Facial image registration

Weiyuan Ni

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Signal - Images - Parole - Télécoms (SIPT)**

Arrêté ministérial : 07/08/2006

Présentée par

M. Weiyuan NI

Thèse dirigée par **Mme. Alice CAPLIER**
et codirigée par **Mme. Annie LUCIANI**

préparée au sein **ICA Laboratory**
et de **l'école doctorale Électronique, Électrotechnique, Automatique et Traitement du Signal (EEATS)**

Recalage d'images de visage

Thèse soutenue publiquement le **11/12/2012**,
devant le jury composé de :

M. Pierre-Yves COULON
Institut polytechnique de Grenoble, Président
Mme. Catherine ACHARD
Université Pierre et Marie Curie, Paris 6, Rapporteur
M. Renaud SEGUIER
Supélec Rennes, Rapporteur
M. Mohsen ARDABILIAN
LIRIS, Université de Lyon, Examinateur
Mme. Annie LUCIANI
Institut polytechnique de Grenoble, Examinatrice
Mme. Alice CAPLIER
Institut polytechnique de Grenoble, Examinatrice
M. Ngoc-Son VU
Vesalis, Invité

Abstract

Face alignment is an important step in a typical automatic face recognition system. This thesis addresses the alignment of faces for face recognition application in video surveillance context. The main challenging factors of this research include the low quality of images (e.g., low resolution, motion blur, and noise), uncontrolled illumination conditions, pose variations, expression changes, and occlusions. In order to deal with these problems, we propose several face alignment methods using different strategies. The first part of our work is a three-stage method for facial point localization which can be used for correcting mis-alignment errors. While existing algorithms mostly rely on a priori knowledge of facial structure and on a training phase, our approach works in an online mode without requirements of pre-defined constraints on feature distributions. The proposed method works well on images under expression and lighting variations. The key contributions of this thesis are about joint image alignment algorithms where a set of images is simultaneously aligned without a biased template selection. We respectively propose two unsupervised joint alignment algorithms : "Lucas-Kanade entropy congealing" (LKC) and "gradient correlation congealing" (GCC). In LKC, an image ensemble is aligned by minimizing a sum-of-entropy function defined over all images. GCC uses gradient correlation coefficient as similarity measure. The proposed algorithms perform well on images under different conditions. To further improve the robustness to mis-alignments and the computational speed, we apply a multi-resolution framework to joint face alignment algorithms. Moreover, our work is not limited in the face alignment stage. Since face alignment and face acquisition are interrelated, we develop an adaptive appearance face tracking method with alignment feedbacks. This closed-loop framework shows its robustness to large variations in target's state, and it significantly decreases the mis-alignment errors in tracked faces.

Keywords : Face alignment, facial point localization, joint image alignment, Lucas-Kanade entropy congealing, gradient correlation congealing, face tracking.

Table des matières

Chapitre 1

Introduction

Face recognition is one of the most active tasks in computer vision. For a typical automatic face recognition system, face detection and face recognition are two main steps. Face detection algorithms aim at locating face regions in input images. Face recognition methods then identify those detected faces by comparing them with reference images based on certain criteria. Within the last decades, researchers have proposed numerous face detection and face recognition methods [126, 128, 131]. Unfortunately, face detectors are still not error-free, i.e., it remains some spatial mis-alignments in detected faces, such as translation, scaling and rotation errors. The studies in [72, 103, 122, 96] demonstrated that mis-alignments inevitably lead to an obvious decrease in performance of face recognition approaches.

One possible solution to mis-alignment is developing face recognition methods robust to mis-alignment, such as invariant features and mis-alignment modeling. Invariant features based methods attempt to represent face images using features robust to scale, rotation, and translation errors. Some typical features adopted in face recognition, e.g., Local Binary Pattern (LBP) [1], Eigenfaces [111], and Gabor wavelets [69], have certain robustness to small mis-alignments. Mis-alignments modeling approaches try to take into account these geometric errors during the training of face models. In [86, 103], additional training samples are generated by manually perturbing images to model spatial mis-alignment. However, these approaches cause a huge increase of training data, and they still cannot fully handle the effect of misalignment. Take [103] as example, although using expanded training data can improve the face recognition performance when test faces undergo mis-alignment errors, there exists a decrease of recognition accuracy when input faces are well aligned.

Another solution to mis-alignment is introducing face alignment as a middle stage between face detection and face recognition. The main steps of this face recognition system are shown in Figure 1.1. The purpose of face alignment is to transform detected faces into a standard pose. Aligned faces are then output to the face recognition phase. Some researches, e.g., [53], have indicated that the use of face alignment methods can significantly improve face recognition accuracy. Based on the above analysis, we choose to adopt face alignment algorithms to deal with mis-alignment.

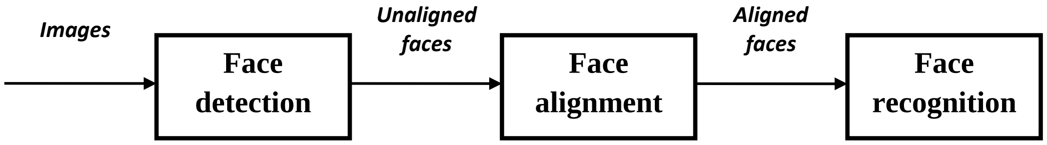


FIGURE 1.1 – Main steps of a face recognition system.

1.1 Impact of mis-alignment on face recognition

In computer vision, face recognition can be seen as a classification or matching problem of face images regarding their represented features. Simply, an unknown face (named *probe* image) is compared with all registered faces (*gallery* images) and it will be assigned to the identity (class) with shortest distance. Evidently, spatial mis-alignments cause unwanted differences between images, an example is shown in Figure 1.2 where (b) contains three images generated from (a) respectively using translation, rotation, and scaling transformation and (c) consists of the difference images between (a) and (b). It is clear that differences appear after simple transformations, even if these faces are cropped from the same image. If the differences caused by mis-alignment are larger than the differences between facial appearances, mis-classification eventually occurs.

The work of [103] tests the impact of mis-alignment on the recognition rates of a typical face recognition method, Fisherface based method [11], which projects faces in a low-dimensional subspace using Linear Discriminant Analysis (LDA). To model mis-alignments, the probe images were manually perturbed by translation, rotation, and scale transformations. The rank-1 recognition rates of Fisherface based face recognition method under different mis-alignments are shown in Figure 1.3. It is clear that small perturbations result in an obvious decrease in the performance of the face recognition method. For example, the recognition rate decreases from 90% to 60 % when probe images are 4.2 degrees rotated. With the increasing of mis-alignment intensity, the recognition accuracy tends to be zero.

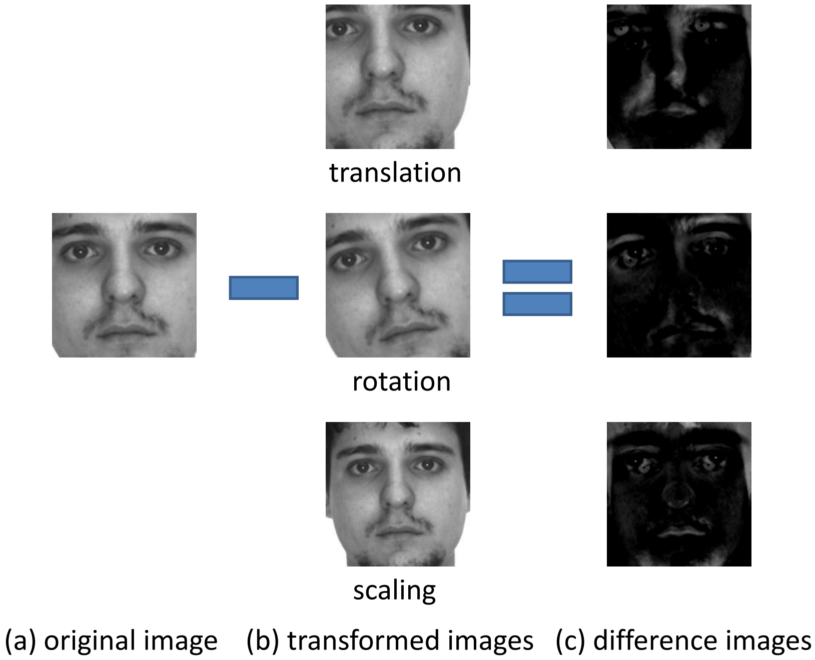
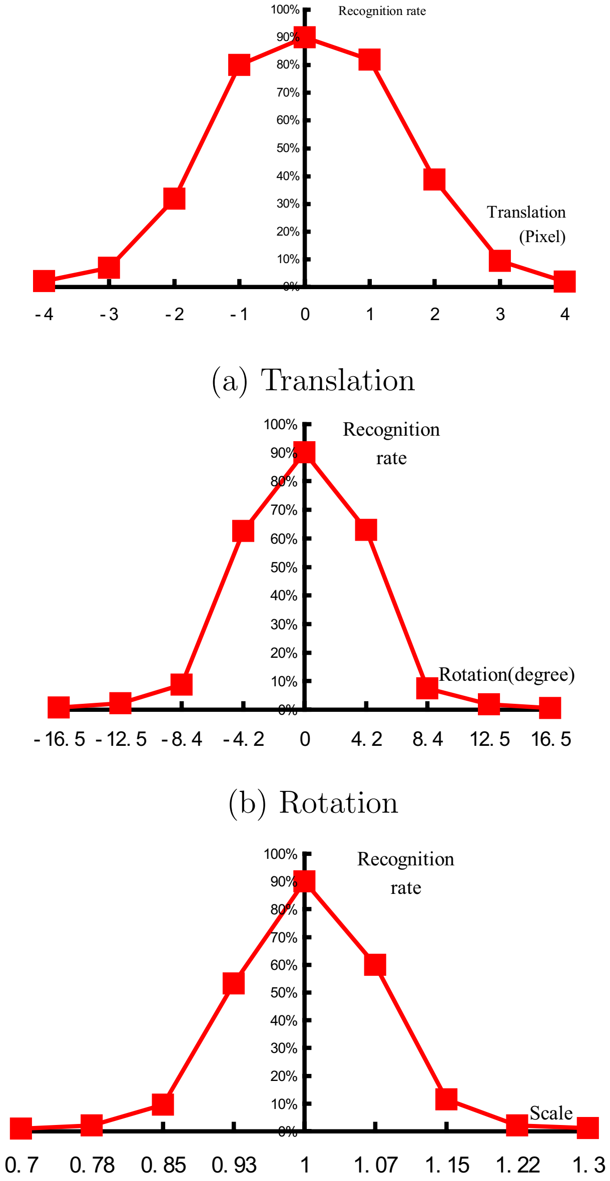


Figure 1.2 – Unwanted differences caused by mis-alignments.

Moreover, [72] evaluates the recognition performance of an Eigenfaces based method on image under downsampling, scaling, rotation , morphing, and luminance changes. In [96], six different face recognition methods, Eigenfaces, Fisherfaces, Elastic Bunch Graph Matching (EBGM) [123], pseudo Two-dimensional Hidden Markov Models (2D-HMM) [101], correlation filters [20], and Laplacianfaces [49], are compared on randomly perturbed images. Also, the work of [122] presents the relationship between face recognition accuracy of a Principal Component Analysis (PCA) baseline algorithm [94] and eye localization error on images from FRGC 1.0 database [94]. According to the presented results, spatial mis-alignments inevitably lead to performance degradation in face recognition tasks.

In the last few years, some efficient face recognition algorithms have been proposed based on more sophisticated facial feature extraction (e.g., Local Binary Pattern (LBP) [1], Histogram of Oriented Gradients (HOG) [74], and Patterns of Oriented Edge Magnitudes (POEM) [119, 120]), nevertheless, the effect of mis-alignment is still far from negligible.



(a) Translation

(b) Rotation

(c) Scale

FIGURE 1.3 – Relationship between the rank-1 recognition rate of Fisherface based algorithm and the mis-alignment of translation, rotation, and scale [103].

1.2 The main challenges for face alignment

Our work focuses on face alignment under uncontrolled conditions of image acquisition. In particular, we are very interested in face recognition in case of video surveillance applications. In those cases, we will have to cope with the following problems :

- **Variations in subject's pose :** Since human faces are 3D objects, variations in subject's pose or viewpoint usually result in nonrigid changes in facial features, and sometimes self-occlusions. These factors make face alignment very difficult.
- **Occlusions :** Apart from the self-occlusions due to poses, there are various other reasons for occlusions, like sun glasses, scarfs, etc. In those cases, face alignment methods need to be robust to outliers.
- **Variations in illumination conditions :** Aligning faces under uncontrolled illumination conditions is a difficult problem, because lighting changes in any factors (e.g., intensity, direction and color) may cause dramatical changes in the appearance of a face.
- **Variations in expression :** Faces of different expressions may have significantly different appearances in facial features. It is always difficult to align a laughing face with open mouth (closed eyes) to a neutral face with closed mouth (open eyes).
- **Low resolution :** Faces extracted from video surveillance images are usually of low resolution (e.g., only 30×30). Detailed information about face is not contained in such a small region. This is challenging for accurate face alignment.
- **Motion blur :** Motion blur often occurs when recording moving objects with a surveillance camera. In that case, blurry facial features make face alignment very difficult.
- **Image noise :** A well-known issue of camera surveillance footages is image noise. Especially for a low resolution image, noise is able to dramatically change the appearance of facial features. Hence, image noise is an unignorable factor in face alignment.

1.3 Main contributions

The work of this thesis focuses on face alignment in video surveillance applications, i.e., faces may undergo occlusion, low image quality, and variations in pose, expression, and illumination, etc. Specifically, we propose the following contributions :

(1) The first important contribution of this work is the development of an unsupervised joint face alignment algorithm, referred to as "Lucas-Kanade entropy congealing" (LKC), where an image ensemble is aligned by minimizing a sum-of-

entropy function defined over all images. We solve this minimization problem using both forward and inverse Lucas-Kanade formulations. Unlike the canonical entropy congealing [53] which estimates transformation parameters sequentially, our LKC algorithms are able to estimate all the transformation parameters at the same time. Also, in a joint alignment manner, there is no requirement of pre-defined templates.

(2) We also propose another efficient unsupervised joint face alignment algorithm named "gradient correlation congealing" (GCC) which uses gradient correlation coefficient as similarity measure. While most existing face alignment methods suffer from outliers, e.g., occlusions, GCC is able to align faces undergoing partial occlusions. Moreover, our algorithm can cope with non-uniform illumination changes (even extremely difficult ones).

(3) In order to align images with large mis-alignment errors, we propose a multi-resolution solution to joint face alignment : in coarse levels, images are processed with lower resolutions to remove major mis-alignment errors ; in fine levels, alignment is refined using higher resolutions.

(4) Since face alignment and face acquisition are interrelated, we develop an adaptive appearance face tracking method with alignment feedbacks. We first apply a self-adaptive dynamical model to predict target candidates in a particle filtering framework. Hence, our tracker is able to work with identical parameters for various situations. In order to decrease the impact of mis-alignment, we employ a multi-view joint face alignment phase based on LKC. Aligned faces are further used as feedbacks to update the appearance model of target face.

(5) Apart from these joint alignment algorithms, a three-stage method is proposed for facial point localization. While existing algorithms mostly rely on a priori knowledge of facial structure and on a training phase, our approach works in an online mode without requirements of pre-defined constraints on feature distributions. Instead of training specific detectors for each facial feature, a generic method is first used to extract a set of interest points from test images. Using POEM histogram, a smaller number of these points are picked as candidates. Then we apply a game-theoretic technique to select facial points from the candidates, while the global geometric properties of face are well preserved.

1.4 Report plan

The remainder of this report is arranged as follows :

• Chapter 2 first reviews classic and recent face alignment methods which are classified into three categories : feature point based, direct alignment, and joint alignment. For each category, there is an overview of relevant alignment approaches, and the basic knowledge of some important algorithms, i.e., Active Shape Model (ASM) [24], Active Appearance Model (AAM) [22], Lucas-Kanade algorithm [83], Learned-Miller congealing [71], and least squares congealing [26]. Also, introduced in this chapter are several famous face databases (AR [87], FERET [95], Yale B [40], SCface [42], and LFW (Labeled Faces in the Wild) [54]), and common evaluation criteria for face alignment algorithms.

- Chapter 3 presents an online approach without requirements of pre-defined constraints on feature distributions, while existing algorithms mostly rely on a priori knowledge of facial structure and on a training phase. More precisely, Section 3.3 first introduces the robust POEM (Patterns Oriented Edge Magnitude) feature which is often used to catch image information in this report. Section 3.4 discusses an effective solution to matching problem using game theory. In section 3.5, we detail the information about a three-stage facial point localization method. The experimental results are discussed in Section 3.6.

- Chapter 4 first respectively discusses two formulations of Lucas-Kanade entropy congealing (LKC), forward (4.2.1) and inverse (4.2.2), which are different in the role of "template". Comparison results on images under different conditions are given in Section 4.2.4. In order to improve the robustness to large mis-alignment errors, we then present a multi-resolution framework for LKC in Section 4.3.1. Experiments in Section 4.3.2 prove the efficiency of the multi-resolution strategy.

- Chapter 5 proposes another unsupervised joint face alignment framework named "gradient correlation congealing" (GCC) which uses gradient correlation coefficient as similarity measure. There are two formulations of this method regarding the selection of "template" : GCC-1 (5.3.2) and GCC-2 (Section 5.3.3). Experimental results in Section 5.4 prove the efficiency of our approaches under different conditions, especially when faces are partially occluded, the proposed GCC-2 algorithm performs much better than other considered methods.

