Contents

Chapter 2

Background and Literature Review

The importance of facial expression in social interaction and social intelligence is widely recognized. Facial expression analysis has been an active research topic since 19th century. The first automatic facial expression recognition system was introduced in 1978 by Suwa et al. [83]. This system attempts to analyze facial expressions by tracking the motion of 20 identified spots on an image sequence. Since then, a lot of work has been done in this domain. Various computer systems have been made to help us understand and use this natural form of human communication.

This chapter reviews the state of the art of what has been done in processing and understanding facial expression. When building an FER system, these main issues must be considered: face detection and alignment, image normalization, feature extraction, and classification. Most of the current work in FER is based on methods that implement these steps sequentially and independently. Before exploring what has been done in literature for implementing these steps, we will briefly describe the problem space for facial expression analysis.

2.1 Problem Space for Facial Expression Analysis

2.1.1 Level of Description

In general there are two types of method to describe facial expression.

Facial Action Coding System

The facial action coding system [24] is a human-observer-based system widely used in psychology to describe subtle changes in facial features. FACS consists of 44 action units which are related to contraction of a specific set of facial muscles (Fig.2.1). Some of the action units are shown in Fig.2.2. Conventional, FACS code is manually labeled by trained observers while viewing videotaped facial behavior in slow motion. In recent years, some attempts have been made to do this automatically [69]. The advantage of FACS is its ability to capture the subtlety of facial expression, however FACS itself is purely descriptive and includes no inferential labels. That means in order to get the emotion estimation, the FACS code needs to be converted into the Emotional Facial Action System (EMFACS [28]) or similar systems.

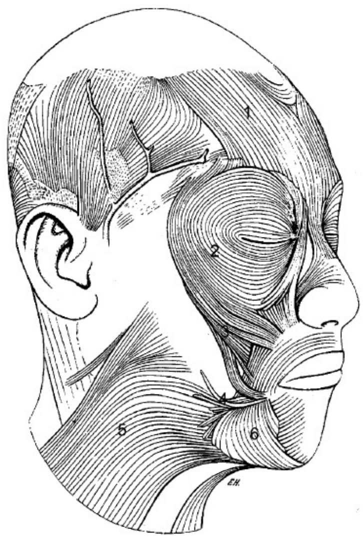


Figure 2.1: Muscles of facial expression. 1, frontalis; 2, orbicularis oculi; 3, zygomaticus major; 4, risorius; 5, platysma; 6, depressor anguli oris [33]

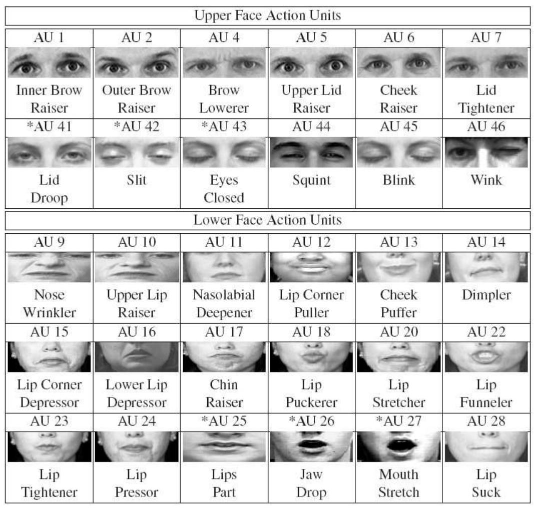


Figure 2.2: FACS action units [35]

Prototypic Emotional Expressions

Instead of describing the detailed facial features, most FER systems attempt to recognize a small set of prototypic emotional expressions. The most widely-used set is perhaps human universal facial expressions of emotion which consists of six basic expression categories that have been shown to be recognizable across cultures 2.3 .

These expressions, or facial configurations have been recognized in people from widely divergent cultural and social backgrounds, and they have been observed even in the faces of individuals born deaf and blind.

These 6 basic emotions, *i.e.*, disgust, fear, joy, surprise, sadness and anger plus "neutral" which means no facial expression are considered in this work. Given a facial image, our system either works as a conventional classifier to determine the most likely emotion or estimates the weights (or possibility) of each emotion as a fuzzy classifier does.

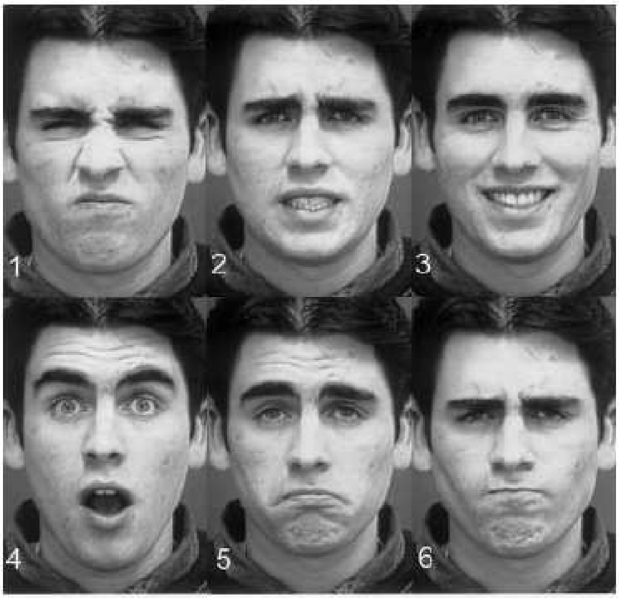


Figure 2.3: Basic facial expression phenotypes. 1, disgust; 2, fear; 3, joy; 4, surprise; 5, sadness; 6, anger

2.2 System Structure

FER can be considered as a special face recognition system or a module of a face recognition system. So it should be instructive to look at the general architecture of a face recognition system. Normally, it consists of four components as depicted in 2.4

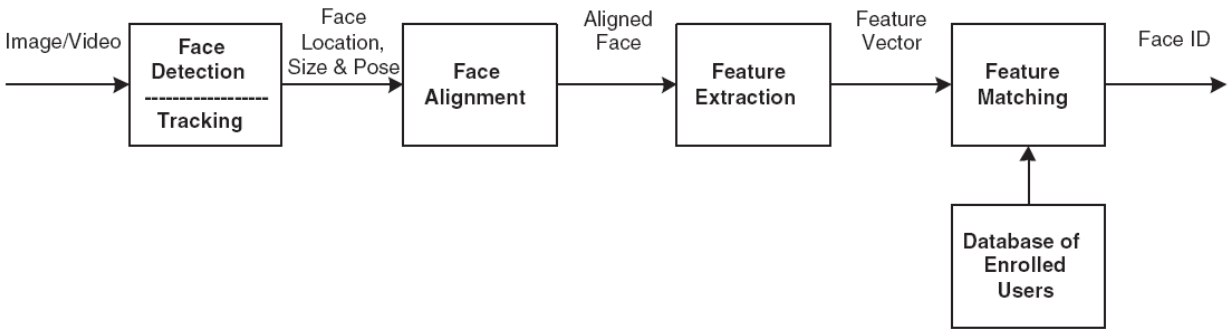


Figure 2.4: Face recognition processing flow

Face detection finds the face areas in the input image. If the input is a video, to be more efficient and also to achieve better robustness, face detection is only performed on key frames and a tracking algorithm is applied on interval frames. Face alignment is very similar to detection, but it is aimed at achieving a more accurate localization. In this step, a set of facial landmarks (facial components), such as eyes, brows and nose, or the facial contour are located; based on that, the face image is rotated, chopped, resized and even warped, this is called geometrical normalization. Usually the face is further normalized with respect to photometrical properties such as illumination and gray scale.

