liu, S. Li, S. Shan, and X. Chen. Au-aware deep networks for facial expression recognition. In *FG*, pages 1–6, 2013.

[116] M. Liu, S. Shan, R. Wang, and X. Chen. Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition. In *CVPR*, pages 1749–1756, 2014.

[117] P. Liu, J. Zhou, I. Tsang, Z. Meng, S. Han, and Y. Tong. Feature disentangling machine - a novel approach of feature selection and disentangling in facial expression analysis. In *ECCV*, pages 151–166, 2014.

[118] X. Liu. Generic face alignment using boosted appearance model. In *CVPR*, 2007.

[119] X. Liu. Discriminative face alignment. *IEEE Trans. on PAMI*, 31(11):1941–1954, 2009.

[120] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[121] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[122] B. Lucas and T. Kanade. An iteratie image registration technique with an application to stereo vision. In *IJCAI*, pages 674 – 679, 1981.

[123] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete expression dataset for action unit and emotion-specified expression. In *CVPR Workshops*, pages 94–101, 2010.

[124] S. Lucey, R. Navarathna, A. Ashraf, and S. Sridharan. Fourier lucas-kanade algorithm. *IEEE Trans. on PAMI*, 35(6):1383–1396, 2013.

[125] P. Luo, X. Wang, and X. Tang. Hierarchical face parsing via deep learning. In *CVPR*, 2012.

[126] T. malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011.

[127] W. Marston. Synstolic blood pressure symptoms of deception. *Journal of Experimental Psychology*, 2(2):117–163, 1917.

[128] A. Martinez and R. Benavente. The ar face database. In *CVC Tech. Report number 24*, 1998.

[129] B. Martinez, M. Valstar, X. Binefa, and M. Pantic. Local evidence aggregation for regression-based faical point detection. *PAMI*, 2012.

[130] P. Martins, R. Caseiro, and J. Batista. Face alignment through 2.5d active appearance models. In *BMVC*, 2010.

[131] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004.

[132] J. Medina and S. Zafeiriou. Bayesian active appearance models. In *CVPR*, 2014.

[133] D. Metaxas. *Physics-based deformable models: applications to computer vision, graphics, and medical imaging*, volume 389. Springer, 1997.

[134] N. Michael, M. Dilsizian, D. Metaxas, and J. Burgoon. Motion profiles for deception detection using visual cues. In *ECCV*, 2010.

[135] S. Milborrow, J. Morkel, and F. Nicolls. The muct landmark face database. In *Pattern Recognition Association of South Africa*, 2010.

[136] S. Milborrow and F. Nicolls. Locating facial features with an extended active shape model. In *ECCV*, pages 504–513, 2008.

[137] L. Morency, A. Quattoni, and T. Darrell. Latent-dynamic discriminative models for continuous gesture recognition. In *CVPR*, 2007.

[138] J. Navarro. A four-domain model for detecting deception. *FBI Law Enforcement Bulletin*, 72(6):19, 2003.

[139] O. Oullier, G. de Guzman, K. Jantzen, J. S. Kelso, and J. Lagarde. Social coordination dynamics: Measuring human bonding. *Social Neuroscience*, 3(2):178–192, 2008.

[140] M. Pantic and L. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *PAMI*, 22(12):1424–1445, 2000.

[141] G. Papandreou and P. Maragos. Adaptive and constrained algorithms for inverse compositional active appearance model fitting. In *CVPR*, 2008.

[142] A. Quattoni, M. Collins, and T. Darrell. Conditional random fields for object recognition. In *NIPS*, 2004.

[143] R. Redner and H. Walker. Mixture densities, maximum likelihood and the em algorithm. *Society for Industrial and Applied Mathematics*, 26(2):195–239, 1983.

[144] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *CVPR*, 2014.

[145] D. Richardson and R. Dale. Looking to understand: The coupling between speakers' and listeners' eye movements and its relationship to discourse comprehension. *Cognitive Science*, 29(6):1045–1060, 2005.

[146] D. Richardson, R. Dale, and K. Shockley. *Synchrony and Swing in Conversation: Coordination, Temporal Dynamics and Communication*. Oxford University Press, 2008.