- Chapter 6 presents an adaptive appearance method for tracking faces in uncontrolled environments. Section 6.2 first introduces an incremental update algorithm for target's appearance model. Section 6.3 discusses the details of our adaptive appearance face tracking method using alignment feedbacks. We test the proposed algorithm on outdoor surveillance videos and real-world YouTube videos, and experimental results are given in Section 6.4.

- In Chapter 7, we discuss the conclusions and perspectives of this thesis.

Chapitre 2

Review of Face Alignment Methods, Databases, and Evaluation Criteria

2.1 Introduction

Face alignment is a specific subject of image alignment/registration which has been widely researched within the last decades. The work of [50] reviews classic medical image alignment methods. In [134, 30], image alignment methods are classified regarding their basic steps, e.g., feature detection, similarity measure, transformation model, and optimization process. [107] divides image alignment methods into direct (pixel-based) alignment, feature-based registration, and global registration. More precisely, approaches using pixel-to-pixel matching are called direct methods, as opposed to the feature-based methods which are usually relied on feature detectors. Global alignment mentioned in [107], referred to as "bundle adjustment" [110], is usually employed for the registration of multiple 3D images.

Be different from general objects, faces possess certain characteristics, e.g., the structure of facial features. Therefore, some specific methods can be used for face analysis, e.g., face models. According to ways of application, we divide face alignment algorithms into three main categories, i.e., feature point based, direct alignment, and joint alignment, which cover most existing relevant methods. There are some relations between different types of face alignment methods, e.g., joint alignment methods may be based on face models (point detection based) or direct alignment cost functions. Sections 2.2 to 2.4 respectively introduce the three categories of face alignment approaches. Section 2.5 introduces several famous face databases, and Section 2.6 presents the main solutions used in literature to evaluate the performance of face alignment algorithms. Conclusion of this chapter is given in Section 2.7.

2.2 Feature point based face alignment

In research community, well-aligned faces are usually cropped from images according to their manually labeled landmarks, e.g., eye centers, nostrils, and mouth corners. Hence, an intuitional solution to mis-alignment is detecting facial feature points and then transforming them to standard positions. An overview of classic and recent feature point based face alignment methods is presented in Section 2.2.1, and two important face models, Active Shape Model (ASM) [24] or Active Appearance Model (AAM) [22], are respectively introduced in Sections 2.2.2 and 2.2.3.

2.2.1 Overview

According to [115], facial feature point detection can be divided into two categories : texture-based and shape-based. Texture-based approaches model the local texture around a given facial point, e.g., the pixel values in a small region around an eye center. Shape-based methods consider all facial landmarks as a shape model, which is trained from labeled faces, and try to find the shapes for unknown faces.

2.2.1.1 Texture-based methods

Some early researches localize two eyes by extracting distinct features, e.g., image gradient [66], projection function [133], template [64, 34], and wavelet [55]. Figure 2.1 illustrates an example of face alignment using detected eye centers. In this case, we can fit a similarity warp which transforms the detected eye centers to their canonical positions. Then, the face can be aligned by this estimated transformation. The work of [122] proves the efficiency of eye detection by applying it to face recognition. In order to improve the face recognition accuracy, [44] also aligns test facial images using an adaptive thresholding based eye center detector.

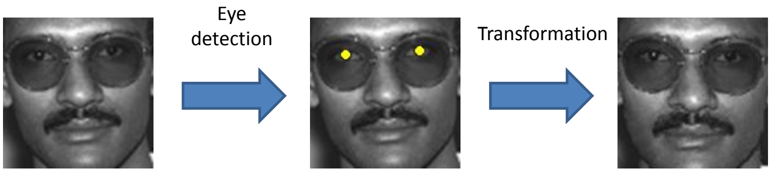


Figure 2.1 – Face alignment using detected eye centers.

Many texture-based detectors aim at locating multiple facial feature points, and prior knowledge about facial feature distribution is often used to improve the performance of the detector. In [12], five facial points (corners of the left and right eyes, corners of the mouth, and the tips of the nose) are localized by multiple Support Vector Machines (SVM). [121] divides faces into several regions of interest(ROI) regarding the geometric information about face, then individual feature patch templates based on Gabor filters are used to detect points in the relevant ROI. [100] proposes a coarse-to-fine facial landmarking detection method using Gabor features and Discrete Cosine Transform (DCT) coefficients. The work of [32] first localizes two eyes using textural information of both feature point and its context, and then estimates the approximate positions of other features with a priori knowledge about face. [115] locates 22 fiducial points using a combination of support vector regression and Markov Random Fields where pairwise spatial relations between facial point positions are learned for detection. To improve the performance of high level face analysis application (e.g., face recognition) using these detected feature points, one way is to directly align the face using a similarity transformation [48] or an affine transformation [12, 15] which best maps detected points to standard positions. Another possible solution is to represent face using a set of feature descriptors and match detected points with their corresponding features separately. For example, [13, 84] adopts Scale-Invariant Feature Transform (SIFT) descriptor [82] for face authentication. Figure 2.2 shows the comparison between gallery and probe faces using extracted SIFT features.

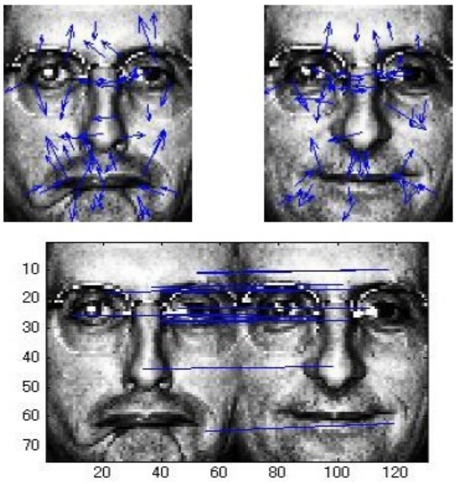


FIGURE 2.2 – SIFT features are extracted and matched between gallery and probe faces [84].

2.2.1.2 Shape-based methods

Shape-based (or model-based) methods try to detect shapes of facial features instead of separate facial points. Typical shape-based algorithms include detectors based on Active Shape Model (ASM) [24] or Active Appearance Model (AAM) [22]. ASM [24] models shape and gray-level appearance to locate flexible objects in new images. This approach was first applied to detect facial feature points in [70]. AAM [22] builds a statistical appearance model using both facial shape and texture. Shape based methods usually construct face models using Principal Component Analysis (PCA) techniques and then fit these trained models to new faces. Figure 2.3 illustrates an example of face alignment using a shape model $\boldsymbol{s}_0$ where a new face in the canonical pose can be generated regarding estimated parameters $\boldsymbol{p}$.

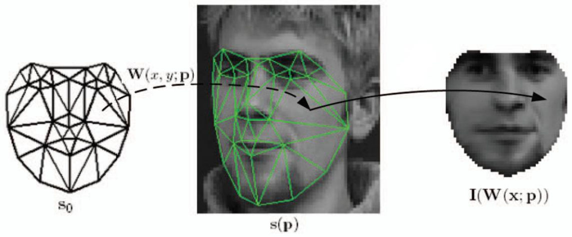

$\mathbf{W}(x, y; \mathbf{p})$
$\mathbf{s}_0$
$\mathbf{s}(\mathbf{p})$
$\mathbf{I}(\mathbf{W}(\mathbf{x}; \mathbf{p}))$

FIGURE 2.3 – Illustration of face alignment using shape-based methods [78].

Within the last decades, many researches have been done to improve the performance of ASM and AAM. In the early work of [25], the use of multi-resolution strategy increases both speed and quality of ASM. To improve the robustness of texture feature in ASM, different type of methods such as Gabor wavelets [61], AdaBoosted histogram classifiers[75], hierarchical classifier network [129] have been employed for discriminating local texture. [132] proposes the Bayesian Tangent Shape Model (BTSM) where shape analysis is formulated in Bayesian framework and tangent shape parameters are estimated using an expectation maximization (EM) method. Hu et al. [52] replaces the eigenspace-based appearance model in AAM with a wavelet network to improve the robustness to illumination changes and occlusions. The work of [33] presents a fast AAM search approach using canonical correlation analysis. In [124, 78], the fitting of face model is formulated as classification problem. [76, 56] employ shape constraints for accurate face alignment. [63] aims at eliminating the negative effect of illumination variations by learning both identity and illumination models. To cope with partial occlusion, [97] proposes a shape-based face alignment algorithm based on multiple feature detectors and random sample consensus (RANSAC) strategy [37]. Shape-based algorithms are often combined with other point detectors to form a coarse-to-fine approach to facial feature detection. For example in [29], facial points are first predicted by a detector called pairwise reinforcement of feature responses (PRFR) and then are refined using an AAM search.

Canonical shape-based algorithms train face models from manually labeled images, i.e., they work in a supervised manner. Annotating each of training images involves labeling 50–70 facial landmarks, this makes annotation of a large database tedious and time consuming. To reduce the cost of annotation, statistical models have been built using a semi-supervised solution where training images are incompletely labeled [43]. Recently, automatic face annotation approaches [102, 5] and other unsupervised methods such as [31] have been presented.

After fitting the face shape, we can deform the face to a canonical pose using non-rigid transformation (as shown in Figure 2.3), then apply these new faces to high level applications. In [58], facial images are first fitted with the face model, then average images of multiple deformed faces are used for face recognition.

2.2.2 Active Shape Model (ASM)

In [70], a set of labeled facial images are used to train a flexible shape model (see some shape examples in Figure 2.4). Each training image $\boldsymbol{X}_i$ is represented :

$$\boldsymbol{x}_i = (x_{1i}, y_{1i}, x_{2i}, y_{2i}, ..., x_{Ni}, y_{Ni}) \tag{2.1}$$

where N is the number of labeled landmarks, (x_{ki}, y_{ki}) stands for the coordinates of kth landmark in the ith training image.

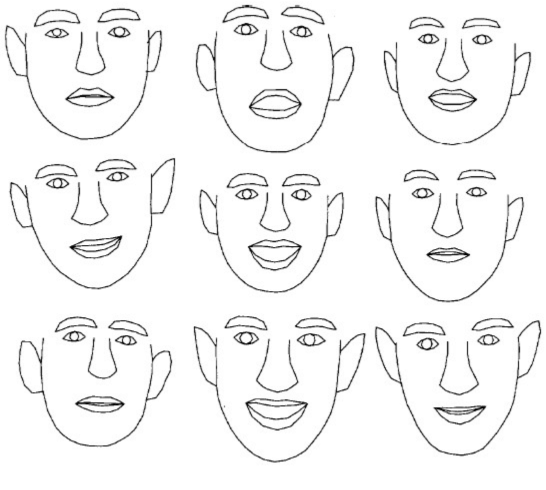


FIGURE 2.4 – Examples of training shapes [70].

We can apply a principal component analysis (PCA) to the data [24]. First, the average shape is calculated by :

$$\overline{\boldsymbol{x}} = \frac{1}{M} \sum_{i=1}^{M} \boldsymbol{x}_i \tag{2.2}$$

where M is the number of training examples.

Then, the covariance matrix of training data can be obtained using :

$$\boldsymbol{S} = \frac{1}{M} \sum_{i=1}^{M} d\boldsymbol{x}_i d\boldsymbol{x}_i^T \tag{2.3}$$

where $d\boldsymbol{x}_i$ is the deviation of $\boldsymbol{x}_i$ from the mean shape : $d\boldsymbol{x}_i = \boldsymbol{x}_i - \overline{\boldsymbol{x}}$.

The eigenvectors and eigenvalues are defined by :

$$\boldsymbol{S}\boldsymbol{p}_k = \lambda_k \boldsymbol{p}_k \tag{2.4}$$

where $\boldsymbol{p}_k$ and λ_k are respectively the kth eigenvector and eigenvalue of $\boldsymbol{S}$, and $\boldsymbol{p}_k^T \boldsymbol{p}_k = 1$.

In Equation (2.4), each eigenvector is related to a “mode of variation”, and the effect of the first three modes on face shape is shown in Figure 2.5. This way, a shape $\boldsymbol{x}$ in the training set can be approximated using the mean shape and a vector of weights :

$$\boldsymbol{x} = \overline{\boldsymbol{x}} + \boldsymbol{P}\boldsymbol{b} \tag{2.5}$$

where $\boldsymbol{P} = (\boldsymbol{p}_1 \boldsymbol{p}_2 ... \boldsymbol{p}_t)$ is the matrix of the first t eigenvectors with highest eigenvalues, and $\boldsymbol{b} = (b_1 b_2 ... b_t)$ is a vector of weights.

In [70], when fitting the shape model to a new face, a gray-level profile perpendicular to the boundary is extracted at each model point, and a new preferred position for the point is selected along the profile. According to the movement of landmarks, we can obtain the new shape parameters using the deformation of Equation (2.5) :

$$\boldsymbol{b} = \boldsymbol{P}^T (\boldsymbol{x} - \overline{\boldsymbol{x}}) \tag{2.6}$$

2.2.3 Active appearance model

Active appearance model (AAM) [22] is extended from the ASM described in Section 2.2.2. In order to distinguish the symbols, we rewrite the shape model in Equation (2.5) as :

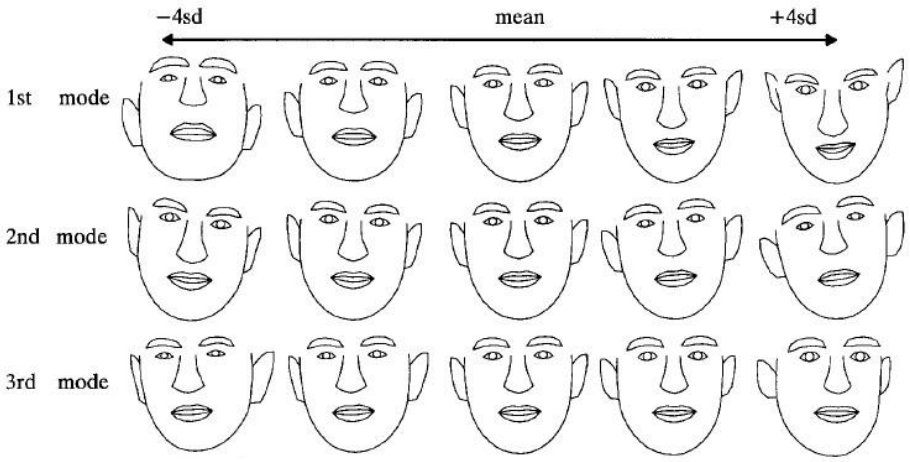


FIGURE 2.5 – The effect of the main modes of shape variation [70].

$$\boldsymbol{x} = \overline{\boldsymbol{x}} + \boldsymbol{P_s b_s} \tag{2.7}$$

where $\boldsymbol{P_s}$ and $\boldsymbol{b_s}$ are respectively the vector of shape variation modes and the vector of shape parameters.

Based on the shapes of training faces, we can obtain their *shape-normalized* images : $\boldsymbol{g}_i$, $i \in [1, M]$ (see some examples in Figure 2.6). Similar to the way of forming the shape model in Section 2.2.2, a PCA method is applied to these shape-normalized images, and we obtain a linear model :

$$\boldsymbol{g} = \overline{\boldsymbol{g}} + \boldsymbol{P_g b_g} \tag{2.8}$$

where $\boldsymbol{P_g}$ and $\boldsymbol{b_g}$ are respectively the vector of gray-level variation modes and the vector of gray-level parameters. Normally, to minimize the effect of global illumination variation, shape-free images are normalized before the PCA process.

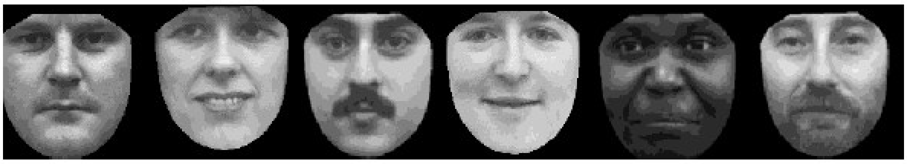


FIGURE 2.6 – Examples of shape-normalized images [70].

Combining both shape and gray-level models, a concatenated vector can be generated as follows :

$$\boldsymbol{b} = \begin{pmatrix} \boldsymbol{W_s b_s} \\ \boldsymbol{b_g} \end{pmatrix} = \begin{pmatrix} \boldsymbol{W_s P_s^T}(\boldsymbol{x} - \overline{\boldsymbol{x}}) \\ \boldsymbol{P}_g^T(\boldsymbol{g} - \overline{\boldsymbol{g}}) \end{pmatrix} \tag{2.9}$$

where $\boldsymbol{W_s}$ is a diagonal matrix of weights for shape parameters, allowing for the difference in units between the shape and gray-level models.

By applying PCA to these vectors, we can obtain a new model :

$$\boldsymbol{b} = \boldsymbol{Qc} \tag{2.10}$$

where $\boldsymbol{Q}$ is the matrix of eigenvectors and $\boldsymbol{c}$ is the vector of *appearance* parameters controlling both the shape and gray-level variations.

The effect of varying first four appearance model parameters is shown in Figure 2.7.