Feature extraction is performed on a normalized face to provide effective information that should be useful for recognizing and classifying labels in which there is interest, such as identity, gender, or expression. The extracted feature vector is sent to a classifier and compared with the training data to produce a recognition output.

2.3 Face Detection

Face detection is the first step in face recognition. It has a major influence on the performance of the entire system. Several cues can be used for face detection, for example, skin color, motion (for videos), facial/head shape, and facial appearance. Most successful face detection algorithms are based on only appearance. This may be because appearance-based algorithms avoid difficulties in modeling 3D structures of faces. However, the variations of 3D structures due to facial expression and head pose actually heavily affect the facial appearance and make the face/non-face boundary highly complex [7]. To deal with this, a vast arrange of methods have been proposed since the 1990s.

Turk and Pentland [87] describe a detection system based on eigen decomposition which is also known as principal component analysis (PCA). In their method, an image is represented by an average face plus a set of weighted eigenfaces. Whereas only the face images are considered in eigenface, Sung and Poggio [82] also consider the distribution of non-face images and apply Bayes rule to obtain a likelihood estimation. Rowley et al. [72] use neural networks and Osuna et al. [68] trained a Kernel Support Vector Machine to classify face and non-face images. In these systems, a bootstrap algorithm is iteratively used to collect meaningful examples for retraining the detector.

Schneiderman and Kanade [75] use AdaBoost learning to construct a classifier based on wavelet representation of the image. This method is computationally expensive because of the wavelet transformation. To overcome this problem, Viola and Jones [93] replace wavelets with Haar features, which can be computed very efficiently [20] [80]. Their system is the first realtime frontal-view face detector [94].

Under Violas framework, some improvements have been proposed. Lienhart et al. [52] use rotated Haar features to deal with in-plane rotation. Li et al. [51] [50] [52] propose a multiview face detection system which can also handle out-of-plane rotation using a detector-pyramid.

In the following sections, we will describe two face detection algorithms: Eigenface is one of the simplest methods and Violas framework may be the most successful one. AdaBoost learning is an important component in Violas framework and this algorithm will also be useful in the feature extraction module, so our presentation focuses on this part.

2.3.1 Eigenface and Template Matching

Eigenface assumes the face image $x = (x_1, x_2, ..., x_N)$ is amenable to a multivariate normal distribution from which the training images are identically independently drawn.

This distribution can be described by the following probability density function:

$$f(x_1, x_2, ..., x_N) = \frac{1}{(2\pi)^{N/2}|\sum|^{1/2}} exp(-\frac{1}{2}(x-\mu)^T \sum^{-1}(x-\mu))$$

where $\sum$ is the covariance matrix and μ is the expectation of x.
Eigenface decomposes $\sum$ using eigen decomposition as

$$\sum = USU^T$$

where U is a unitary matrix and $S = diag(s_1^2, s_2^2, ..., s_N^2)$ is a diagonal matrix with all elements non-negative. Each column of U, U_i, is called an Eigenface. A face image x can be represented by μ and U_i as

$$x = \mu + \sum_i a_i U_i$$

It can be shown that $\frac{a_i}{s_i}$ are i.i.d. standard normal variables. So the probability density function of x is:

$$f(x) = \prod_i -\frac{1}{(2\pi)^{1/2}} exp(\frac{1}{2}\frac{a_i^2}{s_i^2}) = \frac{1}{(2\pi)^{N/2}} exp(-\frac{1}{2}\sum_i \frac{a_i^2}{s_i^2})$$

Equation (2.4) can be used as a probability estimate and we can define a distance measure according to

$$D = \sum_i \frac{a_i^2}{s_i^2}$$

which is called normalized Euclidean distance. A large D implies a small probability of being a face image and vice versa. Based on (2.5) Turk and Pentland [87] built a face detection system. Sung and Poggio's paper [82] used a similar idea, they assume images are produced by a mixture of Gaussian models: a face image Gaussian and a non-face image Gaussian. So they also estimate the probability of x of being a non-face image $f'(x)$ and the final decision is made using a Bayesian classifier.
If we further assume $\sum$ is an identity matrix, (2.5) degenerates into Euclidean distance which means the probability density function is controlled by, $|x - \mu|^2$, the variation of the image from the average face μ. This gives the simplest detection algorithm: template matching, i.e., finding a "face template" μ, and then for each x determine whether it is a face image by thresholding $|x - \mu|$.

2.3.2 Viola's Framework

Almost all the state of the art face detection systems are developed upon Violas Framework, based on AdaBoost and Haar features. The philosophy of AdaBoost is that if it is hard to find a good (strong) classifier directly, we can construct a lot of poor quality ones (weak classifier) $h_m(x), m = 1, ..., M$ and use the combination of them to form a strong classifier [27]:

$$H_M(x) = \frac{\sum_m a_m h_m(x)}{\sum_M a_m} \tag{2.1}$$

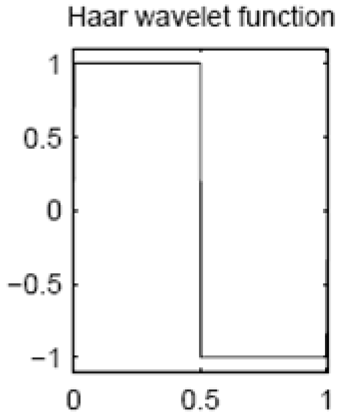
where $a_m 0$ are the combining coefficients. In the discrete version $h_m(x)$ gives either 1 or 1 whereas in the real version, the output can be a real number. Because we need a lot of weak classifiers, $h_m(x)$ is normally chose to have a simple form (so it is easy to construct). In Viola and Joness work, they use threshold classifiers, each of which works on one feature selected from an over-complete set of Haar wavelet-like features. Well first introduce Haar Wavelet, then give the iterative algorithm of AdaBoost.