[147] M. Richardson, K. Marsh, R. Isenhower, J. Goodman, and R. Schmidt. Rocking together: Dynamics of intentional and unintentional interpersonal coordination. *Human Movement Science*, 26(6):867–891, 2007.

[148] R. Rienks, R. Poppe, and D. Heylen. Differences in head orientation behavior for speakers and listeners: An experiment in a virtual environment. *ACM Trans. Applied Perception*, 7(2):1–13, 2010.

[149] S. Rifai, Y. Bengio, A. Courville, P. Vincent, and M. Mirza. Disentangling factors of variation for facial expression recognition. In *ECCV*, pages 808–822, 2012.

[150] S. Rivera and A. Martinez. Learning deformable shape manifolds. *PR*, 2012.

[151] M. Roh, T. Oguri, and T. Kanade. Face alignment robust to occlusion. In *Automatic Face and Gesture Recognition*, 2011.

[152] O. Rudovic, V. Pavlovic, and M. Pantic. Multi-output laplacian dynamic ordinal regression for facial expression and intensity estimation. In *CVPR*, pages 2634–2641, 2012.

[153] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *ICCV Workshop*, 2013.

[154] J. Saragih and R. Goecke. A nonlinear discriminative approach to aam fitting. In *ICCV*, 2007.

[155] J. Saragih, S. Lucey, and J. Cohn. Deformable face fitting with soft correspondence constraints. In *Automatic Face and Gesture Recognition*, 2008.

[156] J. Saragih, S. Lucey, and J. Cohn. Face alignment through subspace constrained mean-shifts. In *ICCV*, 2009.

[157] J. Saragih, S. Lucey, and J. Cohn. Deformable model fitting by regularized landmark mean-shift. *IJCV*, 2011.

[158] S. Sclaroff and J. Isidoro. Active blobs. In *ICCV*, pages 1146 – 1153, 1998.

[159] C. Shan, S. Gong, and P. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27:803–816, 2009.

[160] L. Shang and K.-P. Chan. Nonparametric discriminant hmm and application to facial expression recognition. In *CVPR*, pages 2090–2096, 2009.

[161] X. Shen, Z. Lin, J. Brandt, and Y. Wu. Detecting and aligning faces by image retrieval. In *CVPR*, 2013.

[162] K. Shockley, M. Santana, and C. Fowler. Mutual interpersonal postural constraints are involved in cooperative conversation. *Journal of Experimental Psychology: Human Perception and Performance*, 29:326–332, 2003.

[163] J. Sivic, M. Everingham, and A. Zisserman. who are you? learning person specific classifiers from video. In *CVPR*, 2009.

[164] B. Smith and L. Zhang. Joint face alignment with non-parametric shape models. In *ECCV*, 2012.

[165] P. Sozou, T. Cootes, C. Taylor, and E. Mauro. A nonlinear generalization of point distribution models using polynomial regression. *Image and Vision Computing*, 13(5):451–457, 1995.

[166] C. Sutton, K. Rohanimanesh, and A. McCallum. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In *ICML*, 2004.

[167] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: closing the gap to human-level performance in face verification. In *CVPR*, 2014.

[168] X. Tan and B. Triggs. Fusing gabor and lbp feature sets for kernel-based face recognition. In *FG*, pages 235–249, 2007.

[169] Y. Tian, T. Kanade, and J. Cohn. Evaluation of gabor-wavelet-based facial action unit recognition in image sequences of increasing complexity. In *FG*, pages 229–234, 2002.

[170] F. Torre, Y. Yacoob, and L. Davis. A probabilistic framework for rigid and non-rigid appearance based tracking and recognition. In *Proc. Int. Conf. Automatic Face and Gesture Recognition*, 2001.

[171] P. Tresadern, H. Bhaskar, S. Adeshina, C. Taylor, and T. Cootes. Combining local and global shape models for deformable object matching. In *BMVC*, 2009.

[172] P. Tresadern, M. Ionita, and T. Cootes. Real-time facial feature tracking on a mobile device. *IJCV*, 96(3):280–289, 2012.