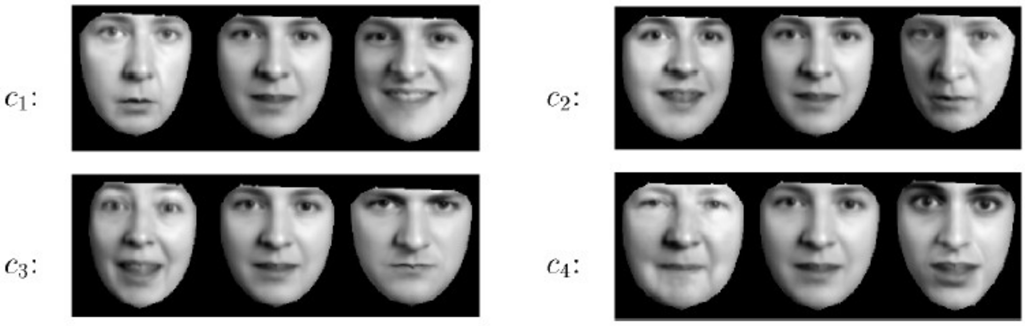


Figure 2.7 – The effect of varying first four appearance model parameters, $c_1 - c_4$ by ±3 standard deviations from the mean [22].

In [22], the AAM search for a new image is achieved by iteratively minimizing the difference between model texture and image.

2.2.4 Advantages and drawbacks

Numerous robust and efficient features, e.g., SIFT [82], POEM (Patterns Oriented Edge Magnitude) [120], can be used for keypoint detection. This advantage makes texture-based detectors remarkably robust. However, existing facial point detectors mostly rely on a priori knowledge of facial structure and on a specific training phase. That is, the construction of training data is complex which makes this type of methods not easy to implement.

The main advantage of shape-based algorithms is that they are able to locate multiple facial points (e.g., 68 points for the canonical AAM) under expression and pose changes. However, training of these statistical models requires numerous labeled images, therefore the application of model-based methods is complicated and computationally expensive. Moreover, faces extracted from video surveillance images usually of low resolution, e.g., 30×30. It is difficult to localize 68 facial landmarks in such a small region.

2.3 Direct face alignment

Direct face alignment approaches aim at warping a test facial image to match a pre-defined template (shown in Figure 2.8). Section 2.3.1 is an overview of direct face alignment methods, and Section 2.3.2 introduces the famous Lucas-Kanade's alignment methods [83, 10].

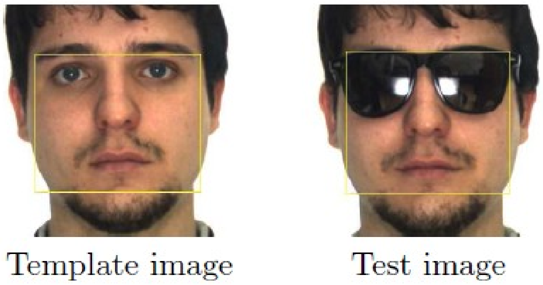


FIGURE 2.8 – Direct alignment tries to warp a test image regarding a pre-defined template.

2.3.1 Overview

Important components of direct face alignment include a similarity measure and an optimization method. More precisely, face alignment task is casted into the optimization of a cost function based on the similarity/difference between images.

2.3.1.1 Similarity measures

The most familiar similarity measure is the sum of squared differences (SSD) which has been widely used for image registration, e.g., [83, 10]. To make error metric more robust to outliers, robust functions such as Geman-McClure function [16] and the median of absolute differences (MAD) [105] are often used to replace the squared error terms. Also, weighted SSD function is proposed for image matching in [9]. Recently, [4] aligns two faces by minimizing the SSD based function in Fourier domain using Gabor filters. Apart from these intensity differences, another type of similarity measure is correlation. More precisely, images can be aligned by the maximization of correlation function. Common correlation functions include normalized cross-correlation, phase correlation [67], and normalized gradient correlation [112], et al. Among them, normalized gradient correlation has been proved robust to partial occlusion in face alignment application. Mutual information (MI) is also used to measure the correlation between a pair of images where face alignment can be achieved by maximizing the mutual information [118].

2.3.1.2 Optimization methods

The usual approach to the optimization of face alignment problem is gradient descent where transformation parameters are obtained by iteratively estimating an update. Gradient descent algorithms have several different formulations. While the classic work of Lucas and Kanade [83] estimates an additive increment to parameters, Shum and Szeliski [104] update parameters in a compositional manner. In [10], Baker and Matthews presented an overview of Lucas-Kanade's algorithms, and they proposed an inverse compositional formulation for intensity based image alignment. This formulation can yield a constant *Hessian* matrix which can be pre-computed (Detailed information is introduced in Section 2.3.2). Hence, the inverse compositional method can decrease the computational cost of the iterative estimation without any significant loss of efficiency. Recently, [35] employs this inverse Lucas-Kanade method for mutual information based face tracking.

2.3.2 Lucas-Kanade's methods

Lucas-Kanade's methods [83, 10] aim at minimizing the sum of squared error between the template image $\boldsymbol{T}$ and the test image $\boldsymbol{I}$:

$$\sum_{\boldsymbol{x}} [\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})) - \boldsymbol{T}(\boldsymbol{x})]^2 \tag{2.11}$$

where $\boldsymbol{x} = (x, y)$ represents the coordinates of a pixel, $\boldsymbol{W(x;p)}$ is the parameterized set of allowed warps, $\boldsymbol{p} = (p_1, p_2, ..., p_n)^T$ is a vector of transformation parameters.

2.3.2.1 Forward additive formulation

The minimization of Equation (2.11) is a non-linear optimization task. To obtain the desired transformation parameters, the classic Lucas-Kanade [83] assumes that a current estimate of $\boldsymbol{p}$ is known and then iteratively searches an increment $\boldsymbol{\Delta p}$, hence the cost function becomes :

$$\sum_{\boldsymbol{x}}[\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}+\boldsymbol{\Delta p}))-\boldsymbol{T}(\boldsymbol{x})]^2 \tag{2.12}$$

To solve the minimization problem, Equation (2.12) is first linearized by performing a first order Taylor expansion on $\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}+\boldsymbol{\Delta p}))$ (which supposes that $\boldsymbol{\Delta p}$ is a very small increment) :

$$\sum_{\boldsymbol{x}}[\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))+\nabla\boldsymbol{I}\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}\boldsymbol{\Delta p}-\boldsymbol{T}(\boldsymbol{x})]^2 \tag{2.13}$$

where $\nabla\boldsymbol{I}=(\frac{\partial\boldsymbol{I}}{\partial\boldsymbol{x}},\frac{\partial\boldsymbol{I}}{\partial\boldsymbol{y}})$ is the gradient of image $\boldsymbol{I}$, $\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}$ is the *Jacobian* of the warp. For example, the affine warp has the Jacobian :

$$\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}=\begin{pmatrix}x & 0 & y & 0 & 1 & 0\\ 0 & x & 0 & y & 0 & 1\end{pmatrix} \tag{2.14}$$

Then, we can calculate the partial derivative of the expression in Equation (2.13) with respect to $\boldsymbol{\Delta p}$:

$$\sum_{\boldsymbol{x}}[\nabla\boldsymbol{I}\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}]^T[\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))+\nabla\boldsymbol{I}\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}\boldsymbol{\Delta p}-\boldsymbol{T}(\boldsymbol{x})]=0 \tag{2.15}$$

Easily, we can obtain $\boldsymbol{\Delta p}$ by setting Equation (2.15) equal to zero :

$$\boldsymbol{\Delta p}=\boldsymbol{H}^{-1}\sum_{\boldsymbol{x}}[\nabla\boldsymbol{I}\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}]^T[\boldsymbol{T}(\boldsymbol{x})-\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))] \tag{2.16}$$

where $\boldsymbol{H}$ is the *Hessian* matrix :

$$\boldsymbol{H}=\sum_{\boldsymbol{x}}[\nabla\boldsymbol{I}\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}]^T[\nabla\boldsymbol{I}\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}] \tag{2.17}$$

At each iteration, $\boldsymbol{p}$ is updated using :

$$\boldsymbol{p}\leftarrow\boldsymbol{p}+\boldsymbol{\Delta p} \tag{2.18}$$

2.3.2.2 Inverse compositional formulation

Many researches (e.g., [104]) point out there is a huge computational cost in re-calculating the Hessian matrix at each iteration of the classic Lucas-Kanade algorithm. Hence, the inverse formulation has been developed to yield constant Hessian matrix which can be pre-computed. The inverse compositional algorithm aims at minimizing :

$$\sum_{\boldsymbol{x}}[\boldsymbol{T}(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{\Delta p})) - \boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))]^2 \tag{2.19}$$

The first order Taylor expansion on Equation (2.19) is :

$$\sum_{\boldsymbol{x}}[\boldsymbol{T}(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{0})) + \nabla\boldsymbol{T}\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}\boldsymbol{\Delta p} - \boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))]^2 \tag{2.20}$$

Similar to the previous forward additive formulation, we can obtain the increment $\boldsymbol{\Delta p}$:

$$\boldsymbol{\Delta p} = \boldsymbol{H}^{-1}\sum_{\boldsymbol{x}}[\nabla\boldsymbol{T}\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}]^T[\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})) - \boldsymbol{T}(\boldsymbol{x})] \tag{2.21}$$

where the Hessian matrix is :

$$\boldsymbol{H} = \sum_{\boldsymbol{x}}[\nabla\boldsymbol{T}\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}]^T[\nabla\boldsymbol{T}\frac{\partial\boldsymbol{W}}{\partial\boldsymbol{p}}] \tag{2.22}$$

It is clear that Equation (2.22) is constant across all iterations, hence we can pre-compute the Hessian matrix to reduce the computational cost.

The update process of $\boldsymbol{p}$ using $\boldsymbol{\Delta p}$ is :

$$\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}) \leftarrow \boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}) \circ \boldsymbol{W}(\boldsymbol{x};\boldsymbol{\Delta p})^{-1} \tag{2.23}$$

2.3.3 Advantages and drawbacks

The major advantage of direct face alignment methods is that they make use of all the image information, since they measure the contribution of every pixel in the image [107]. Nevertheless, direct face alignment methods require a pre-defined template which limits their applications. More precisely, templates related to the same subjects as the test images are often used to present experimental results, i.e., the identities of test images are already *known*. This seems not so logical in face recognition, because the final objective of face recognition is to identify the input facial images.

2.4 Joint face alignment

This section discusses joint alignment methods which work on an image *ensemble*. Section 2.4.1 presents a review of important joint face alignment algorithms. Background knowledge about two major joint alignment methods, Learned-Miller Congealing [71, 53] and Least Squares Congealing [26, 27], is introduced in Sections 2.4.2 and 2.4.3.

2.4.1 Overview

Be different from above feature point based and direct approaches, joint face alignment methods work by simultaneously aligning a set of facial images. Figure 2.9 illustrates an example of joint alignment process where original (unaligned) faces are located by Viola-Jones face detector [117].

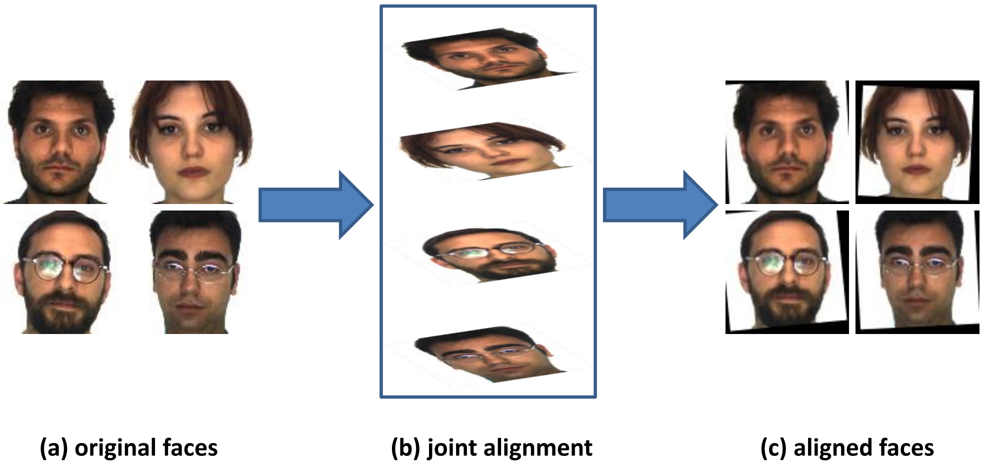


FIGURE 2.9 – Example : joint alignment of four facial images.

The early work of joint alignment [38, 39] aligned basis images using an expectation maximization (EM) algorithm and a finite set of allowable transformations. The so-called "congealing" method was first proposed by Learned-Miller for the alignment of binary images and magnetic resonance images [71, 89]. In congealing, the only assumption is the type of geometric mis-alignment, and transformation parameters are obtained by minimizing a sum-of-entropy function. Later, Congealing has been applied to more complex images, faces and cars [53], using scale-invariant feature transform (SIFT) [82] descriptor as the feature. Since SIFT descriptor are of high dimension, a dimension reduction stage is used to decrease the dimension of features in these entropy congealing methods [53].

As mentioned in Section 2.3, the classic Lucas-Kanade's algorithm [83] was proposed for iterative image-to-image alignment using gradient-descent optimization. Least Squares Congealing (LSC) [26, 27] is an extension of Lucas-Kanade's algorithm, in which entropy function is replaced with a SSD cost function and in which the transformation parameters are all estimated at the same time using a Gauss-Newton gradient descent approach. Storer et al. [85] aligned a set of images using mutual information (MI) as the measure of mis-alignment. Being both intensity-based, the two methods might be more sensitive to variations in illumination, scale, and occlusion. To improve the robustness of alignment, Liu et al. [79] applied HOG feature (histogram of oriented gradients [74]) to LSC, instead of pixel intensity. LSC methods were also combined with face models to localize facial landmarks [80, 108, 130].

Some researches pay attention to the rank of data matrix containing all images. More precisely, well-aligned images are supposed to be correlated to each other, hence the matrix of aligned images should have low rank. [116] minimizes a log-determinant measure that can be viewed as a smooth surrogate for the rank function. The work of [92] decomposes original facial images as a low-rank matrix with certain corruption and mis-alignment errors, then the alignment problem is solved by scalable convex optimization techniques. In [7, 125], mis-alignment parameters are simultaneous estimated during the learning of feature subspace. Another interesting joint alignment approach named RASL [93] aligns images by sparse and low-rank decomposition. RASL is able to handle both spatial mis-alignments and corruptions (e.g., occlusion and shadows), and it generates an aligned image and a low-rank (corruption recovered) image for each input sample. However, the reconstruction of low-rank images is valid only when all input images are related to the same individual. Therefore, if RASL is applied to unknown faces of different individuals, only the aligned images are useful.

2.4.2 Learned-Miller Congealing

In [71], congealing algorithm was proposed for transforming a set of images to make them more similar, according to some measure of similarity. This method can be defined as the minimization of a sum-of-entropy function :

$$arg \min_{\boldsymbol{\Phi}} - \sum_{\boldsymbol{x}} \sum_{j \in \boldsymbol{J}} p_j(\boldsymbol{x}) log_2(p_j(\boldsymbol{x})) \tag{2.24}$$

where $\boldsymbol{\Phi} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N\}$ is the set of warp parameter vectors for different images (N is the number of images), $\boldsymbol{x} = (x, y)^T$ is a column vector containing the pixel coordinates, $\boldsymbol{J}$ is the set of image features, e.g., intensity values, and $p_j(\boldsymbol{x})$ is the probability of feature j in the pixel set at $\boldsymbol{x}$.

For the alignment of binary images, the sum-of-entropy function can be calculated by :

$$-\sum_{\boldsymbol{x}}(\frac{N_0}{N}log_2\frac{N_0}{N} + \frac{N_1}{N}log_2\frac{N_1}{N}) \tag{2.25}$$

where N_0 and N_1 are respectively the number of occurrences of 0 (black) and 1 (white) in the binary-valued pixel set at $\boldsymbol{x}$. As illustrated in Figure 2.10, $\frac{N_0}{N}$ and $\frac{N_1}{N}$ respectively calculate the probabilities of black and white pixels at $\boldsymbol{x}$.

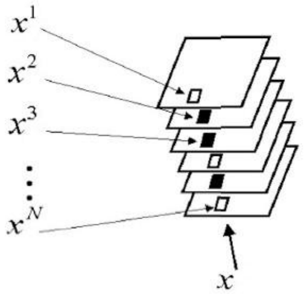


FIGURE 2.10 – A pixel set is a collection of pixels drawn from the same location in each of a set of N images. [71].

In the application to gray level images, Huang et al. [53] pointed out that the high variations of intensity in the foreground object as well as the variations due to lighting will cause high entropy even under a proper alignment. Therefore, instead of intensity values, some robust feature descriptors can be used with Congealing, e.g., SIFT descriptor. To reduce the dimension of feature space, pixel's descriptors are modeled as being generated from a mixture of Gaussians model by clustering. Hence, pixel at position $\boldsymbol{x}$ of the ith image can be represented by a vector of probabilities :

$$((p_1^i(\boldsymbol{x}), p_2^i(\boldsymbol{x}), ..., p_M^i(\boldsymbol{x})) \tag{2.26}$$

where M is the number of clusters, $p_k^i(\boldsymbol{x})$ is the probability that this pixel belongs to cluster k.