Haar Wavelet

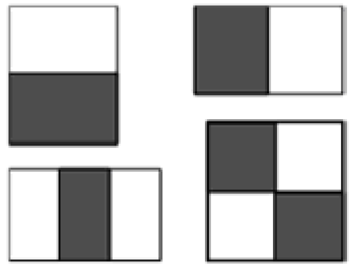
Wavelet analysis is a tool for signal processing which can perform local analysis. It is useful in face detection because we want to focus on a localized area of the image and find whether it contains a face. Haar wavelet is the first known wavelet, and probably the simplest one. A one-dimensional Haar wavelet function is just a step function, as shown in Fig.2.5(a).

In image processing, two-dimensional wavelets are used which look like the ones in Fig.2.5(b). The functions in Fig.2.5(b) dont satisfy some conditions of a wavelet (because we dont need those in our application), so strictly speaking they are "Haar-like features". An important property of Haar wavelet-like features is that they can be computed efficiently using an integral image. The integral image $II(x, y)$ of image $I(x, y)$ is defined by:

$$II(x, y) = \sum_{x'<x, y'<y} I(x', x') \tag{2.2}$$



(a) 1-dimensional Haar Wavelet function



(b) Four types of 2-d Haar wavelet-like features

Figure 2.5: Haar Wavelet and Haar features

This can be computed in one pass over the original image as follows:

$$S(x, y) = S(x, y - 1) + I(x, y) \tag{2.3}$$

where $S(x, y)$ is the cumulative sum of the xth row. Apparently $S(x, 0) = 0$ and $II(0, y) = 0$. Any rectangular sum in an image can be expressed in terms of its integral image as illustrated in 2.6

The sum of the pixels within rectangle D in 2.6 can be computed using the integral image value on points a, b, c, d as follows:

$$II(x2, y2) = \sum_{x<x2, y<y2} I(x, y) = A + B + C + D \tag{2.4}$$

$$II(x2, y1) = \sum_{x<x2, y<y1} I(x, y) = A + C \tag{2.5}$$

$$II(x1, y2) = \sum_{x<x1, y<y2} I(x, y) = A + B \tag{2.6}$$

$$II(x1, y1) = \sum_{x<x2, y<y2} I(x, y) = A \tag{2.7}$$

$$D = II(x2, y2) + II(x1, y1) - II(x2, y1) - II(x1, y2) \tag{2.8}$$

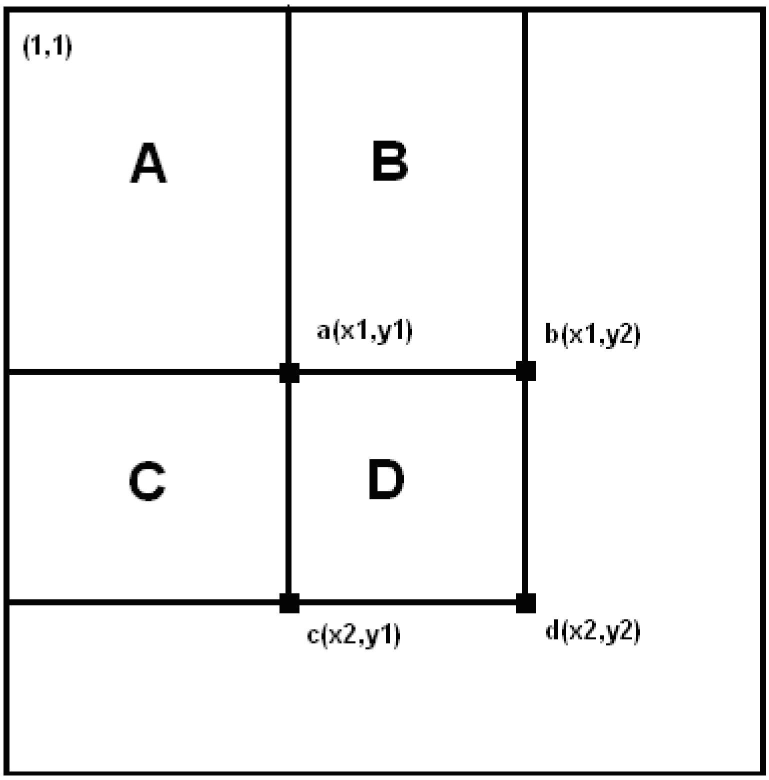


Figure 2.6: Compute rectangle sum using integral image

The use of integral images leads to enormous savings and makes it possible to use Haar wavelets in a real-time detection system.

Constructing Weak Classifier

In Violas Framework, "the AdaBoost learning procedure is used to solve the following three fundamental problems: 1) learning effective features from a large feature set; 2) constructing weak classifiers, each of which is based on one of the selected features; and 3) boosting the weak classifiers to construct a strong classifier". Well talk about the first in this section.Suppose a set of N labeled training examples $(x1, y1), ..., (xN, yN)$ is given, where $yi = 1, 1$is the label of image xi. We also assume a weightwi is assigned to each example, and how to compute this weight will be given in the next section. K Haar features $F_k, k = 1, 2, ...K$ can be computed for each image; in Violas detection system constructing a weak classifier means determining a feature Fkm, a direction $bm = 1, 1$ and a threshold f_m:

$$h_m = \begin{cases} +1 & if(F_{k_m} - f_m)b_m > 0 \\ -1 & \text{otherwise} \end{cases}$$

This simple classifier is called a stump. We choose the three parameters k_m, b_k and f_k to minimize some objective function, for example weighted classification error. This optimization can be done in two steps: for eachFk , find the best$bk, fk0$and corresponding weighted error $errk$; then choose (km, bk, fk) which corresponds to the smallest$errk$.

Boosting Strong Classifier

AdaBoost iteratively learns a sequence of weak classifiershm and constructs a strong oneHM using their linear combination. In each cycle AdaBoost does two things: find the best linear combination coefficients, and update the sample weights. Both of these tasks are based on the upper bound on classification error ofHM. [74] shows that the bound can be derived from the following exponential loss function:

$$J(Hm) = \sum e^{-y_i H_m(x_i)} = \sum e^{-yi \sum a_m h_m(x_i)} \tag{2.9}$$

Given the current strong classifier $H_(m-1)(x) = \sum a_m h_m(x)$ and the newly learned weak classifier hM, the best combining coefficientaM is the minimizer of the following optimization problem:

$$a_M = argminJ(H_(m-1)(x) + ah_m(x)) \tag{2.10}$$

$$a_M = \log(1 - err_M)/err_M \tag{2.11}$$

where err_M is the weighted error of h_M defined as

$$err_M = \sum w_i^(M-1)(sgn(h_M(x)) - y_i)/2 \tag{2.12}$$

Equation (2.17) suggests if the new weak classifier has a good performance (small err_M), well give it a large a_M; otherwise it will be assigned a small coefficient. To update the weight of each training sample, lets rewrite (2.16) as follows:

$$J(H_m) = \sum e^- y_i \sum a_m h_m(x) \tag{2.13}$$

Equation (2.19) means the contribution in $J(H_M)$ of the ith example is the error of $h_M(x)$ on x_i, $e^{-y_i a_M h_M(x_i)}$, multiplied by $e^{-y_i H_{(M-1)}(x_i)}$. So we can define the weight of the ith example after the Mth iteration as:

$$w^{(M)}(x_i) = e^{-y_i H_M(x_i)} \tag{2.14}$$

Intuitively this is saying if the current strong classifier already can classify an example x_i correctly, then it doesnt really matter what performance the new weak classifier can achieve on x_i; on the other hand, we can expect $h_M(x)$ to fix this problem.