[173] P. Tsiamyrtzis, J. Dowdall, D. Shastri, I. Pavlidis, M. Frank, and P. Ekman. Imaging facial physiology for the detection of deceit. *International Journal of Computer Vision*, 71(2):197–214, 2005.

[174] B. E. Turvey. Introduction to terrorism: Understanding and interviewing terrorists criminal profiling: An introduction to behavioral evidence analysis (3rd ed.). *US: Elsevier Academic Press*.

[175] G. Tzimiropoulos. Project-out cascaded regression with an application to face alignment. In *CVPR*, 2015.

[176] G. Tzimiropoulos and M. Pantic. Optimization problems for fast AAM fitting in-the-wild. In *ICCV*, 2013.

[177] M. Uricar, V. Franc, and V. Hlavac. Detector of facial landmarks learned by the strctured output svm. In *VISAPP*, 2012.

[178] M. Valstar, B. Martinez, X. Binefa, and M. Pantic. Facial point detection using boosted regression and graph models. In *CVPR*, 2010.

[179] M. Valstar, M. Mehu, B. Jiang, M. Pantic, and K. Scherer. Meta-analysis of the first facial expression recognition challenge. *IEEE TSMC-B*, 42(4):966–979, 20012.

[180] M. Valstar and M. Pantic. Induced disgust, happiness and surprise: an addition to the mmi facial expression database. In *LREC Workshop on Emotion*, pages 65–70, 2010.

[181] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[182] G. Varni, G. Volpe, and A. Camurri. A system fo real-time multimodal analysis of nonverbal affective social interaction in user-centric media. *IEEE Trans. on Multimedia*, 12(6):576–590, 2010.

[183] C. Vogler, Z. Li, and A. Kanaujia. The best of both worlds: combining 3d deformable model with active shape models. In *ICCV*, 2007.

[184] A. Vrij. Behavioral correlates of deception in a simulated police interview. *Journal of Psychology: Interdisciplinary and Applied*, 129(1):15–28, 1995.

[185] A. Vrij. *Detecting lies and deceit: The psychology of lying and the implications for professional practice*. New York: John Wiley & Sons, Ltd, 2000.

[186] A. Vrij. Why professionals fail to catch liars and how they can improve. *Legal and Criminological Psychology*, 9:159–181, 2004.

[187] A. Vrij. *Detecting lies and deceit: Pitfalls and opportunities*. U.K.: Wiley, 2008.

[188] A. Vrij, K. Edward, and R. Bull. People's insight into their own behavior and speech content while lying. *British Journal of Psychology*, 92(2):373–389, 2001.

[189] A. Vrij, K. Edward, K. Roberts, and R. Bull. Detecting deceit via analysis of verbal and nonverbal behavior. *Journal of Nonverbal Behavior*, 24(4):239–263, 2000.

[190] X. Wang, T. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *ICCV*, 2009.

[191] Y. Wang, S. Lucey, and J. Cohn. Enforcing convexity for improved alignment with constrained local models. In *CVPR*, pages 1–8, 2008.

[192] Z. Wang, S. Wang, and Q. Ji. Capturing complex spatio-temporal relations among facial muscles for facial expression recognition. In *CVPR*, pages 3422–3429, 2012.

[193] C. H. White and J. K. Burgoon. Adaptation and communicative design: Patterns of interaction in truthful and deceptive conversations. *Human Communication Research*, 27(1):9–37, 2001.

[194] J. Whitehill, M. Bartlett, G. Littlewort, I. Fasel, and J. Movellan. Towards practical smile detection. *IEEE TPAMI*, 31(11):2106–2111, 2009.

[195] H. Wu, X. Liu, and G. Doretto. Face alignment via boosteed ranking model. In *CVPR*, 2008.

[196] C. Xi, P. Weike, J. Kwok, and J. Carbonell. Accelerated gradient method for multi-task sparse learning problem. In *ICDM*, pages 746–751, 2009.

[197] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+3d active appearance models. In *CVPR*, 2004.

[198] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, 2013.