Then the sum-of-entropy function in Equation (2.24) can be rewritten as :

$$-\sum_{\boldsymbol{x}}\sum_{k}^{M} D_k(\boldsymbol{x})log_2(D_k(\boldsymbol{x})) \tag{2.27}$$

Here $D_k(\boldsymbol{x})$ is called "distribution field" which is the probability of the kth element in the pixel set at $\boldsymbol{x}$:

$$D_k(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} p_k^i(\boldsymbol{x}) \tag{2.28}$$

Considering Equations (2.27) and (2.28) together, for the ith image in the set, the alignment can be achieved by maximizing :

$$\sum_{\boldsymbol{x}}\sum_{k}^{M} p_k^i(\boldsymbol{x})log(D_k(\boldsymbol{x})) \tag{2.29}$$

In [53], the maximization of Equation (2.29) is based on an iterative sequential searching procedure. Each transformation parameter relates to a pre-defined step size and an estimation order. To estimate the update for one parameter, an increase and a decrease equaling to the relative step size are applied respectively. Only the variation that increases the value of Equation (2.29) is updated to the parameter. Otherwise, the parameter is not changed. After one parameter has been estimated, the estimation moves to the next. Figure 2.11 shows an illustration of congealing procedure in which the entropy value over aligned distribution field n is lower than that of original distribution field 1.

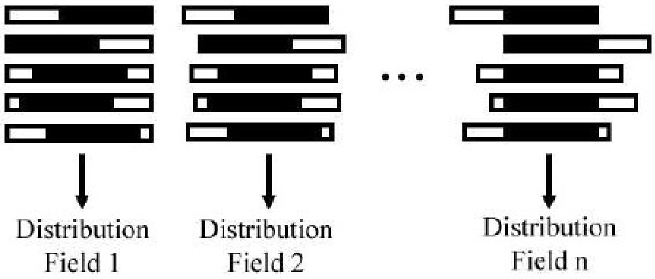


FIGURE 2.11 – An illustration of congealing procedure [53].

2.4.3 Least Squares Congealing

Least Squares Congealing (LSC) [26, 27] is based on the classic Lucas-Kanade's algorithm described in Section 2.3.2. In LSC, the alignment problem is defined as the minimization of a SSD function calculated over a set of images :

$$\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}[\boldsymbol{I}_j - \boldsymbol{I}_i]^2 \tag{2.30}$$

where M is the number of images, $\boldsymbol{I}_i$ stands for the ith image in the set.

Hence, for one image $\boldsymbol{I}_i$ in the set, we aim at estimating the transformation which minimizes :

$$\sum_{\substack{j=1\\j\neq i}}^{M}[\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))_j - \boldsymbol{I}_i]^2 \tag{2.31}$$

Similar to the image-to-image Lucas-Kanade's algorithm, we can iteratively solve the optimization problem in Equation (2.31) by calculating an update Δp :

$$\sum_{\substack{j=1\\j\neq i}}^{M}[\boldsymbol{I_j}(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})) + \frac{\partial \boldsymbol{I_j}(\boldsymbol{p})}{\partial \boldsymbol{p}}^T \boldsymbol{\Delta p} - \boldsymbol{I_i}]^2 \tag{2.32}$$

where $\frac{\partial \boldsymbol{I_j}(\boldsymbol{p})}{\partial \boldsymbol{p}}$ is the steepest descent image : $\sum_{\boldsymbol{x}} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}} \nabla \boldsymbol{I_j}$.

Then, we can obtain Δp using :

$$\boldsymbol{\Delta p} = \boldsymbol{H}^{-1}\sum_{\substack{j=1\\j\neq i}}[\frac{\partial \boldsymbol{I_j}(\boldsymbol{p})}{\partial \boldsymbol{p}}(\boldsymbol{I_j}(\boldsymbol{p}) - \boldsymbol{I_i})] \tag{2.33}$$

where $\boldsymbol{H}$ is the *Hessian* matrix :

$$\boldsymbol{H} = \sum_{\substack{j=1\\j\neq i}} \frac{\partial \boldsymbol{I_j}(\boldsymbol{p})}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{I_j}(\boldsymbol{p})}{\partial \boldsymbol{p}}^T \tag{2.34}$$

At each iteration, $\boldsymbol{p}$ is updated to the rest of images $\boldsymbol{I}_j$ using :

$$\boldsymbol{p}_j \leftarrow \boldsymbol{p}_j + \boldsymbol{\Delta p}, where\ j \neq i \tag{2.35}$$

This is a forward additive formulation of Least Squares Congealing, an inverse compositional solution is also introduced in [27] to deal with some specific problems, i.e., inequality between additive and compositional updates and object loss of outlier images.

2.4.4 Advantages and drawbacks

The main advantage of joint alignment methods is that they are able to simultaneously align a set of images without a biased template selection. Hence, this type of methods can be used for aligning unknown faces in a face recognition application. Moreover, many joint alignment methods, e.g., Learned-Miller Congealing (LMC), Least Squares Congealing (LSC), work in an unsupervised manner. Hence, they are easy to implement. However, there are certain limitations in current existing joint alignment methods. For example, LMC needs of pre-defined step size of updates and the estimation of parameters is sequential. This results in slow convergence ability, and this method is easily to get stuck in undesired local minima. LSC is sensitive to variations in illumination, scale, and occlusion, because it is an intensity based method.

2.5 Face databases

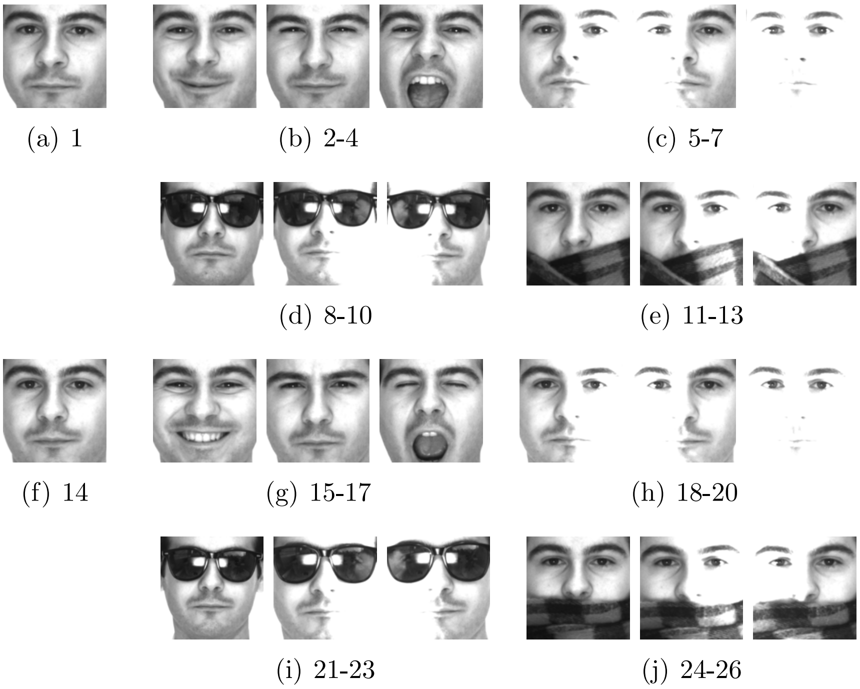
This section introduces several famous face databases on which we conduct comparison experiments in the following chapters. We divide these face databases into two groups : "controlled" and "uncontrolled". The controlled databases, including AR face [87], FERET [95], and Yale B [40], are used for "ideal performance evaluation". The uncontrolled databases, containing SCface [42] and LFW (labeled Faces in the Wild) [54], are more challenging for face alignment and more suited to mimic a video surveillance environment.

2.5.1 Controlled databases

2.5.1.1 AR face database

AR database [87] was created by Aleix Martinez and Robert Benavente in the Computer Vision Center (CVC) at the U.A.B. This database contains over 4,000 frontal facial images of 126 people (70 men and 56 women) with different facial expressions, lighting conditions, and occlusions. These pictures were taken under strictly controlled conditions. No restrictions on wear (clothes, glasses, etc.), make-up, hair style, etc. were imposed to participants. Figure 2.12 shows examples of AR face images related to one person under different conditions. Each person participated in two sessions. Images in Session 1 were taken under different conditions :

neutral (Figure 2.12 (a)), expressions (Figure 2.12 (b)), illumination (Figure 2.12 (c)), sun glasses + illumination (Figure 2.12 (d)), scarf + illumination (Figure 2.12 (e)). Session 2 consists of images taken under the same conditions as in Session 1 but separated by two weeks (Figure 2.12 (f)-(j)). The images in this database are all labeled with three feature points (two eyes and nose tip), and a part of images provide manual annotations of 22 facial landmarks (see Figure 2.13). These landmarks can be used for performance evaluation.


(a) 1 (b) 2-4 (c) 5-7

(d) 8-10 (e) 11-13

(f) 14 (g) 15-17 (h) 18-20

(i) 21-23 (j) 24-26

FIGURE 2.12 – Examples of AR face images related to one individual.

2.5.1.2 FERET

FERET database [95] is a standard dataset used for facial recognition system evaluation. The related face recognition technology program is managed by the Defense Advanced Research Projects Agency (DARPA) and the National Institute of Standards and Technology (NIST), USA. This database has facial images related to more than 1,000 subjects. Each individual is related to 5 to 11 images. Figure 2.14 shows example images related to one individual : two frontal view images with

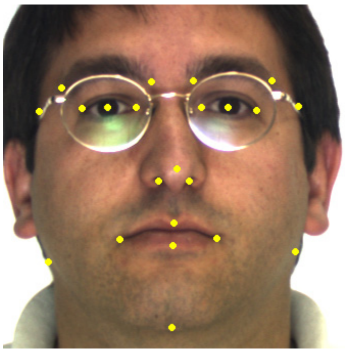


FIGURE 2.13 – 22 landmarks manually labeled in the AR face images.

different facial expressions ("fa" and "fb"), one image ("fc") taken with a different camera and different lighting condition, and two images taken after certain intervals (duplicate).

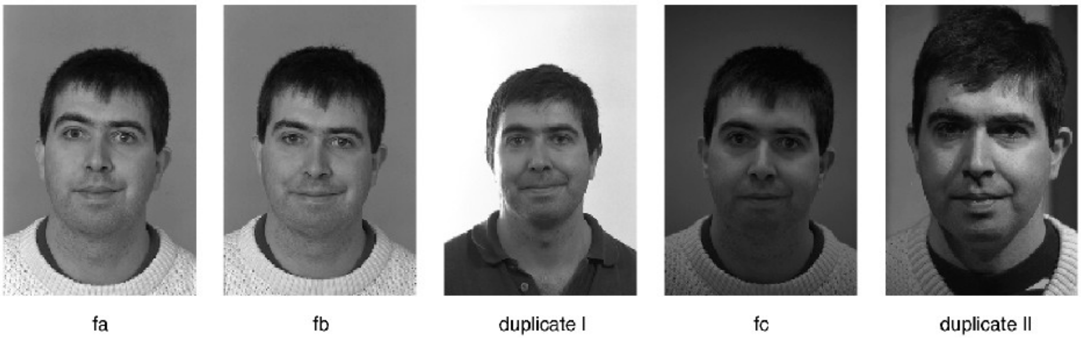


FIGURE 2.14 – Examples of different types of image in FERET database. The duplicate I image was taken within one year of the "fa" image and duplicate II was taken at least one year of the "fa" image.

2.5.1.3 Yale B

Yale B database [40] provides 5,760 single light source images related to 10 subjects each seen under 576 viewing conditions (64 lighting conditions × 9 poses). For each pose, the 64 images are divided into five different subsets according to the direction of lighting source under which the considered image is taken (see some examples in Figure 2.15). Regarding this division, the images belonging to subset 1 is quite easy (the illumination is equally frontal) whilst subsets 4 and 5 are very challenging.

Also, provided in the Yale B database, for each face, are the coordinates of three landmarks : two eye centers and mouth center.

2.5.2 Uncontrolled databases

2.5.2.1 SCface

SCface database [42] was designed mainly as a means of testing face recognition algorithms in real-world conditions. This database contains 4160 static facial images taken in uncontrolled indoor environment using several video surveillance cameras of various qualities. Some example images of one person taken by different cameras are shown in Figure 2.16. For each camera, images were taken with different distances and viewpoints. This database also provides the coordinates of two eye centers, nose tip, and mouth for each face. In other words, there are 4 landmarks per image for the evaluation on this database.

2.5.2.2 LFW (Labeled Faces in the Wild) database

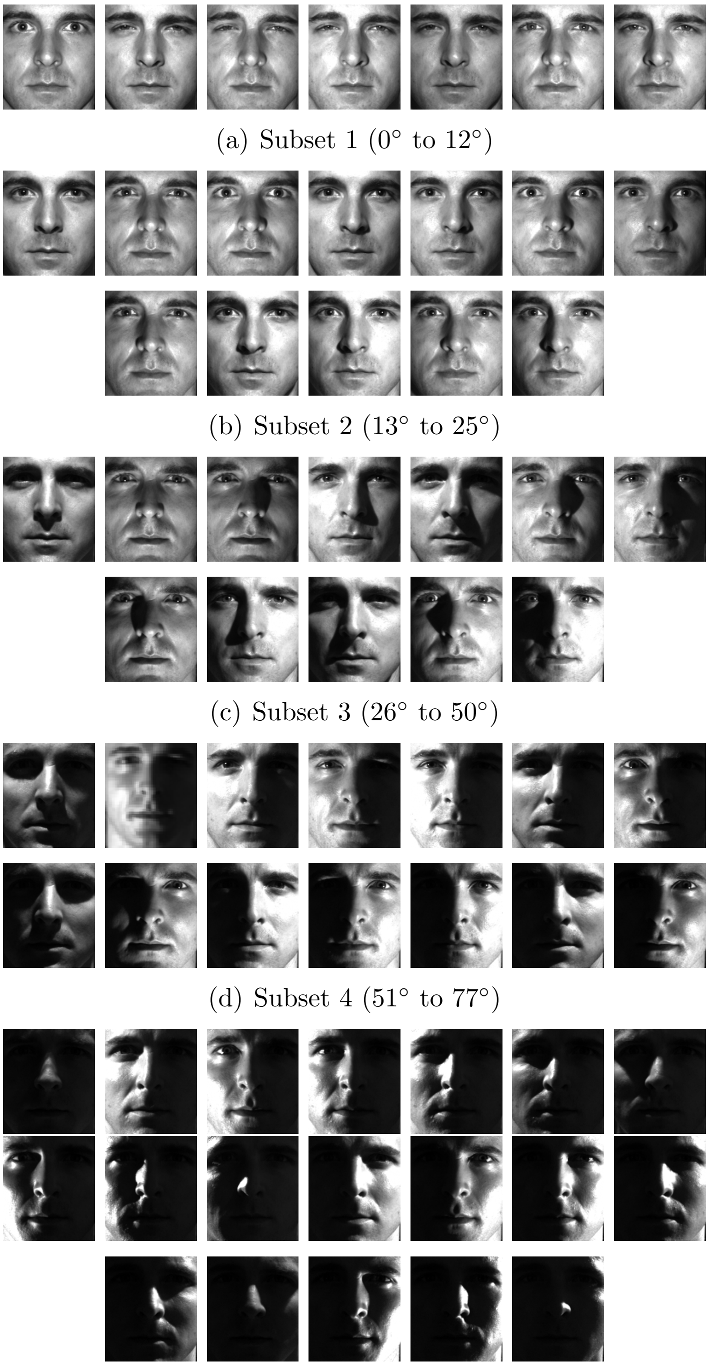
LFW database [54] contains more than 13,000 images of faces collected from the web. Each face has been labeled with the name of the person pictured. 1680 of the people pictured have two or more distinct photos in the data set. The only constraint on these faces is that they were detected by the Viola-Jones face detector. LFW database is challenging for face recognition research because faces in this database undergo uncontrolled variations. The operational goal of this set is to study the problem of pair matching (given two facial images, decide whether they are related to the same individual or not). Examples of matched pairs from this database are shown in Figure 2.17.

2.6 Evaluation criteria for face alignment

This section discusses the main solutions used in literature to evaluate the performance of face alignment methods. We will also employ some of these evaluation criteria to verify the ability of our methods in the following chapters.