So the updated formula of $w^{(m)}(x_i)$ is:

$$w^{(M)}(x_i) = w^{(M-1)}(x_i) e^{-y_i a_M h_M(x_i)} \tag{2.15}$$

Usually, the training set is unbalancedthe number of training images with y_i= 1 is larger(smaller) than the number of images with y_i=−1. In such cases, the y_i=1 (y_i=−1) set will dominate the training process. To eliminate this assymetry, one way is to initialize the sample weights such that:

$$\sum_i w_i^{(0)} y_i = 0 \tag{2.16}$$

The algorithm is summarized in Alg.1.

2.4 Face Alignment

The aim of face alignment is to achieve more accurate localization of the face, and usually the facial components (feature points). A typical result of face detection and face alignment is compared in Fig.2.7. As we can see, detection considers the images in terms of areas whereas alignment has a precision of pixels.

Various algorithms have been proposed since 1990's. Gu *et al.* [16] use histogram information to localize mouth corners and eye corners. In [15], Ian and Marian implement a preliminary alignment system using Gabor filter to detect pupils and philtrum. Curve fitting algorithms, especially Active Shape Model [14] and its offspring may be the most successful alignment methods nowadays. Cootes *et al.* [14] [19] [13] propose Active Shape Model and apply it to face image. After that, ASM is widely used in face image processing; a great amount of effort has been made to improve its speed, accuracy and robustness. In [18] the Active Shape Model is combined with the Gabor filter; Li *et al.* [21] propose Direct Appearance Model; [20] and [22] enhance ASM by using 2-D local textures feature for local search.

This section will mainly focus on curve-fitting types of method, in particular Active Shape Model and its ancestor Active Contour Model.

Input:

Training examples Z=(x1,y1),...,(xn,yn), where N=a+b, of which a examples have y_i=1 and b examples have y_i=-1

The number M of weak classifiers to be combined.

Initialization:

$$w_i^0 = \begin{cases} \frac{1}{2a} & \text{for those examples with yi=1} \\ \frac{1}{2b} & \text{otherwise} \end{cases}$$

end

Forward Inclusion:

for m=1, ..., M do

{

choose optimal h_M to minimize the weighted error

choose a_m according to (2.17)

update $w_i^m \leftarrow w_i^{(M-1)} e^{-y_i a_M h_M(x_i)}$ and normalize to $\sum_i w_i^{(m)} = 1$

}

end

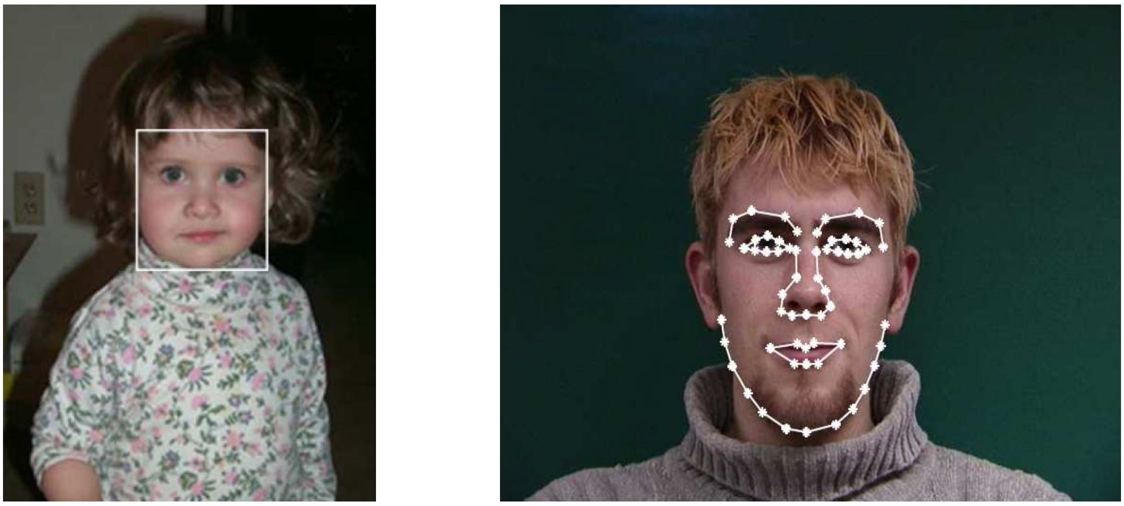
end

Output:

Classification function: $H_M(x)$ as in 2.20

Class label prediction:$y(x) = sgn(H_M(x))$

Algorithm 1: AdaBoost learning algorithm [17]



(a) Face Detection [35] (b) Face Alignment [56]

Figure 2.7: Detection Vs Alignment

2.4.1 Curve Fitting

The basic problem of curve fitting is to locate the contour of an object in an image. Most curve fitting methods (explicitly or implicitly) consider two forces: internal force (elastic force) caused by deformation and external force (image force) caused by density gradient. As in physics, both internal force and external force cause potential energy, and the curve fitting is achieved by minimizing the overall energy functional (Fig.2.8).

In general, a contour can be represented by c = c(s), parameterized by its arc length, s. The energy function can be expressed by:

$$\varepsilon = E_{elastic}(c) + E_{image}(c) = \int_{c} (a(s)e_{elastic}(c,s) + b(s)e_{image}(c,s))ds \quad (2.17)$$

So the optimization problem can be expressed as

$$\arg\min_{c} E_{elastic} + E_{image}(c) \quad (2.18)$$

Usually, (2.24) is solved using iterative searching algorithms. A large number of curve fitting methods have been proposed using a wide range of energy functions and searching schemes.