[199] F. Yang, J. Huang, and D. Metaxas. Sparse shape registration for occluded facial feature localization. In *Automatic Face and Gesture Recognition*, 2011.

[200] F. Yang, J. Wang, E. Shechtman, L. Bourdev, and D. Metaxas. Expression flow for 3d-aware face component transfer. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 27(3):60, 2011.

[201] H. Yang and I. Patras. Sieving regression forest votes for facial feature detection in the wild. In *ICCV*, 2013.

[202] P. Yang, Q. Liu, and D. Metaxas. Boosting coded dynamic features for facial action units and facial expression recognition. In *CVPR*, pages 1–6, 2007.

[203] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3d dynamic facial expression database. In *FG*, pages 1–6, 2008.

[204] L. Yin, X. Wei, Y. Sun, J. Wang, and M. Rosato. A 3d facial expression database for facial behavior research. In *FG*, pages 211–216, 2006.

[205] Z. Ying, Z. Wang, and M. Huang. Facial expression recognition based on fusion of sparse representation. 6216:457–464, 2010.

[206] X. Yu, J. Huang, S. Zhang, W. Yan, and D. Metaxas. Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model. In *ICCV*, 2013.

[207] X. Yu, Z. Lin, J. Brandt, and D. Metaxas. Consensus of regression for occlusion-robust facial feature localization. In *ECCV*, 2014.

[208] X. Yu, F. Yang, J. Huang, and D. Metaxas. Explicit occlusion detection based deformable fitting for facial landmark localization. In *Automatic Face and Gesture Recognition*, 2013.

[209] X. Yu, S. Zhang, Z. Yan, F. Yang, J. Huang, N. E. Dunbar, M. L. Jensen, J. K. Burgoon, and D. N. Metaxas. Is interactional dissynchrony a clue to deception: Insights from automated analysis of nonverbal visual cues. In *HICSS*, 2013.

[210] T. Zavaschi, A. B. Jr., L. Oliveira, and A. L. Koerich. Fusion of feature sets and classifiers for facial expression recognition. *Expert Systems with Applications*, 40(2):646–655, 2013.

[211] C. Zhang and Z. Zhang. A survey of recent advances in face detection. *Microsoft Research Technique Report*, 2010.

[212] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *ECCV*, 2014.

[213] S. Zhang, Y. Zhan, M. Dewan, J. Huang, D. Metaxas, and X. Zhou. Sparse shape composition: A new framework for shape prior modeling. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1025–1032, 2011.

[214] S. Zhang, Y. Zhan, M. Dewan, J. Huang, D. Metaxas, and X. Zhou. Towards robust and effective shape modeling: Sparse shape composition. *Medical Image Analysis*, 16(1):265–277, 2012.

[215] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, 2014.

[216] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu. Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In *FG*, pages 454–459, 1998.

[217] G. Zhao and M. Pietiainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE TPAMI*, 29(6):915–928, 2007.

[218] Y. Zheng, X. Zhou, B. Georgescu, S. Zhou, and D. Comaniciu. Example based non-rigid shape detection. In *ECCV*, 2006.

[219] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, and D. Metaxas. Learning active facial patches for expression analysis. In *CVPR*, pages 2562–2569, 2012.

[220] F. Zhou, J. Brandt, and Z. Lin. Exemplar-based graph matching for robust facial landmark localization. In *ICCV*, 2013.

[221] M. Zhou, K. Veon, S. Mavadati, and J. Cohn. Facial action unit recognition with sparse representation. In *FG*, pages 336–342, 2011.

[222] S. Zhou and D. Comaniciu. Shape regression machine. In *Information Proceeding in Medical Imaging*, 2007.

[223] Y. Zhou, L. Gu, and H. Zhang. Bayesian tangent shape model: estimating shape and pose parameters via bayesian inference. In *CVPR*, 2003.

[224] Y. Zhou, W. Zhang, X. Tang, and H. Shum. A bayesian mixture model for multi-view face alignment. In *CVPR*, 2005.

[225] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.

[226] X. Zhu and D. Ramanan. Face detection, pose estimation and landmark localization in the wild. In *CVPR*, 2012.