1. Convergence rate of landmarks

During the alignment process, the new positions of landmarks in cropped faces can be obtained using the estimated transformation parameters. If the Euclidean distance between a landmark and its relevant groundtruth is smaller than a predefined threshold, the landmark is taken as converged. Hence, the convergence rate



(a) Subset 1 (0° to 12°)

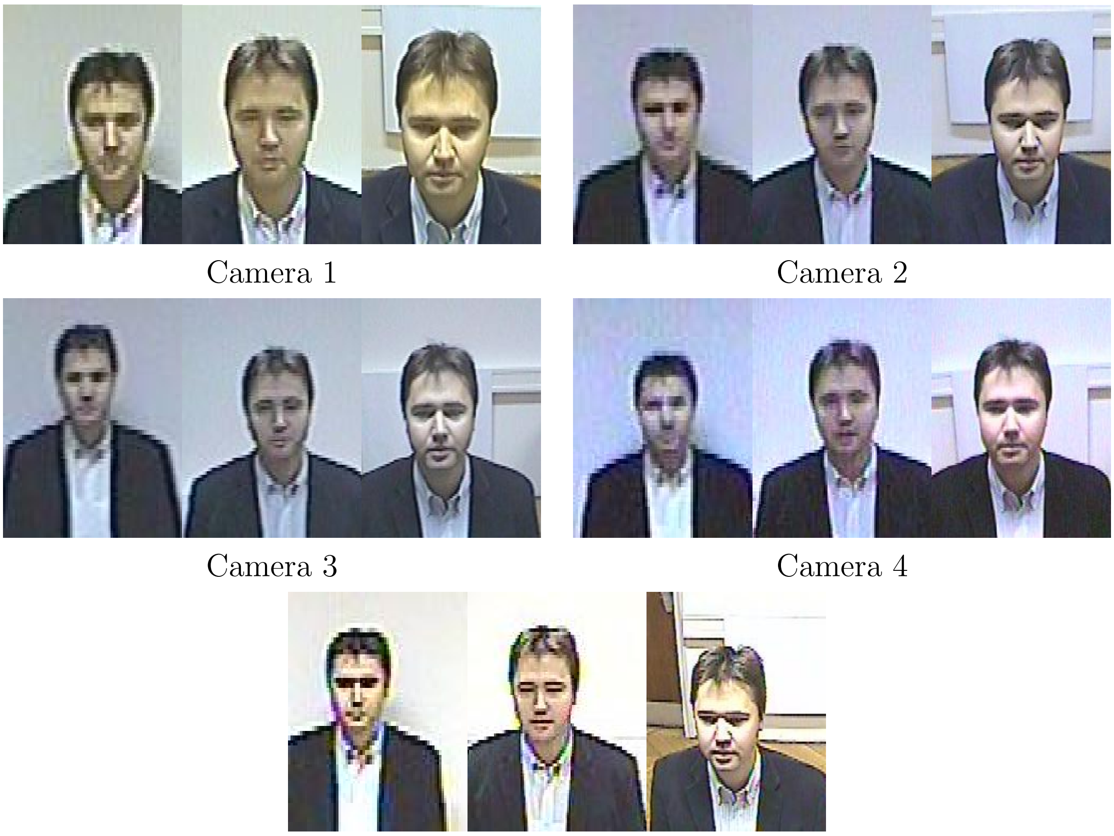
(b) Subset 2 (13° to 25°)

(c) Subset 3 (26° to 50°)

(d) Subset 4 (51° to 77°)

(e) Subset 5 (more than 78°)

FIGURE 2.15 – Examples of images from one individual under different conditions.



Camera 1

Camera 2

Camera 3

Camera 4

Camera 5

FIGURE 2.16 – Examples of images from SCface database.

can be calculated by :

$$rate = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N_l} \left(\left\| \boldsymbol{x}_j^i - \widetilde{\boldsymbol{x}}_j^i \right\| < t \right)}{M \times N_l}, \tag{2.36}$$

where $\boldsymbol{x}_j^i$ and $\widetilde{\boldsymbol{x}}_j^i$ respectively represent the new coordinates and the groundtruth of landmark j from the ith image, M and N_l are respectively the numbers of test images and of landmark points, t stands for the threshold.

Convergence rates can be reported in different ways : e.g., with respect to iterations, thresholds, and deviation of perturbations. The detailed information will be introduced in the experiment settings.

2. Average of cropped faces

Since the goal of alignment is to transform faces into a standard pose, the average

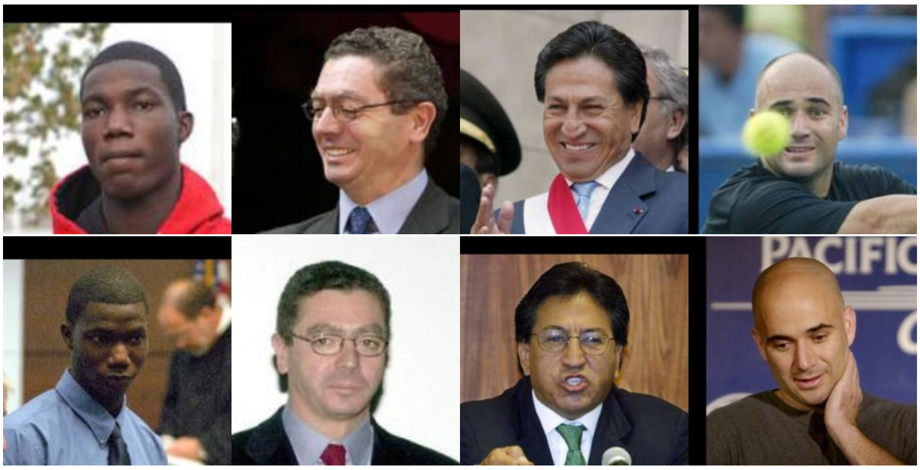


FIGURE 2.17 – Examples of matched image pairs from LFW database.

of the cropped (aligned) faces should have clear facial features. Therefore, the average images before and after alignment are calculated and compared. This criterion is usually used for the evaluation of joint alignment methods. Figure 2.18 is an example of averages images calculated from unaligned and aligned faces.

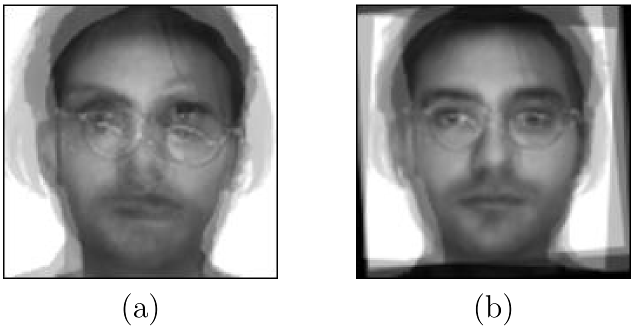


FIGURE 2.18 – Gray-level average images of (a) unaligned faces and (b) aligned faces.

3. Improvement in face recognition

To verify the benefits of alignment, face recognition rates on original and aligned images can be respectively calculated and compared.

4. Complexity

The computational cost or running speed of face alignment algorithm is also important, especially for real-time applications.

2.7 Conclusion

This chapter mainly introduces classic and recent works about face alignment. However, the results of most existing face alignment methods are presented on high quality still images taken under controlled conditions, and these methods usually do not perform well in uncontrolled environments where images undergo low resolution, motion blur, noise, occlusion, large variations in pose, expression, and illumination. All these challenging factors exist in our target data which are video frames from surveillance cameras. That is, there is still a lot of work to do to improve the performance of face alignment in video surveillance context.

In this chapter, image alignment algorithms are divided into three categories, i.e., feature point based, direct alignment, and joint alignment. Also, feature point based methods consist of texture-based methods and shape-based methods. In this thesis, we do not work on shape-based methods because they are too complicated and computationally expensive for video surveillance applications. Since direct face alignment methods are not suitable for face recognition applications due to the requirement of a pre-defined template, we do not adopt this type of methods in our work. Hence, in this thesis we focus on the study of texture-based point detection and joint face alignment. While existing texture-based detectors are mostly based on a priori knowledge of facial structure and a training phase, the first part of our work presented in Chapter 3 is an online approach without requirements of pre-defined constraints on feature distributions. Then, we respectively propose two new joint alignment methods in Chapters 4 and 5 to cope with the challenges for video-based face alignment. Our methods circumvent many of the limitations existing in the previous joint alignment methods. Moreover, in Chapter 6 we combine joint alignment with a face tracker for accurate face extraction.

Chapitre 3

An Online Three-stage Method for Facial Point Localization

3.1 Introduction

An intuitional solution to mis-alignment is detecting facial feature points and then transforming them to standard positions. As discussed in Chapter 2, the main advantage of this type of methods is that numerous robust and efficient features can be adopted for accurate keypoint detection. Hence, this chapter focuses on the problem of facial point localization.

Finding facial features respectively under expression and illumination variations is always a difficult problem. One popular solution for improving the localization performance is to use the spatial relation between facial feature positions. Existing algorithms mostly rely on a priori knowledge of facial structure and on a training phase. In [29, 115], pairwise spatial relations between facial point positions are learned for detection. With the knowledge of facial feature distributions, [121] divides faces into several regions of interest (ROI), then individual feature patch templates are used to detect points in the relevant ROI. Ding et al.[32] first localize two eyes and estimate the approximate positions of other features with a priori knowledge about face. However, the construction of training data for these methods is complicated and computationally expensive.

To circumvent the training stage, this chapter describes an online approach without requirements of pre-defined constraints on feature distributions. Instead of training specific detectors for each facial feature, a generic method is first used to extract a set of interest points from test images. With a robust feature descriptor named Patterns Oriented Edge Magnitude (POEM) histogram [119], a smaller set of these points are picked as candidates. Then we apply a game-theoretic technique to select facial points from the candidates, while the global geometric properties of face are well preserved. The experimental results demonstrate that our method achieves satisfactory performance on face images under expression and lighting variations.

The remainder of this chapter is arranged as follows : Section 3.2 first discusses four invariant feature detectors. Background information about POEM descriptor and game-theoretic matching is respectively introduced in Sections 3.3 and 3.4. The details of our online three-stage method for facial point localization are discussed in 3.5. The experimental results are analyzed in Section 3.6 and conclusions are given in Section 3.7.

3.2 Invariant Feature Detectors

Facial images contain some basic types of features : blobs (e.g., eye centers and nostrils) and corners (e.g., eye corners and mouth corners). To avoid training specific facial point detectors which are not easy to implement, here we adopt generic invariant feature detectors to locate these keypoints in faces. According to the studies in [88], we choose four efficient invariant feature detectors including Laplacian-of-Gaussian (LoG) [77], Difference-of-Gaussian (DoG) [81], Harris-Laplacian [88], and Hessian-Laplacian [88] for further evaluation on faces. These algorithms aim at finding interest points invariant to scale, rotation, and translation as well as not sensitive to changes of illumination and viewpoint. Being different from detectors specifically trained for facial features, these methods work in an online manner.

3.2.1 Laplacian-of-Gaussian (LoG)

In [77], the Laplacian-of-Gaussian (LoG) operator is used to detect blob-like features. An input image $I(x, y)$ ((x, y) is the vector of coordinates) is first smoothed by a Gaussian filter :

$$g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{3.1}$$

where σ is the Gaussian smoothing parameter.

Then we have a scale-space representation :

$$L(x, y, \sigma) = g(x, y, \sigma) * I(x, y) \tag{3.2}$$

The Laplacian operator is defined by :

$$\nabla^2 L = L_{xx}(x, y, \sigma) + L_{yy}(x, y, \sigma) \tag{3.3}$$

where L_{xx} and L_{yy} respectively denote second derivative of L in direction x and y.

In order to obtain a multi-scale blob detector with automatic scale selection, the scale-normalized Laplacian operator is used :

$$\nabla^2_{norm} L = \sigma^2 (L_{xx}(x, y, \sigma) + L_{yy}(x, y, \sigma)) \tag{3.4}$$

Based on Equation (3.4), interest points are found by attaining the local maxima of :

$$\left|\nabla^2_{norm} L\right| = \sigma^2 \left|(L_{xx}(x, y, \sigma + L_{yy}(x, y, \sigma))\right| \tag{3.5}$$

3.2.2 Difference-of-Gaussian (DoG)

The computational cost of Equation (3.5) is relatively high since it needs to estimate second derivatives in x and y direction respectively. Later, Lowe [82] proved that :

$$\nabla^2_{norm} L \approx \frac{L(x, y, k\sigma) - L(x, y, \sigma)}{k - 1} \tag{3.6}$$

where k is a constant multiplicative factor.

Hence, a cheaper way to approximate LoG is the use of Difference-of-Gaussian (DoG) [81] :

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{3.7}$$

As shown in Figure 3.1, local extrema of DoG images are detected by comparing a pixel (marked with X) to its 26 neighbors at the current and adjacent scales (circles). These extrema determine the location and scale of interest points.

3.2.3 Harris-Laplacian detector

Harris-Laplace detector [88] consists of two steps : a multi-scale point detection and a scale selection.

1. Multi-scale initial point detection

Harris-Laplacian detector is developed from the algorithm of Harris and Stephens [47] which is mainly based on a second moment matrix defined by :

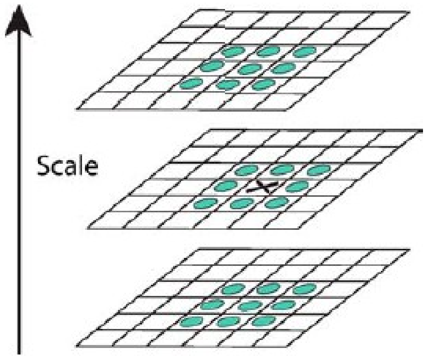


FIGURE 3.1 – Detection of local extrema across DoG images [82].

$$\mu(x, y, \sigma_I, \sigma_D) = \sigma_D^2 g(x, y, \sigma_I) * \begin{bmatrix} L_x^2(x, y, \sigma_D) & L_x L_y(x, y, \sigma_D) \\ L_x L_y(x, y, \sigma_D) & L_y^2(x, y, \sigma_D) \end{bmatrix} \tag{3.8}$$

where σ_I is the integration scale, σ_D is the differentiation scale.

The selection function is defined by :

$$cornerness = det(\mu(x, y, \sigma_I, \sigma_D)) - \alpha trace^2(\mu(x, y, \sigma_I, \sigma_D)) \tag{3.9}$$

where α is a constant.

Location of initial points at different scales are obtained by calculating local maxima of Equation (3.9).

2. Scale selection

Scale selection has been studied in [77], the idea is to find the *characteristic scale* of a point, for which a given function attains an extremum over scales. To this purpose, LoG function in Equation (3.5) is often used for scale selection. The work of [88] verify for each initial points whether the LoG attains a maximum at the scale of the point. The points for which LoG attains no extremum or the response is lower than a threshold are rejected.

3.2.4 Hessian-Laplacian detector

Hessian-Laplacian detector [88] is based on Hessian matrix which is second order derivatives. The Hessian matrix for point at (x, y) is defined by :

$$H(x,y,\sigma) = \begin{bmatrix} L_{xx}(x,y,\sigma) & L_{xy}(x,y,\sigma) \\ L_{xy}(x,y,\sigma) & L_{yy}(x,y,\sigma) \end{bmatrix} \tag{3.10}$$

Initial interest points are localized spatially by finding local extrema of both the determinant and trace of Equation (3.10) :

$$det(H(x,y,\sigma)) = \sigma^2(L_{xx}(x,y,\sigma)L_{yy}(x,y,\sigma) - L_{xy}^2(x,y,\sigma)) \tag{3.11}$$

$$trace(H(x,y,\sigma)) = \sigma(L_{xx}(x,y,\sigma) + L_{yy}(x,y,\sigma)) \tag{3.12}$$

We can notice that the trace of Hessian matrix in Equation (3.12) is identical to the LoG in Equation (3.4). The scale selection process of Hessian-Laplacian detector is similar to that of Harris-Laplacian algorithm in Section 3.2.3.

3.3 POEM : Patterns of Oriented Edge Magnitudes

POEM (Patterns of Oriented Edge Magnitudes) is a robust feature descriptor which catches both the edge information of local patches and the relation between this information in a neighboring region. POEM was first used in face recognition [119, 120], and has shown its ability for a good representation of facial features. The main steps of calculating POEM descriptor are (see Figure 3.2) :

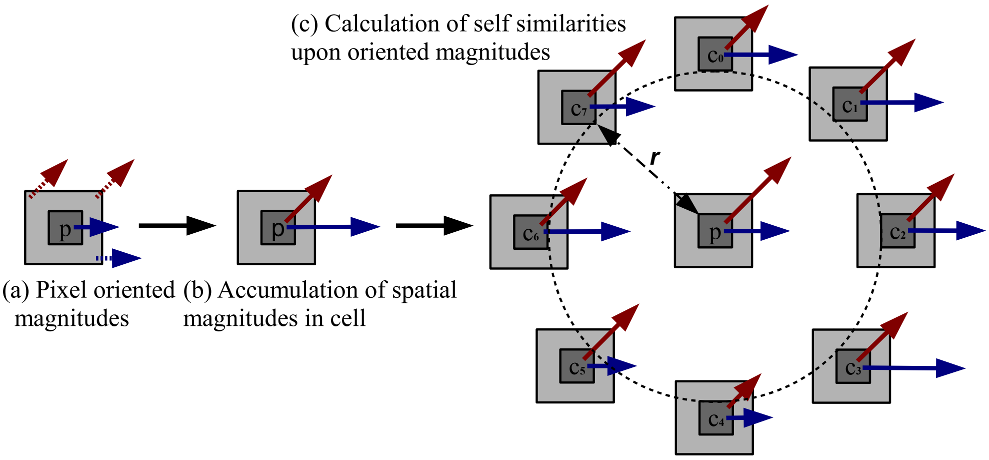

(c) Calculation of self similarities upon oriented magnitudes
C0
C1
C2
C3
C4
C5
C6
C7
r
p
p
p
(a) Pixel oriented magnitudes
(b) Accumulation of spatial magnitudes in cell

FIGURE 3.2 – Main steps of POEM extraction [119].

(1) The first step is the calculation of the gradient image. The gradient orientation of each pixel is then evenly discretized over 0-π ("unsigned" type) or 0-2π ("signed" type). Hence, for each pixel, the gradient is a 2-dimension vector with its original magnitude and its discretized direction. Take the pixel p in Figure 3.2 (a) for example, the length and direction of the continuous arrow emitting from p respectively represent the magnitude and the discretized direction.