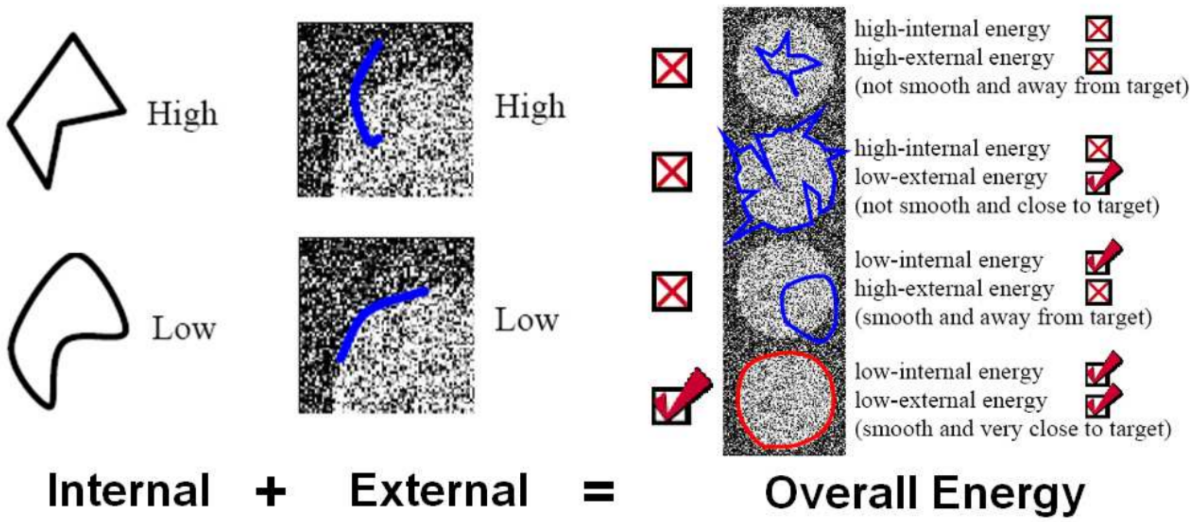


Figure 2.8: Energy function and curve fitting

Its worth pointing out that sometimes the energy function is not explicitly given, instead its implicitly defined in the searching scheme, for example, in Active Shape Model.

2.5 Feature Extraction

Feature extraction converts pixel data into a higher-level representation of shape, motion, color, texture, and spatial configuration of the face or its components. The extracted representation is used for subsequent classification. Feature extraction generally reduces the dimensionality of the input space. The reduction procedure should (ideally) retain essential information possessing high discrimination power and high stability [14]. In the face recognition area, various features have been used.

The coefficients of Eigenface can be used as features and recently, an extension of Eigenface [90] defines Tensorface which has shown a promising choice of feature. Active Appearance Model [17] decomposes the facial image into shape and texture. The shape vector which is coded using ASM describes the contour of the facial components, whereas the texture vector gives the shape-free facial texture. Matsuno et al. [41] extract features using a two-dimensional mesh, called Potential Net. All the above-mentioned methods are considered as holistic features, because they are related to the overall structure of the image. There is another kind of features called local features, each of which focuses only on a small region. The most straightforward idea may be to directly use image sub-windows as local features: for example, in [2] Colmenarez *et al.* use nine sub-windows located around the facial components. Wavelet filters have been used too, the most popular of which is the Gabor filter which has been shown [3] [4] to be a reasonable model of visual processing in primary visual cortex. Yin and Wei use topographic primitive features to represent faces [11]. In [12], instead of defining features ahead of time, Yu and Bhanu use a evolutionary algorithm to generate features automatically. For video-based FER, the dynamic of expression can also serve as features. [6] proposes Geometric Deformation Feature which represents the geometrical displacement of certain selected landmark nodes. In [1], Aleksic and Katsaggelos use Facial Animation Parameters which are based on Active Shape Model.

In this section, we'll briefly introduce some of these feature extraction methods

2.5.1 Tensorface

Tensorface [9] is a multilinear extension of Eigenface. Instead of representing the face image using a linear equation.

$$x = \overline{x} + \sum_i a_i x_i \tag{2.19}$$

It models the face by a multilinear system which is equivalent to

$$x = \overline{x} + \sum_{i_1} ... \sum_{i_1} a_{i_N} ... a_{i_N} x_{i_1 ... i_N} \tag{2.20}$$

Compared to Eigenface which ignores the label of images, Tensorface analyzes a face ensemble with respect to its underlying factors(labels): for example, identities, views, and illuminations. The "principal components" in this multilinear system are referred to as Tensorfaces which are shown in Fig.2.9. The Tensorface coefficients a_{i_k} can be used as features in a recognition task, and because the original image can be reproduced using 2.26, Tensorface coefficients can also be used for image synthesis where we first generate a set of coefficients, then use them to synthesize images.