(2) Magnitude Accumulation. For a pixel p, a local histogram of gradients over all pixels within a cell centered on p (the light region in Figure 3.2 (a)) is calculated and assigned to p. Vote weights are the gradient magnitude and the number of bins equals to the number of orientations. As shown in Figure 3.2 (b), the feature is now a vector of m values where m is the number of discretized directions, and each value stands for the accumulated magnitude in one direction.

(3) Computation of self-similarity-based operator. At each orientation θ_i, the magnitude at pixel p is compared with l surrounding pixels (e.g., C_0 - C_7 in Figure 3.2 (c)) in a radius r :

$$POEM^{\theta_i}(p) = \sum_{j=1}^{l}(I_p^{\theta_i} - I_{c_j}^{\theta_i} > \tau)2^j \tag{3.13}$$

where $I_p^{\theta_i}$, $I_{c_j}^{\theta_i}$ are the magnitudes of central and surrounding pixels p, c_j, and the threshold τ is 0.2. This procedure is similar to the calculation of LBP operator [90], more information can be found in Appendix B.

So for each pixel, there will be a set of m values :

$$POEM(p) = \left\{POEM^{\theta_1}(p), ..., POEM^{\theta_m}(p)\right\} \tag{3.14}$$

(4) Finally, for a pixel, we calculate m histograms of POEM (one for each orientation) over a small window, centered on that pixel. These m histograms are concatenated and used as the feature descriptor of the considered pixel.

As pointed out in [119], POEM presents the following properties : POEM (1) is an oriented, spatial multi-resolution descriptor capturing rich information about the original image, (2) is a multi-scale self-similarity based structure that results in robustness to exterior variations, (3) is of low complexity.

3.4 Game-theoretic matching

This section discusses a game-theoretic matching approach where the matching problem is cast into a strategic game. Section 3.4.1 introduces the basic knowledge about game theory, and Section 3.5 presents the details of matching as a strategic game.

3.4.1 Game theory

3.4.1.1 Traditional game theory

Game theory is a method of studying the mathematical models of conflict and cooperation between intelligent rational decision-makers. In a traditional game, decision-makers pursue well-defined exogenous objectives (they are *rational*) and take into account their knowledge or expectations of other decision-makers' behavior (they reason *strategically*) [91]. Game theory is mainly used in economics, political science, and psychology, as well as logic and biology.

In a strategic game, *players* compete against with their adversaries by selecting a *strategy* from an allowed set, and each player will get a *payoff* defined by the relationship between two plays' choices. Table 3.1 describes a canonical example of a two-player game, "*Prisoner's dilemma*" : Two suspects in a crime are put into separate cells. If they both confess, each will be sentenced to 2 years in prison. If only one of them confesses, he will be freed and the other will receive a sentence of 3 years. If both stay silent, each of them will only need to serve 1 year in prison. Here two prisoners can be taken as players, (*stay silent*, *confess*) is the set of available strategies (or *pure strategies* in the language of game theory), and the results of sentence are payoffs.

	Prisoner 2 stays silent	Prisoner 2 confesses
Prisoner 1 stays silent	Each serves 1 year	Prisoner 1 : 3 year Prisoner 2 : goes free
Prisoner 1 confesses	Prisoner 1 : goes free Prisoner 1 : 3 year	Each serves 2 years

Table 3.1 – The prisoners' dilemma.

A game is called *symmetric* when the payoffs for playing a particular strategy depend only on the other strategies used, not on the players. That is, if the payoff is invariant to the change of players' identities, then a game is symmetric. For example, Prisoner's dilemma in Table 3.1 is a symmetric game.

If each player has chosen a strategy and no player can benefit by changing his/her strategy while the other players keep theirs unchanged, then the current set of strategy choices and the corresponding payoffs constitute a *Nash equilibrium.* For example, the game in Table 3.1 has a unique Nash equilibrium (*confess*, *confess*) because whatever one player does, the other prefers *confess* to *stay silent.*

Here, we define a pure strategy set $O = \{1, 2, ..., n\}$, and a related $n \times n$ payoff matrix $C = (c_{ij})$ where c_{ij} $(i, j \in O)$ represents the payoff of an individual playing strategy i against one choosing strategy j. In contrast to pure strategy, a *mixed strategy* is a probability distribution $\boldsymbol{x} = (x_1, ..., x_n)^T$ over the strategy set O. This means that the plays' actions are not deterministic but regulated by probabilistic rules. Clearly, mixed strategies lie in a n-dimensional space :

$$\Delta = \left\{ \boldsymbol{x} \in \mathbb{R}^n : \sum_{i=1}^{n} x_i = 1 \; and \; x_i \geq 0, i = 1, ..., n \right\} \tag{3.15}$$

The *support* of a mixed strategy $\sigma(\boldsymbol{x}) = \{i \in O | x_i > 0\}$ defines the set of elements with non-zero probability. If a player plays pure strategy i against a mixed strategy x, the payoff will be :

$$(C\boldsymbol{x})_i = \sum_j c_{ij} x_j \tag{3.16}$$

Hence, the expected payoff by adopting a mixed strategy $\boldsymbol{y}$ against $\boldsymbol{x}$ is $\boldsymbol{y}^T C \boldsymbol{x}$. The *best replies* against a mixed strategy x are :

$$\beta(\boldsymbol{x}) = \left\{ y \in \Delta : y^T C x = max_z z^T C x \right\} \tag{3.17}$$

A mixed strategy $\boldsymbol{x}$ is a Nash equilibrium if it is the best reply to itself, i.e. :

$$\forall \boldsymbol{y} \in \Delta, \boldsymbol{y}^T C \boldsymbol{x} \leq \boldsymbol{x}^T C \boldsymbol{x} \tag{3.18}$$

An important property of mixed strategy Nash equilibria $\boldsymbol{x}$ is that [91] :

Theorem 3.1 *Every pure strategy i in the support of a mixed strategy Nash equilibrium $\boldsymbol{x}$ is a best response to $\boldsymbol{x}$.*

Theorem 3.1 can be proofed as follows : (1) Suppose that there is a pure strategy i in $\sigma(\boldsymbol{x})$ that is not a best response to $\boldsymbol{x}$. Then a player can increase his/her payoff by transferring probability from i to a pure strategy that is a best response ; hence $\boldsymbol{x}$ is not a best response to itself. (2) Suppose that a mixed strategy $\boldsymbol{x}'$ that gives a higher payoff than does $\boldsymbol{x}$ in response to itself. Then at least one pure strategy in support of $\boldsymbol{x}'$ must give higher payoff than some strategies in $\sigma(\boldsymbol{x})$, so that not all strategies in $\sigma(\boldsymbol{x})$ are best responses to $\boldsymbol{x}$. This implies that :

Corollary 3.1 *For every pure strategy i in the support of a mixed strategy Nash equilibrium $\boldsymbol{x}$ yields the same payoff $(C\boldsymbol{x})_i = \boldsymbol{x}^T C\boldsymbol{x}$, while all strategies outside the support of $\boldsymbol{x}$ earn a payoff that is less than of equal $\boldsymbol{x}^T C\boldsymbol{x}$.*

3.4.1.2 Evolutionary game theory

Considering an idealized scenario where pairs of players repeatedly drawn at random from a large population to participate in a symmetric two-player game (i.e., a continuous iterative process), this is the conception of *evolutionary game theory.* Be different from traditional game theory, players are not supposed to behave rationally or to have complete knowledge of the details of the game. Players either inherit modes of behavior from their forebears or are assigned them by mutation. During the evolutionary process, only the strategies with high support will survive, and strategies with low support are driven to extinction.

In evolutionary game theory, a strategy is said to be an *evolutionary stable strategy* (ESS) if it is a Nash equilibrium and :

$$\forall \boldsymbol{y} \in \Delta, \boldsymbol{x}^T C\boldsymbol{x} = \boldsymbol{y}^T C\boldsymbol{x} \Rightarrow \boldsymbol{x}^T C\boldsymbol{y} > \boldsymbol{y}^T C\boldsymbol{y} \tag{3.19}$$

3.4.2 Matching as a strategic game

In [2], the matching problem is cast into a strategic game. Let A_1 and A_2 be the two sets of features for matching, hence the set of candidate feature pairs is $\mathbb{A} = A_1 \times A_2$. Assuming that there are n candidate pairs and a $n \times n$ matrix C can be calculated for measuring the compatibilities between candidate pairs. Our purpose is to find a subset $\mathbb{A}_{match} \subseteq \mathbb{A}$ which contains only matched pairs. This match subset should satisfy two criteria : *high internal compatibility*, i.e., pairs belong to $\mathbb{A}_{match}$ are mutually highly compatible, and *low external compatibility*, i.e., pairs outside $\mathbb{A}_{match}$ are scarcely compatible with those inside.

To this point, we can find that the match subset have similar characteristics to Nash equilibria which satisfy both the internal and external compatibility criteria. More precisely, in Corollary 3.1, for any strategy $i \in \sigma(\boldsymbol{x})$, $(C\boldsymbol{x})_i = \boldsymbol{x}^T C\boldsymbol{x}$, hence the payoff of every strategy in the support of a Nash equilibrium $\boldsymbol{x}$ is constant, while all strategies outside the support $\sigma(\boldsymbol{x})$ earn a payoff $\leq \boldsymbol{x}^T C\boldsymbol{x}$. Considering the calculation of $(C\boldsymbol{x})_i$ in Equation (3.16), strategies in $\sigma(\boldsymbol{x})$ are compatible with each other. However, the external criterion is not strict : there could exist feature pairs not in $\sigma(\boldsymbol{x})$ that earn a payoff equal to $\boldsymbol{x}^T C\boldsymbol{x}$ in the match set, this may lead to a ambiguous match. Therefore, we use ESS as a good solution, as in Equation (3.18) a constraint has been added to guarantee that external criterion is strictly satisfied.

In order to cast the matching problem into a strategic game, we can take candidate pairs as pure strategies : $O = \{1, ..., n\}$, where $n = |\mathbb{A}|$. Hence, matched pairs belong to the support of an ESS which can be estimated iteratively by [3] :

$$\boldsymbol{x}_i(t+1) = x_i(t)\frac{(C\boldsymbol{x}(t))_i}{\boldsymbol{x}(t)^T C\boldsymbol{x}(t)} \tag{3.20}$$

where t is the iteration number.

This game theoretic solution has also been applied for feature grouping [109] and surface registration [2].

3.5 Online Three-stage Facial Point Localization

Inspired by the work of [2], where the game-theoretic technique is used for 3D image registration and where the global consistency between correspondences is well preserved, we propose here an online, three-stage method for facial point localization. While [2] matches the features of images for the *same* scene/object, we try to find the correspondences between feature points of two different face images with *different* identities and even of *different* expressions and illuminations. We cast the feature point localization problem into a coarse-to-fine matching task, an illustration of our algorithm is shown in Figure 3.3. In our model, the template ($\boldsymbol{T}$) is an image with manually labeled *target points* and for each test image ($\boldsymbol{I}$), we aim at finding the corresponding feature points. In the first step, instead of training specific detectors for each facial feature, as commonly used in other algorithms [29, 121], a generic method is applied to extract a set of interest points from $\boldsymbol{I}$. Then, for each target point in $\boldsymbol{T}$, a smaller set of these interest points are picked as candidates, using the robust feature descriptor POEM histogram [119]. Finally, we apply the game-theoretic technique to select desired facial points from candidates, without requirements of pre-defined constraints on feature distributions.

3.5.1 Step 1 : Detection of interest points

Unlike some approaches requiring trained detectors for specific facial features [29, 121], we first use a more generic method to extract a set of interest points which are invariant to scale, rotation and translation and which are also robust to illumination changes. A smaller set of these points will be picked as candidates

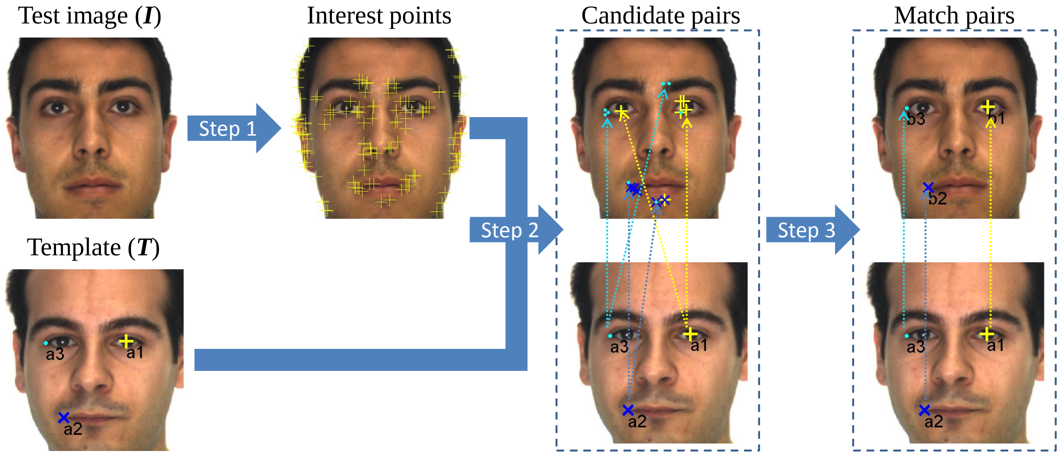


FIGURE 3.3 – Overview of our method. For clarity, only 3 facial target points are located as examples. In Step 1, interest points are found by a generic detector. For each point in the template, a small set of points are picked as candidates in Step 2. The desired facial points are selected from candidates in Step 3.

in the following step. The fundamental idea behind this is that we believe some facial features, e.g. eye corners, mouth corners and nostrils are invariant to similarity transformations with respect to the change of identity, expression and illumination. We have tested the following interest point detectors based on Laplacian-of-Gaussian (LoG) [77], Difference-of-Gaussian (DoG) [81], Hessian-Laplacian [88], and Harris-Laplacian [88]. Detailed information about these detectors has been introduced in Section 3.2. According to the visual results on several images (see an example in Figure 3.4), we adopt Harris-Laplacian detector here, since it can find more desired facial feature points, e.g., mouth corners, eyes corners, and nostrils.

3.5.2 Step 2 : Candidate points Screening

After the extraction of interest points, the localization of facial points turns into a matching problem between the target points from $\boldsymbol{T}$ and the interest points from $(\boldsymbol{I})$. Considering the efficiency of matching, for each target point, only K (e.g. $K \leq 10$) points in $\boldsymbol{I}$ with the nearest descriptor are picked as candidates.

Hence, a robust feature is required to distinguish the interest points. We propose here to use the POEM feature descriptor [119]. The main steps of calculating POEM histogram have been introduced in Section 3.3. Depending on the distances between histograms, K interest points with the nearest descriptor are picked as candidates

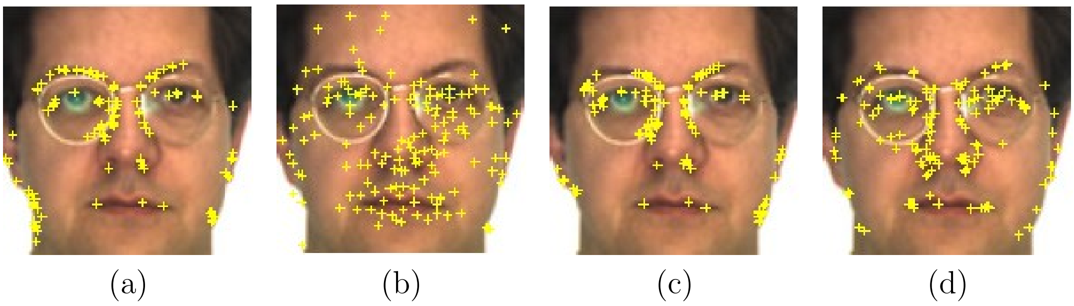


FIGURE 3.4 – Interest points detected by different methods : (a) LoG, (b) DoG, (c) Hessian-Laplacian and (d) Harris-Laplacian detector.

for each target point.

3.5.3 Step 3 : Multi-template game-theoretic matching

Up to this point, there are several candidate points in $\boldsymbol{I}$ for each target point in $\boldsymbol{T}$. Let $A_1 = \{\boldsymbol{a}_1, ..., \boldsymbol{a}_N\}$ and $A_2 = \{\boldsymbol{b}_1, ..., \boldsymbol{b}_L\}$ be the target and candidate point sets respectively, where $\boldsymbol{a}_i$,$\boldsymbol{b}_j$ represent the coordinates. Thus a target point $\boldsymbol{a}_i$ corresponds to K candidate point pairs : $(\boldsymbol{a}_i, \boldsymbol{b}_1)$,...,$(\boldsymbol{a}_i, \boldsymbol{b}_K)$. In this stage, we aim at finding the match pairs for every target point, e.g. $(\boldsymbol{a}_1, \boldsymbol{b}_1)$,$(\boldsymbol{a}_2, \boldsymbol{b}_2)$, and $(\boldsymbol{a}_3, \boldsymbol{b}_3)$ in Figure 3.3. As facial features have certain geometric structure, there exists a compatible transformation for all these match pairs. The selection process can be seen as a matching game [2], in which candidate pairs $(\boldsymbol{a}_i, \boldsymbol{b}_j)$ are defined as pure strategies available to players and the payoffs for every combination of strategies are calculated as :

$$\pi((\boldsymbol{a}_1, \boldsymbol{b}_1), (\boldsymbol{a}_2, \boldsymbol{b}_2)) = \frac{min(\|\boldsymbol{a}_1 - \boldsymbol{a}_2\|, \|\boldsymbol{b}_1 - \boldsymbol{b}_2\|)}{max(\|\boldsymbol{a}_1 - \boldsymbol{a}_2\|, \|\boldsymbol{b}_1 - \boldsymbol{b}_2\|)} \tag{3.21}$$

where $\|\cdot\|$ represents the Euclidian distance.