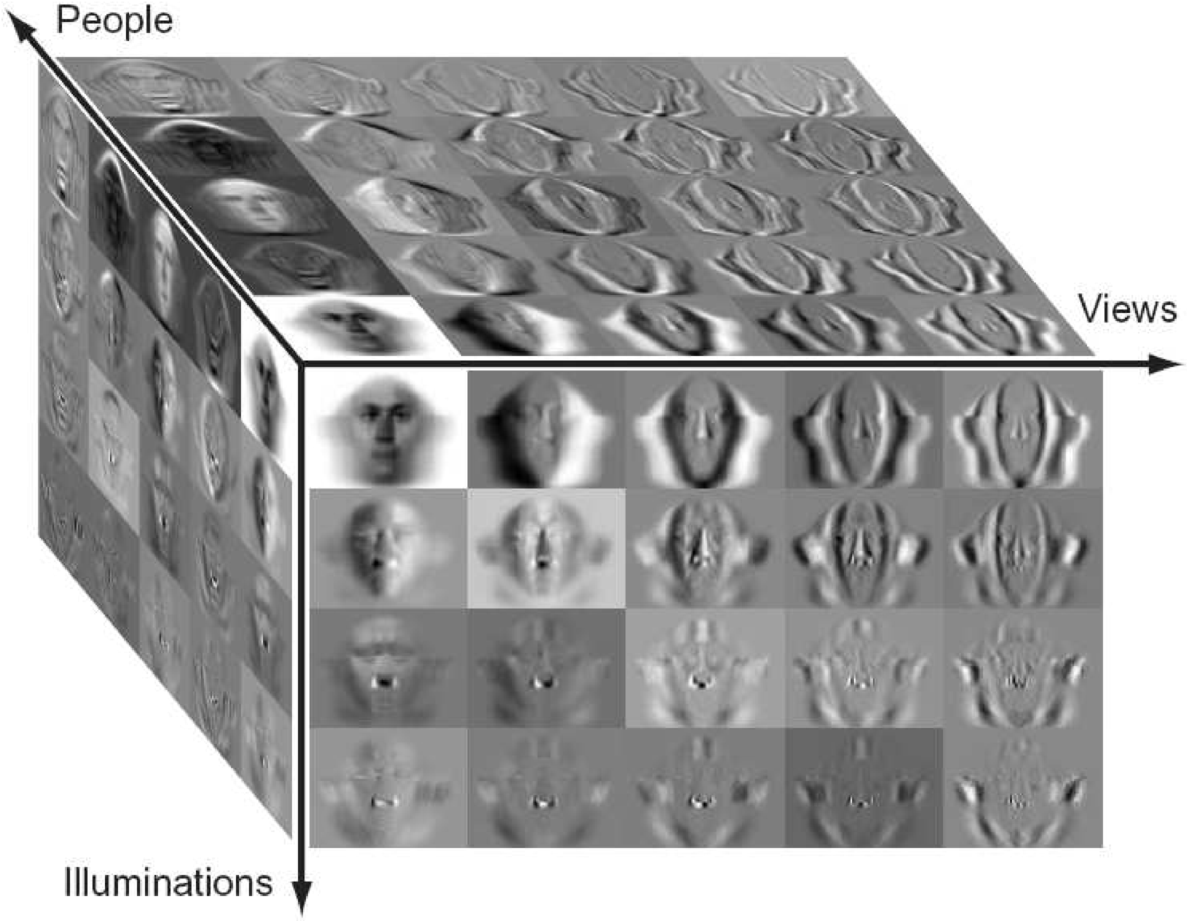


Figure 2.9: A partial visualization of TensorFaces bases for an ensemble of 2,700 facial images spanning 75 people, each imaged under 6 viewing and 6 illumination conditions [10]

Since Tensorface is shown to be a promising method in face recognition, some improvements have been proposed. [10] proposes Multilinear Independent Component Analysis where they try to find the independent directions of variation. In [8] Shashua et al. introduce Non-Negative Tensor Factorization which is a generalization of Non-negative Matrix Factorization.

2.5.2 Potential Net

Matsuno *et al.* [7] [5] propose Potential Net to extract facial features. As shown in Fig.2.10, Potential Net is a two dimensional mesh of which nodes are connected to their four neighbors with springs, while the most exterior nodes are fixed to the frame of the Net. Similar to curve fitting, Potential Net considers two forces: each node in the mesh is driven

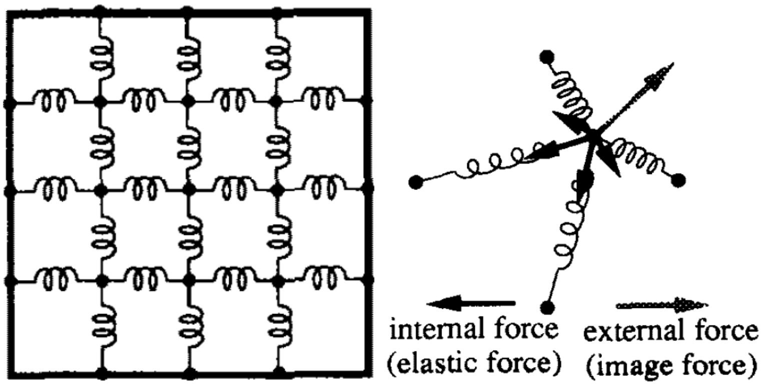


Figure 2.10: Structure of Potential Net and nodal deformation [7]

by the external forces which come from the image gradient and the elastic forces of springs propagate local deformation throughout the Net. Eventually, equilibrium is reached, and the nodal displacements represent the overall pattern of facial image.

2.5.3 Active Appearance Model

Cootes et al. propose Active Appearance Model [?] to represent objects which have deformable shape as well as diverse textures. In Active Appearance Model, the shape variation, texture variation and their interdependence are modeled using three linear systems, respectively as depicted in Fig.2.11. The shape of the facial contour is coded into a shape vector V_{shape} by applying Active Shape Model, then the texture is warped to have identical shape. This "shape-free texture" is converted into a texture vector $V_{texture}$ using PCA, finally $V_{texture}$ is concatenated with V_{shape} and translated into AAM coeficient.

AAM coeficient has fairly small dimensionality, normally 30-40 depending on the training parameters; also it is intuitive and easy to visualize. It has some limitations too, the major one is that AAM is purely linear, which means it is incapable of modeling nonlinear variation of the facial image.

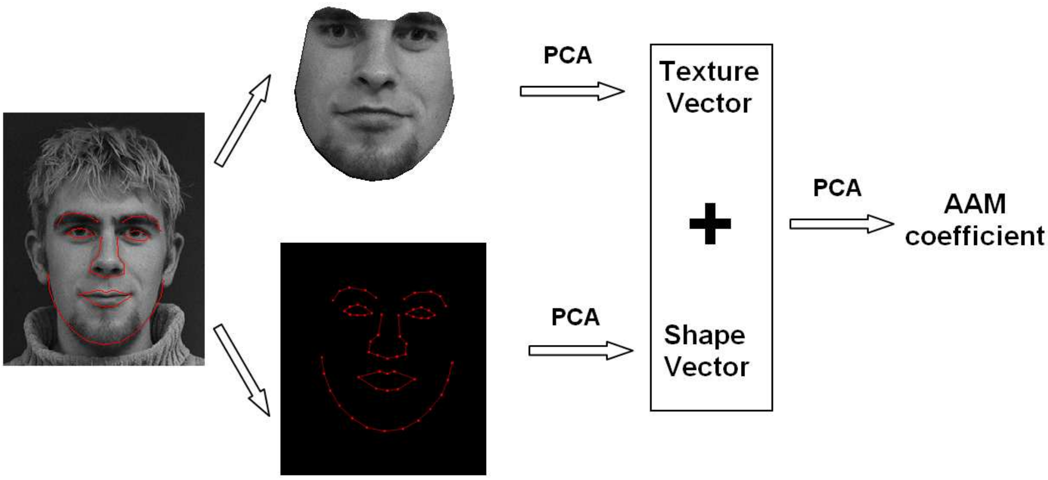


Figure 2.11: Active Appearance Model processing flow

2.5.4 Sub-window Features

In [15] the face is divided into nine facial features grouped in 4 regions, i.e., Right eyebrow, Left eyebrow, Eyes andNose and Mouth. The appearance of each facial feature is provided by the image sub-window located around its position. Fig.2.12 illustrates the four facial regions and the nine facial features.

2.6 Classifcation

A wide range of classiers have been applied to the automatic expression recognition prob- lem: In [61], Matsuno et al. classify expression by thresholding the Normalized Euclidean distance in the feature space. [15] implements a Bayesian recognition system where they nd the facial expression that maximizes the likelihood of a test image. Other meth- ods which have been used in facial expression recognition include Fisher discrimination analysis [77], Locally Linear Embedding [100], Higher Order Singular Value Decomposition [95] and so on [103] [1] [44]. Among them the most successful ones are Neural Net- work [60] [106] [42] [34] and Support Vector Machine [102] [63] [4]. All the above-mentioned methods are widely-used in statical learning, so we only briefy introduce Support Vector Machine because it will be used in other systems.

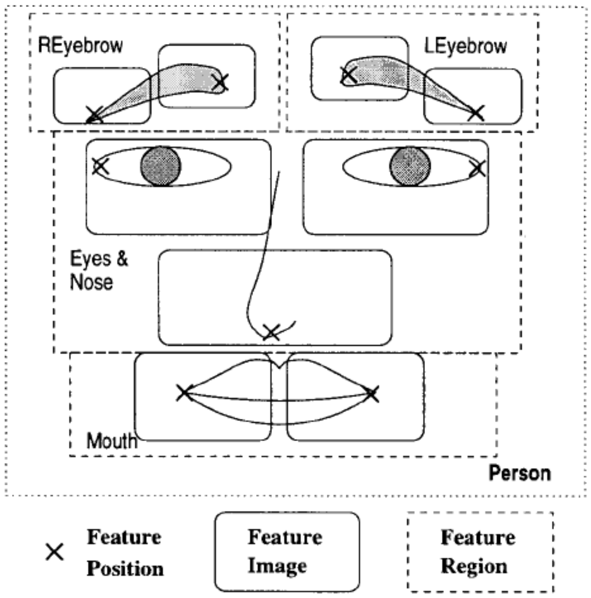


Figure 2.12: Figure 2.12: Scheme of the Facial Features and Regions [15]

2.6.1 Support Vector Machine

Support Vector Machine attempts to construct a linear classifier which maximizes the margin between two classes, so its also known as Optimal Margin Classifier [9]. Fig.2.13 gives an SVM classifier where $\frac{1}{|w|}$ gives the margin and samples along the hyper-planes are called the support vectors.

It has been proven that SVM minimizes the Structural Risk Function which is considered as a better error estimation than the normallyused Empirical Risk Function in terms of generalization capacity.

We consider data points of the form: $\{(x_1, y_1), ..., (x_n, y_n)\}$ where y_i is either 1 or -1, a label denoting the class to which the point x_i belongs. The basic version of SVM can be written as

$$\begin{array}{rl} argmax & \frac{1}{||w||} \\ s.t. & y_i(x^T w - b_0) \geq 1, \quad \forall i \end{array}$$

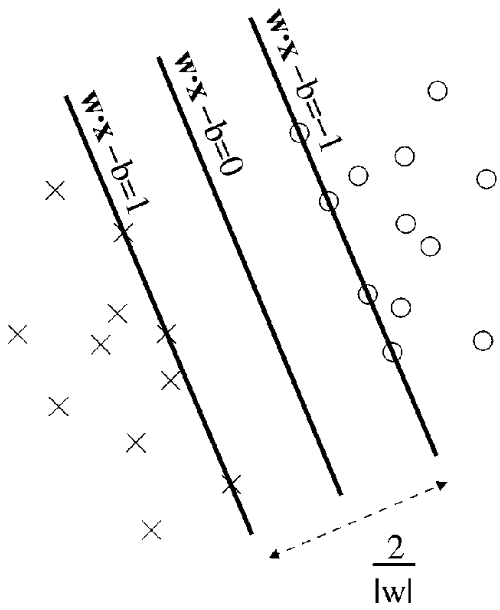


Figure 2.13: Figure 2.13: Maximum-margin hyper-planes for a SVM trained with samples from two classes [99]

$||...||$ in (2.27) can be replaced by any distance measure. If norm-2 is used, the problem is equivalent to

$$\begin{array}{rl} argmin & \frac{1}{2}||w||^2 \\ s.t. & y_i(x^T w - b_0) \geq 1, \quad \forall i \end{array}$$

Equation (2.28) is a quadratic programming and according to the strong duality theorem it can be converted to:

$$\begin{array}{rl} argmax & \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j \\ s.t. & \alpha_i \geq 0, \quad \forall i \end{array}$$

which is called the dual problem of (2.28).In practice, we always use (2.29) as it is easier to handle numerically. Moreover, in (2.29) all the computation of x i is written in terms of inner product, and that means it can be generalized to a nonlinear case by employing Kernel technique.

2.7 Face Database

"Because of its non rigidity and complex three-dimensional structure, the appearance of a face is affected by a large number of factors including identity, face pose, illumination, facial expression, age, occlusion, and facial hair. The development of algorithms robust to these variations requires databases of sufficient size that include carefully controlled variations of these factors. Furthermore, common databases are necessary to comparatively evaluate algorithms. Collecting a high quality database is a resource-intensive task: but the availability of public face databases is important for the advancement of the field" [35]. In this section we briefly review some publicly available databases for face recognition, face detection, and facial expression analysis, and we'll mainly focus on the three databases which we will use in this thesis.

To facilitate this statement, we divide face databases into two categories according to their designing goals. In the first part, we'll introduce databases which are normally used for face recognition; those which are dedicated to expression recognition will be discussed in the second part. As only a few databases are of the second type, and FER system shares some common modules with identity recognition system, in this work we also use some databases of the first type.

2.7.1 Databases For Identity Recognition

Most face databases are of this category (Table.2.1). To test for robustness, some of them are captured under different poses, illuminations and expressions. However, because they're mainly designed for identity recognition, the expressions are added as noise and usually not well controlled. So in general these databases are considered not suitable for FER research. In our work, we only use them to train peripheral modules (processing and Feature Extraction).

The IMM Face Database [66]

The IMM Face Database comprises 240 still images of 40 individuals (7 females and 33 males), all without glasses. For each person, 6 images are provided:

Table 2.1: Some of the most popular Face Recognition Databases [35]

Database	No. of subjects	Pose	Illumination	Facial Expressions
AR	116	1	4	4
BANCA	208	1	++	1
CAS-PEAL	66-1040	21	9-15	6
CMU HYPER	54	1	4	1
CMU PIE	54	1	4	1
Equinox IR	91	1	3	3
FERET	1199	9-20	2	2
Harvard RL	10	1	77-84	1
IMM FACE	40	3	2	3+
KFDB	1000	7	16	5
MIT	15	3	3	1
MPI	200	3	3	1
ND HID	300+	1	3	2
NIST MID	1573	2	1	++
ORL	10	1	++	++
UMIST	20	++	1	++
U.Texas	284	++	1	++
U. Oulu	125	1	16	1
XM2VTS	295	++	1	++
Yale	15	1	3	6
Yale B	10	9	64	1

- Frontal face, neutral expression, diffuse light.
- Frontal face, happy expression, diffuse light.
- Face rotated approx. 30 degrees to the persons right, neutral expression, diffuse light.
- Face rotated approx. 30 degrees to the persons left, neutral expression, diffuse light.

- Frontal face, neutral expression, spot light added at the persons left side.

- Frontal face, joker image (arbitrary expression), diffuse light.

The images are stored in 640 480 JPEG files. Owing to technique problems, most images are RGB, but some are grey-scale [66]. One good thing about this database is that manually labeled face contour is available. The following facial structures were annotated using 58 landmarks: eyebrows, eyes, nose, mouth and jaw. These landmarks are divided into seven point paths; three closed and four open as shown in Fig.2.14. In our work, this database will be used to train the ASM and AAM model.