With Equation 3.21, strategies that correspond to rigid transformation have high payoff values, while less compatible pairs get lower scores. Take Figure 3.3 for example, $\pi((\boldsymbol{a}_1, \boldsymbol{b}_1)$,$(\boldsymbol{a}_2, \boldsymbol{b}_2))$ and $\pi((\boldsymbol{a}_1, \boldsymbol{b}_1)$,$(\boldsymbol{a}_3, \boldsymbol{b}_3))$ are higher than $\pi((\boldsymbol{a}_1, \boldsymbol{b}_2)$, $(\boldsymbol{a}_2, \boldsymbol{b}_3))$. Since players always want to get higher payoffs, they prefer to pick strategies that are compatible with their opponents' choices. As the game is repeated by a large population of players, a set of strategies with high mutual compatibility will be assigned to high weights. The compatible set of strategies can be obtained by calculating evolutionary stable states (ESS's), see Section 3.4 for details. Finally, the point pairs with high weights are taken as match pairs.

Since facial features in test images vary with the change of identity and expression, the matching problem will suffer from the error of candidate screening. More precisely, the correspondence $\boldsymbol{b}_i$ of one target point $\boldsymbol{a}_i$ may not be involved in the candidate set of $\boldsymbol{a}_i$. In that case, all pairs that contain $\boldsymbol{a}_i$ will get low weights after the matching game, i.e. this facial point is *miss-located*. To increase the robustness of game-theoretic matching, we apply multiple templates to match with test images. Only if one of these templates gives a match point of target point $\boldsymbol{a}_i$, this facial point can be successfully located. Hence, the probability of "miss-located" is very low. If a facial point is located by several templates, the average location is used as the final result.

3.6 Experimental results

3.6.1 Experiment settings

Database : AR face database [87].

Evaluation criterion. Let $\boldsymbol{b}_i$ and $\boldsymbol{b}_i^+$ be the predicted and manually labeled locations (ground truth), the localization error is calculated as : $m_i = \left\| \boldsymbol{b}_i - \boldsymbol{b}_i^+ \right\| / d_{eye}$, where d_{eye} is the average distance between two eye pupils in ground truth.

If we choose a threshold c, the correct localization rate will be :

$$rate = \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} \left(m_i^j < c \right)}{M \times N} \tag{3.22}$$

where M is the number of test images and N is the number of target points per template.

3.6.2 Matching of labeled points

In order to verify the effectiveness of our method for facial features, with the assumption of perfect selection of candidates, we first applied our method to match two sets of labeled points from two different images. We randomly selected 20 images of different individuals with neutral expression from AR-face database and ran game-theoretic matching between every two images, i.e. 190 image pairs. Original images with the resolution 768 × 576 are used directly in this experiment.

For each image pair, we take one image as template ($\boldsymbol{T}$) and calculate feature descriptors for all labeled points in both images. For each point in $\boldsymbol{T}$, 5 points with nearest descriptor in another image ($\boldsymbol{I}$) are used as candidates. AR-face images have been manually labeled with 22 landmarks, so there are 110 point pairs which are then regarded as strategies in the matching game.

A point in image $\boldsymbol{I}$ assigned to the corresponding point in $\boldsymbol{T}$, means a correct match. We adopted different POEM parameters to determine the closest neighbors, and the matching results can be seen in Figure 3.5. The average match rate is about 98% and the results are not sensitive to parameter selection. Hence, our method works well for the matching of facial points.

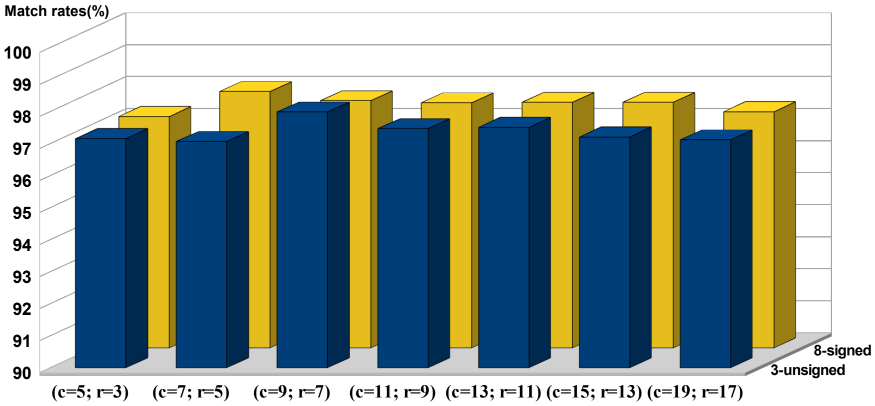


FIGURE 3.5 – Labeled points matching with POEM descriptor of different parameters. "3-unsigned" means that the POEM descriptor has 3 orientations discretized over $0-\pi$, and "8-signed" stands for a descriptor of 8 orientations over $0-2\pi$. c and r respectively represent the cell size and the radius of the POEM descriptor.

3.6.3 Facial features localization

Here, we aim at locating 10 facial points in test images (Figure 3.8). We form two image sets for evaluation : Data 1 consists of frontal faces with neutral, smile and anger expression and Data 2 is a set of face images under side illumination (see Section 2.5.1.1). All the face areas are extracted by Viola-Jones detector [117].

3.6.3.1 Using different number of templates

We first evaluated the impact of adopting different numbers of templates. The templates and 350 test images were randomly selected from Data 1. The localization results can be seen in Figure 3.6. It is clear that matching with single template gets lower accuracy than multiple templates, due to the high probability of "miss-located". While the results with 10, 15 and 20 templates are very similar, the accuracy of using 5 templates is slightly worse. For efficiency, we adopt 10 templates in the following experiments, which have no overlap with test images.

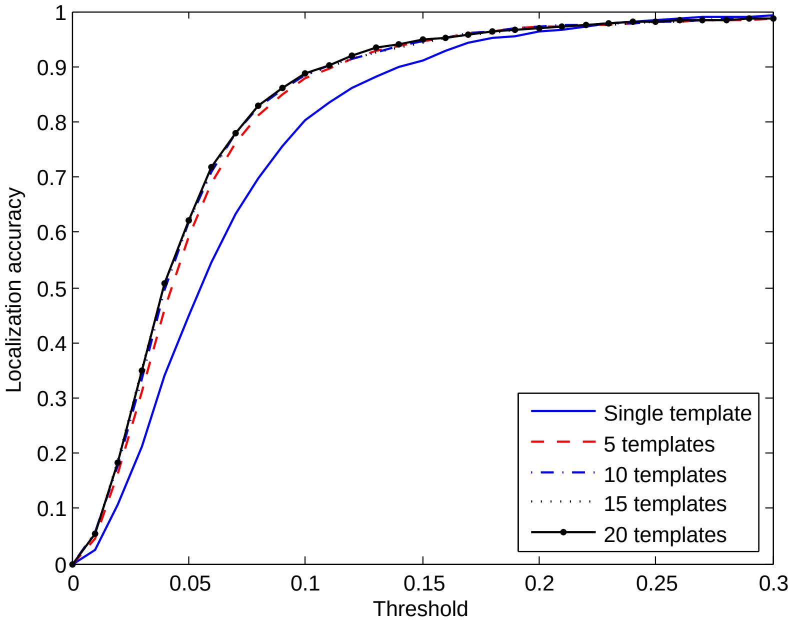


FIGURE 3.6 – Localization results using different numbers of templates.

3.6.3.2 Verification of the importance of game-theoretic matching

To show the importance of game-theoretic matching, we also tried to localize points without this step, i.e. we directly picked the points with closest descriptor in $\boldsymbol{I}$ as the correspondence of a target point in $\boldsymbol{T}$. Suffering from the variation of facial features, the closest-feature-based method is more like a random selection from detected interest points (Figure 3.7, Data 1), while game-matching-based method achieves a good performance. Hence the game-theoretic technique, which carries the information of face structure, is very important to facial point localization.

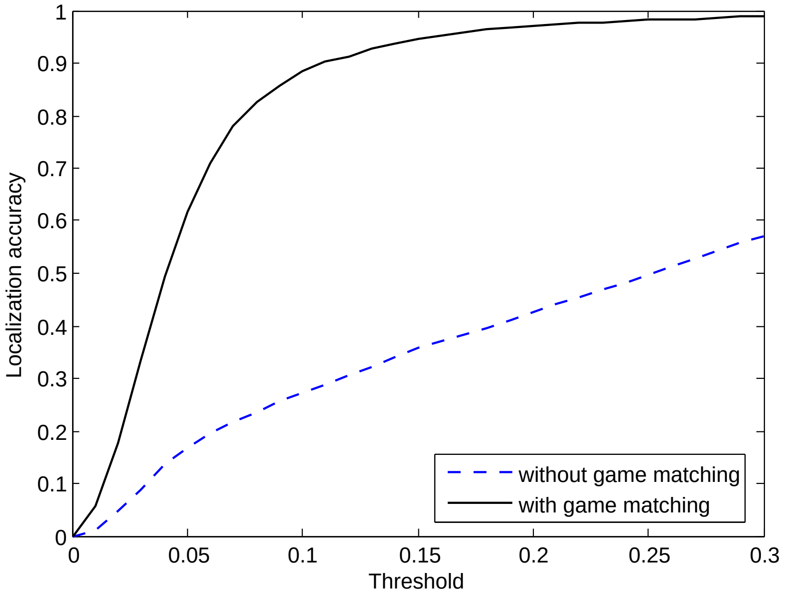


Figure 3.7 – Localization results with or without game matching step.

3.6.3.3 Using different feature descriptors under neutral condition

This section compares the performance of our method when different feature descriptors are used : intensity, SIFT [82] (the calculation of SIFT descriptor is introduced in Appendix A) and POEM descriptor. When using intensity values, the sum of squared differences (SSD) between two sub-regions is computed as the measure of distance. We calculated the three feature descriptors with the same window size, and the localization results can be seen in Figure 3.9 (Data 1). The facial point localization method works better with POEM than with SIFT, and SSD does not seem to be suitable in this case. Using a threshold $m < 0.15$, our approach is successful in 95% of points (see some examples in Figure 3.8), while localization accuracy with SIFT only reaches 82%. The rates of other methods, e.g. 96% for PRFR [29] and TST [28], 95% for [115], are very close to our result. Considering that our approach runs without specific trained detectors nor face models, the localization performance is satisfactory.

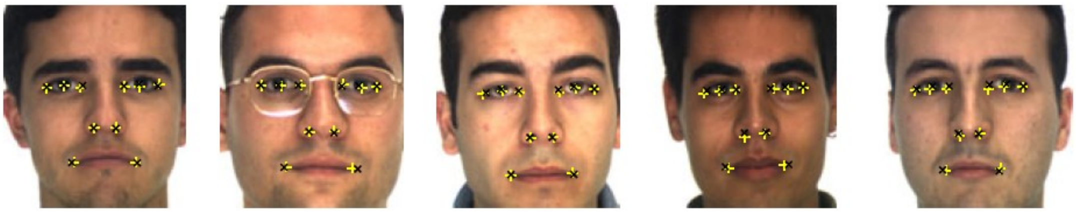


Figure 3.8 – Examples of localized facial points, where "+" is the output of our method and "×" is the manually labeled location.

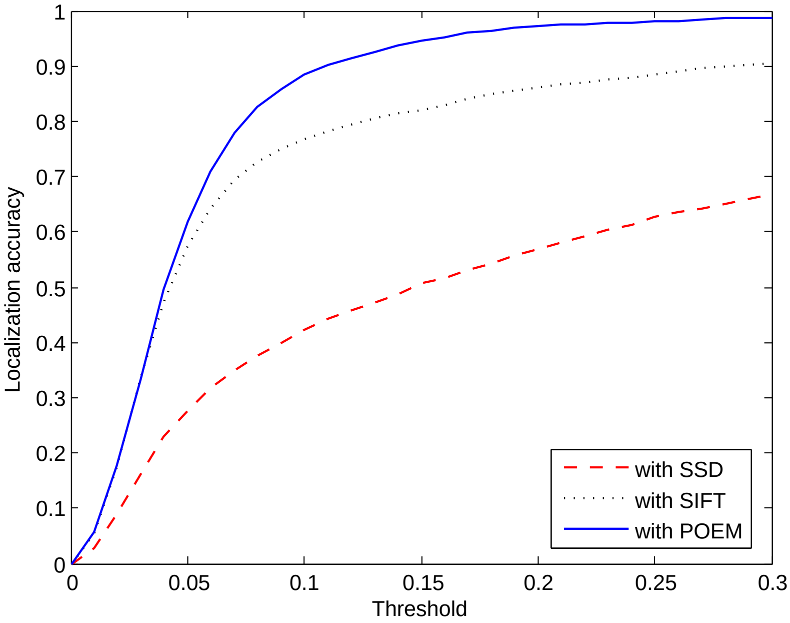


Figure 3.9 – Result of using Data 1 under neutral condition.

3.6.3.4 Using different feature descriptors under illumination changes

Few evaluations have been done specifically for locating facial points under lighting changes. Here, 100 images were selected randomly as test images from Data 2, and template set consists of 5 images from Data 1 and 5 images from Data 2. Three kinds of features are also compared in this case, and the results are shown in Figure 3.10. The game-theoretic method with POEM still gives better result than with other two features. For $m < 0.15$, our method reaches a success rate of 90% and the method with SIFT gets 81%. The accuracies are slightly lower than in neutral condition but still acceptable.

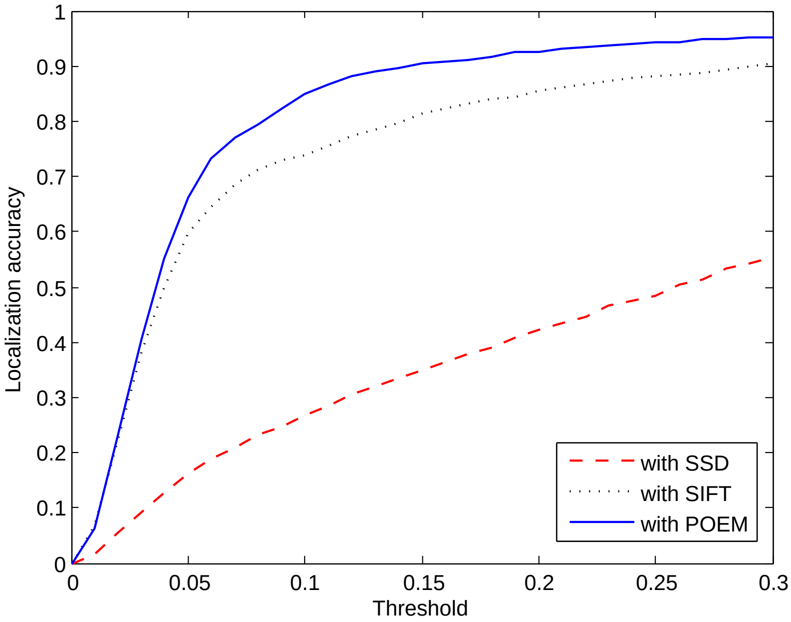


FIGURE 3.10 – Result of using Data 2 under lighting changes.

3.7 Conclusion

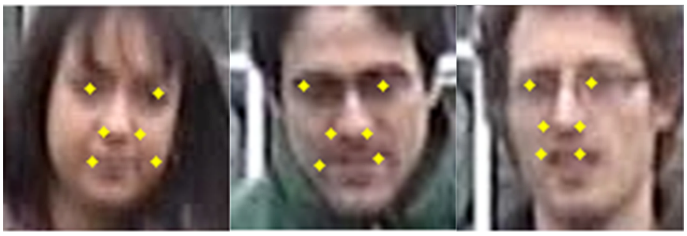


FIGURE 3.11 – Localization facial points on low quality video images.

This chapter presents an online approach to locating facial points, requiring no pre-defined constraints on feature distributions. We cast the localization problem in a matching game which preserves global geometric consistency of facial points. The experimental results demonstrate that the proposed algorithm achieves satisfactory performance on controlled AR images with expression and illumination changes. As discussed in Section 1.2, there are other challenging factors for face alignment in video surveillance context : occlusion, pose, noise, motion blur, and low resolution. Unfortunately, point-based methods seem not to perform well on low quality video surveillance images. Figure 3.11 show several examples of locating feature points on faces cropped from surveillance videos. Even the state-of-the-art facial point detectors such as BoRMaN [115] are not capable of working on these images. This is because point-based methods rely on local features which are sensitive to noise and blurs, i.e., local features may vary intensively with image quality. Also, occlusion and pose are still difficult factors for texture-based detectors.

To this end, the following work is focused on joint alignment methods which are more suitable for low quality images.

Chapitre 4

Lucas-Kanade Entropy Congealing for Joint Face Alignment

4.1 Introduction

This chapter presents our joint image alignment algorithm, referred to as Lucas-Kanade entropy congealing (LKC), in which images are simultaneously aligned by minimizing a sum-of-entropy function. The canonical entropy congealing was first proposed by Learned-Miller for joint alignment of binary images and magnetic resonance images [71, 89]. In congealing, the only assumption is the type of geometric mis-alignment. Later, congealing has been applied to more complex images, faces and cars [53], using SIFT descriptor [82] as the feature. This joint alignment method has certain advantages, it is unsupervised and not sensitive to image quality. However, the minimization of the entropy function is based on a sequential searching estimation with a pre-defined step size of updates (see details in Section 2.4.2). This usually results in slow convergence speed.