Figure 2.14: Example image from IMM face database

CMU Pose, Illumination, and Expression Database [79]

The CMU-PIE database is among the most comprehensive databases in this area. It systematically samples a large number of pose and illumination conditions along with a variety of facial expressions. The PIE database was captured under 21 illuminations (lit by 21 flashes) from 13 directions (using 13 synchronized cameras). In total, there are 41,368 images obtained from 68 individuals. In our experiment, we only use a sub-set of this database which consists of images of 62 people. 25 images were selected for each individual with 5 different viewpoints and 5 different illuminations. Part of the data set is shown in Fig.2.15.

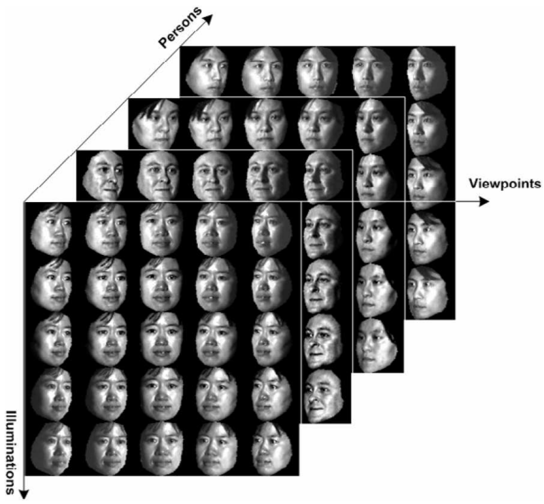


Figure 2.15: A subset of CMU PIE database [53]

2.7.2 Databases for Expression Recognition

"The human face is able to display an astonishing variety of expressions. Collecting a database that samples this space in a meaningful way is a difficult task" [35]. As a result, there are many fewer databases available for expression recognition (Table 6.1). As mentioned in 2.1.1, there are two ways to describe facial expressions. Available databases can be categorized into two classes according to the description they used. In one group [38] expressions are coded in FACS, while in the other group [57] images are labeled by their sprototypic emotional expressions.

Japanese Female Facial Expression Database [57]

The JAFFE database contains 213 images of 10 Japanese female models. Their images are labeled by emotions: six basic emotions (anger, disgust, fear, joy, happy, sad and surprise) are considered and Neutral is added as the 7th emotion which is defined through the absence of expression. Fig.2.16 shows example images for one subject along with emotion

Table 2.2: Commonly used expression recognition databases [35]

Database	No. of subjects	No. of Expressions	Image Resolution	Video/Image
JAFFE	10	7	256 X 256	Image
U. Maryland	40	6	560 X 240	Video
Cohn-Kanade	100	23	640 X 480	Video

Figure 2.16: Example images from JAFFE database[35]

labels. The images were originally printed in monochrome and then digitized using a flatbed scanner.

2.8 Chapter Summary

In this chapter, we first talked about the background of facial analysis, then gave an overview of the development in this area, and we also briefly introduced some state of the art techniques which might be useful for our system. At the end, we had a glance at some face databases for identity and expression recognition. Starting in the next chapter, we'll discuss the design of our FER system.

Bibliography

[1] P. Aleksic and A. Katsaggelos. Automatic facial expression recognition using facial animation parameters and multistream hmms. Information Forensics and Security, IEEE Transactions on, 1:311, March 2006.

[2] A. Colmenarez, B. Frey, and T. S. Huang. A probabilistic framework for embedded face and facial expression recognition. pages 592597.

[3] R. De-Valois and K. De-Valois. Spatial Vision. Oxford Univ. Press, New York, 1990.

[4] J. Jones and L. Palmer. An evaluation of two-dimensional gabor filter model of simple receptive fields in cat striate cortex. Journal of Neurophysiology, 58(6):12331258,1987.

[5] S. Kimura and M. Yachida. Facial expression recognition and its degree estimation. In CVPR 97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR 97), page 295, Washington, DC, USA, 1997. IEEE Computer Society.

[6] I. Kotsia and I. Pitas. Facial expression recognition in image sequences using geometric deformation features and support vector machines. IEEE Transactions on Image Processing, 16(1): 172187, 2007.

[7] K. Matsuno, C.-W. Lee, S. Kimura, and S. Tsuji. Automatic recognition of human facial expressions. In ICCV 95: Proceedings of the Fifth International Conference on Computer Vision, page 352, Washington, DC, USA, 1995. IEEE Computer Society.

[8] A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In ICML 05: Proceedings of the 22nd international conference on Machine learning, pages 792799, New York, NY, USA, 2005. ACM.

[9] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In ECCV (1), pages 447460, 2002.

[10] M. A. O. Vasilescu and D. Terzopoulos. Multilinear independent components analysis. In CVPR 05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05) - Volume 1, pages 547553, Washington, DC, USA, 2005. IEEE Computer Society.

[11] L. Yin and X. Wei. Multi-scale primal feature based facial expression modeling and identification. In FGR 06: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition, pages 603608, Washington, DC, USA, 2006. IEEE Computer Society.

[12] J. Yu and B. Bhanu. Evolutionary feature synthesis for facial expression recognition. Pattern Recogn. Lett., 27(11):1289 1298, 2006.

[13] T. Cootes, C. Taylor, and A. Lanitis. Multi-resolution search with active shape models. pages A:610612, 1994.

[14] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. Comput. Vis. Image Underst., 61(1):3859, 1995.

[15] I. Fasel, M. Bartlett, and J. Movellan. A comparison of gabor filter methods for automatic detection of facial landmarks, 2002.

[16] H. Gu, G. Su, and C. Du. Feature points extraction from faces. Proceedings of Image and Vision Computing New Zealand Conference, pages 154158, November 2003.

[17] A. K. Jain and S. Z. Li. Handbook of Face Recognition. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[18] F. Jiao, S. Li, H.-Y. Shum, and D. Schuurmans. Face alignment using statistical models and wavelet features. cvpr, 01:321, 2003.

[19] A. Lanitis, C. J. Taylor, and T. F. Cootes. Automatic interpretation and coding of face images using flexible models. IEEE Trans. Pattern Anal. Mach. Intell., 19(7):743756, 1997.

[20] Y. Liu, Y. Li, L. Tao, and G. Xu. Multi-view face alignment guided by several facial feature points. In ICIG 04: Proceedings of the Third International Conference on Image and Graphics (ICIG04), pages 238241, Washington, DC, USA, 2004. IEEE Computer Society.

[21] Y. ShuiCheng and Q. Cheng. Multi-view face alignment using direct appearance models. In FGR 02: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, page 324, Washington, DC, USA, 2002. IEEE Computer Society.

[22] S. Xin and H. Ai. Face alignment under various poses and expressions. In Tao et al. [85], pages 4047.