To deal with these limitations, we propose a Lucas-Kanade [10] based optimization algorithm to estimate all transformation parameters at the same time rather than in a sequential way as in Learned-Miller Congealing. More precisely, transformation updates are obtained by calculating the *Jacobian* and *Hessian* matrice of cost function. This chapter respectively presents two formulations of LKC, i.e., forward and inverse methods, which are different in the role of "template". Since the *distribution field* is invariant during one iteration, it is taken as the template in the forward formulation. However, there exists a huge computational cost in re-calculating the *Jacobian* and *Hessian* at each iteration. Baker and Matthews [10] pointed out that usually the key of efficiency is to switch the role of template and test data. Therefore, we exchange the roles of template and test image, this inverse compositional formulation yields constant parts of *Jacobian* and *Hessian* which can be precomputed to decrease computational complexity. Another contribution of our work is that POEM descriptor [119] is combined with congealing to catch more information about face.

Most optimization methods (including Lucas-Kanade algorithm) aim at finding the local minima of a cost function. Unfortunately, large mis-alignment errors in images (e.g., substantial translation and scaling, high-angle rotation) often lead to *undesirable* local minima. The optimization process might get stuck in wrong minima, i.e., images are not correctly aligned. Hence, a multi-resolution framework is applied to LKC to improve its robustness to large mis-alignment errors : in coarse levels, images are processed with low resolutions to remove major mis-alignment errors ; in fine levels, alignment is refined using higher resolutions. Moreover, the use of multi-resolution strategy brings an improvement of computational speed.

The remainder of this chapter is arranged as follows : Sections 4.2.1 and 4.2.2 respectively introduce forward and inverse formulations of LKC, and their comparison results are given in Section 4.2.4. In Section 4.3.1, we present the details of multi-resolution framework for joint alignment, and its experimental results are discussed in Section 4.3.2. Section 4.4 is the conclusion of this chapter.

4.2 Lucas-Kanade entropy congealing

As introduced in Section 2.4.2, entropy congealing [53] aligns an image by minimizing a sum-of-entropy function :

$$arg \min_{\mathbf{\Phi}} - \sum_{\boldsymbol{x}} \sum_{j \in \boldsymbol{J}} p_j(\boldsymbol{x}) log_2(p_j(\boldsymbol{x})) \tag{4.1}$$

where $\mathbf{\Phi} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N\}$ is the set of warp parameter vectors for different images, N is the number of images, $\boldsymbol{x} = (x, y)^T$ is a column vector containing the pixel coordinates, $\boldsymbol{J}$ is the set of image features, and $p_j(\boldsymbol{x})$ is the probability of feature j in the pixel set at $\boldsymbol{x}$.

In the application to gray level images, [53] pointed out that the high variations of intensity in the foreground object as well as the variations due to lighting will cause high entropy even under a proper alignment. Therefore, instead of intensity values, some robust feature descriptors can be used with Congealing. In [53], SIFT [82] is used as the feature descriptor. Here, we combine Congealing with the robust feature descriptor POEM [119] which catches both the edge information of local patches and the relation between this information in a neighboring region (the main steps of calculating POEM descriptor have been introduced in Section 3.3). Be different from [119], here each pixel is represented by a vector of POEM values in Equation (3.14). As pointed out in [119], POEM presents the following properties : POEM (1) is an oriented, spatial multi-resolution descriptor capturing rich information about the original image, (2) is a multi-scale self-similarity based structure that results in robustness to exterior variations, (3) is of low complexity. Compared to SIFT used in [53] which only extracts local shape information, POEM descriptor considers both local shape information and relationship between this information of different local structures. Indeed, in [119, 120], POEM catches better details about face than SIFT, and its performance for face recognition is considerably higher than this of SIFT on both FERET [95] and LFW [54] datasets.

However, it is difficult to directly calculate probabilities $p_j(\boldsymbol{x})$ of POEM descriptor. Hence, similar to [53], all feature descriptors are modeled as being generated from a mixture of Gaussians model by K-mean clustering. In this way, pixel at position $\boldsymbol{x}$ of the ith image can be represented by a vector of probabilities :

$$(p_1^i(\boldsymbol{x}), p_2^i(\boldsymbol{x}), ..., p_M^i(\boldsymbol{x})) \tag{4.2}$$

where M is the number of clusters, $p_k^i(\boldsymbol{x})$ is the probability that this pixel belongs to cluster k.

Then the sum-of-entropy function in Equation (4.1) can be rewritten as :

$$-\sum_{\boldsymbol{x}} \sum_{k}^{M} D_k(\boldsymbol{x}) log_2(D_k(\boldsymbol{x})) \tag{4.3}$$

where $D_k(\boldsymbol{x})$ is called "distribution field" calculated by :

$$D_k(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} p_k^i(\boldsymbol{x}) \tag{4.4}$$

Obviously, the alignment of ith image can be achieved by maximizing :

$$\sum_{\boldsymbol{x}} \sum_{k}^{M} p_k^i(\boldsymbol{x}) log(D_k(\boldsymbol{x})) \tag{4.5}$$

4.2.1 Forward compositional LKC

Here, we first present a forward Lucas-Kanade method for facial image alignment. Since the distribution field $D_k(\boldsymbol{x})$ is updated only when all the images are transformed after an iteration, $D_k(\boldsymbol{x})$ can be taken as invariable during the estimation of transformation parameters for one image. Therefore, we can take $D_k(\boldsymbol{x})$ as the template and $p_k(\boldsymbol{x})$ as test data, and Equation (4.5) is rewritten as :

$$\sum_{\boldsymbol{x}} \sum_{k}^{M} p_k^i(\boldsymbol{w}(\boldsymbol{x}, \boldsymbol{v}^i)) log(D_k(\boldsymbol{x})) \tag{4.6}$$

where $\boldsymbol{w}(\boldsymbol{x}, \boldsymbol{v}^i)$ denotes the warp for the ith image. In order to optimize Equation (4.6), assuming that the current estimation of $\boldsymbol{v}^i$ is known, we iteratively update the parameters with an increment $\Delta \boldsymbol{v}^i$, then Equation (4.6) becomes :

$$\sum_{\boldsymbol{x}} \sum_{k}^{M} p_k^i(\boldsymbol{w}(\boldsymbol{x}, \boldsymbol{v}^i \circ \Delta \boldsymbol{v}^i)) log(D_k(\boldsymbol{x})) \tag{4.7}$$

Hereafter, to save space, the transformation parameters and the sequence number of image are not presented. The *Jacobian* and *Hessian* of Equation (4.6) can be obtained using chain and product rules :

$$\begin{aligned} \boldsymbol{G} &= \sum_{\boldsymbol{x}} \sum_{k}^{M} \frac{\partial p_k}{\partial \boldsymbol{v}} log(D_k) \\ &= \sum_{\boldsymbol{x}} \sum_{k}^{M} \nabla p_k \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}} log(D_k) \end{aligned} \tag{4.8}$$

$$\begin{aligned} \boldsymbol{H} &= \sum_{\boldsymbol{x}} \sum_{k}^{M} [(\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}})^T \nabla \cdot \nabla p_k \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}} log(D_k) \\ &\quad + \nabla p_k \frac{\partial^2 \boldsymbol{w}}{\partial \boldsymbol{v}^2} log(D_k)] \\ &= \sum_{\boldsymbol{x}} \sum_{k}^{M} (\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}})^T \nabla \cdot \nabla p_k \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}} log(D_k) \end{aligned} \tag{4.9}$$

where ∇p_k is the gradient of probability, $\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}}$ is the *Jacobian* of the warp.

Because it is a maximization problem in this situation, $\Delta \boldsymbol{v}$ is calculated by :

$$\Delta \boldsymbol{v} = \boldsymbol{H}^{-1} \boldsymbol{G} \tag{4.10}$$

Then $\boldsymbol{v}$ is updated using :

$$\boldsymbol{v} = \boldsymbol{v} \circ \Delta\boldsymbol{v} \tag{4.11}$$

As summary, the main steps of forward compositional LK Congealing are shown in Figure 4.1.

```
repeat
  (Pre-computed :)
  (1) Calculate new distribution field $D(\boldsymbol{x})$ using Equation (4.4)
  for $i = 1$ to $N$ do
    (2) Warp $p^i(\boldsymbol{x})$ with $\boldsymbol{w(x, v^i)}$ to compute $p^i(\boldsymbol{w(x, v^i)})$
    (3) Calculate the gradient $\nabla p^i(\boldsymbol{w})$ of $p^i(\boldsymbol{w(x, v^i)}$
    (4) Calculate the gradient $\nabla \cdot \nabla p^i(\boldsymbol{w})$ of $\nabla p^i(\boldsymbol{w})$
    (5) Calculate $\nabla p^i(\boldsymbol{w})\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}}$
    (6) Calculate $(\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}})^T \nabla \cdot \nabla p^i(\boldsymbol{w})\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}}$
    (7) Calculate $\boldsymbol{G}$ using Equation (4.8)
    (8) Calculate $\boldsymbol{H}$ using Equation (4.9)
    (9) Calculate $\Delta\boldsymbol{v}^i$ using Equation (4.10)
    (10) Update $\boldsymbol{v}^i$ using Equation (4.11)
  end for
until Equation (4.3) is converged.
```

FIGURE 4.1 – Main steps of forward compositional LK Congealing.

4.2.2 Inverse compositional LKC

There is a huge computational cost in re-calculating the *Jacobian* and *Hessian* at each iteration. Usually the key to efficiency is switching the role of template and test data [10]. Following this idea, after a change of variables, our goal becomes to find a transformation for the distribution field D_k to maximize :

$$\sum_{\boldsymbol{x}} \sum_{k}^{M} p_k(\boldsymbol{w}(\boldsymbol{x}, \boldsymbol{v}_p)) log(D_k(\boldsymbol{w}(\boldsymbol{x}, \boldsymbol{v}_d))) \tag{4.12}$$

where $\boldsymbol{v}_p$, $\boldsymbol{v}_d$ are the transformation parameters for $p_k(\boldsymbol{x})$ and $D_k(\boldsymbol{x})$.

Similarly, the optimization of Equation (4.12) can be solved by iteratively calculating an update $\Delta\boldsymbol{v}$ in the following equation :

$$\sum_{\boldsymbol{x}}\sum_{k}^{M} p_k(\boldsymbol{w}(\boldsymbol{x},\boldsymbol{v}_p))log(D_k(\boldsymbol{w}(\boldsymbol{x},\boldsymbol{v}_d\circ\Delta\boldsymbol{v}))) \tag{4.13}$$

The *Jacobian* and *Hessian* of Equation (4.12) can also be obtained using chain and product rules :

$$\begin{aligned}\boldsymbol{G} &= \sum_{\boldsymbol{x}}\sum_{k}^{M} p_k \frac{\partial log(D_k)}{\partial \boldsymbol{v}_d} \\ &= \sum_{\boldsymbol{x}}\sum_{k}^{M} \frac{p_k}{D_k}\nabla D_k \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}_d}\end{aligned} \tag{4.14}$$

$$\begin{aligned}\boldsymbol{H} = \sum_{\boldsymbol{x}}\sum_{k}^{M} p_k[-\frac{1}{D_k^2}(\nabla D_k \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}_d})^T \nabla D_k \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}_d} \\ + \frac{1}{D_k}(\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}_d})^T \nabla\cdot\nabla D_k \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}_d}]\end{aligned} \tag{4.15}$$

Then, the transformation parameters $\Delta\boldsymbol{v}$ for $D_k(\boldsymbol{x})$ is calculated by :

$$\Delta\boldsymbol{v} = \boldsymbol{H}^{-1}\boldsymbol{G} \tag{4.16}$$

$\Delta\boldsymbol{v_d}$ will be updated to $\boldsymbol{v}_p$ after an inverse transformation, see Equation (4.17), i.e., D_k will not be changed in this step. Also, the main steps of inverse compositional LK Congealing are shown in Figure 4.2.

$$\boldsymbol{v}_p = \boldsymbol{v}_p \circ \Delta\boldsymbol{v}^{-1} \tag{4.17}$$

In inverse compositional LKC, since the distribution field D_k is invariable during an iteration, the following parts of *Jacobian* and *Hessian* in Equations (4.14) and (4.15) are also invariable :

$$\frac{1}{D_k}\nabla D_k \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}_d} \tag{4.18}$$

$$-\frac{1}{D_k^2}(\nabla D_k \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}_d})^T \nabla D_k \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}_d} + \frac{1}{D_k}(\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}_d})^T \nabla\cdot\nabla D_k \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}_d} \tag{4.19}$$

```
repeat
  (Pre-computed :)
  (1) Calculate new distribution field $D(\boldsymbol{x})$ using Equation (4.4)
  (3) Calculate the gradient $\nabla D(\boldsymbol{x})$ of $D(\boldsymbol{x})$
  (4) Calculate the gradient $\nabla \cdot \nabla D(\boldsymbol{x})$ of $\nabla D(\boldsymbol{x})$
  (5) Calculate $\frac{1}{D(\boldsymbol{x})} \nabla D(\boldsymbol{x}) \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}_d}$
  (6) Calculate $-\frac{1}{D(\boldsymbol{x})^2} (\nabla D(\boldsymbol{x}) \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}_d})^T \nabla D(\boldsymbol{x}) \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}_d} + \frac{1}{D(\boldsymbol{x})} (\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}_d})^T \nabla \cdot \nabla D(\boldsymbol{x}) \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}_d}$
  for $i = 1$ to $N$ do
    (2) Warp $p^i(\boldsymbol{x})$ with $\boldsymbol{w(x, v^i)}$ to compute $p^i(\boldsymbol{w(x, v^i)})$
    (7) Calculate $\boldsymbol{G}$ using Equation (4.14)
    (8) Calculate $\boldsymbol{H}$ using Equation (4.15)
    (9) Calculate $\Delta \boldsymbol{v}^i$ using Equation (4.16)
    (10) Update $\boldsymbol{v}^i$ using Equation (4.17)
  end for
until Equation (4.3) is converged.
```

FIGURE 4.2 – Main steps of inverse compositional LKC.

Therefore, Equations (4.18) and (4.19) can be pre-computed at the beginning of each iteration (see Steps (3)-(6) in the outer loop of Figure 4.2). The computational costs of *Jacobian* and *Hessian* for the forward and inverse LKC based methods at each iteration are shown in Table (4.1). In the inverse formulation, the complexity of computing *Hessian* decreases from $O(MN_xN_v^2)$ to $O(MN_xN_v)$. Given N images, the costs of forward and inverse methods are respectively $O(MNN_xN_v^2)$ and $O(MNN_xN_v + MN_xN_v^2)$ at each iteration. Since N is much bigger than N_v ($N >> N_v$, e.g., $N_v = 4$, $N > 100$), $O(MNN_xN_v^2) >> O(MNN_xN_v) >> O(MN_xN_v^2)$. As a result, $O(MNN_xN_v^2) > O(MNN_xN_v + MN_xN_v^2)$. In other words, if running on a large number of images, the computational cost of inverse method will be much lower than that of the forward method.

4.2.3 Parameter drift

Learned-Miller [71] pointed out that there exists a problem called parameter "drift" in joint alignment application. More precisely, some estimated transformation parameters increase the value of cost function, but they can not align the images as desired. For example, iteratively shrinking all images will cause all the pixels containing useful information to disappear, and at the meantime the joint gradient

		Forward method	Inverse method
Pre-computation	G	-	$O(MN_xN_v)$
	H	-	$O(MN_xN_v^2)$
Per image	G	$O(MN_xN_v)$	$O(MN_xN_v)$
	H	$O(MN_xN_v^2)$	$O(MN_xN_v)$
N images	G	$O(MNN_xN_v)$	$O(MNN_xN_v)$
	H	$O(MNN_xN_v^2)$	$O(MNN_xN_v)$

Table 4.1 – Computational costs of *Jacobian* and *Hessian* for forward and inverse LKC based algorithms at each iteration. N_x and N_v are respectively the numbers of pixels and of warp components.

correlation coefficient will reach the maximum value. The example in Figure 4.3 shows that there is a clear shrinkage in images suffering of parameter drift.

In order to avoid parameter drift, we adopt a similar process to [26]. First we select several corner points (e.g., left-top, right-bottom) of the images and record their initial positions. After one iteration, we calculate the average positions of selected points, then find a warp $\boldsymbol{p}_d$ to transform the average of points to their initial positions. Based on the type of mis-alignment, different numbers of points are selected for this estimation (two points for similarity transformation and three for affine transformation). Finally, $\boldsymbol{p}_d$ is composed with the transformation parameters of all images at the end of this iteration. In this way, the average of transformations is always close to identity transformation, and the parameter drift case can be avoided.

4.2.4 Experimental results

4.2.4.1 Experiment data

We conduct comparison experiments on gray images from AR [87], SCface [42], FERET [95], LFW (Labeled Face in the Wild) [54] databases, and a set of video surveillance images. Whilst the first two databases, AR and SCface, are used to verify the strength of the proposed algorithm regarding two evaluation criteria : average image and convergence rate (detailed information about evaluation methods

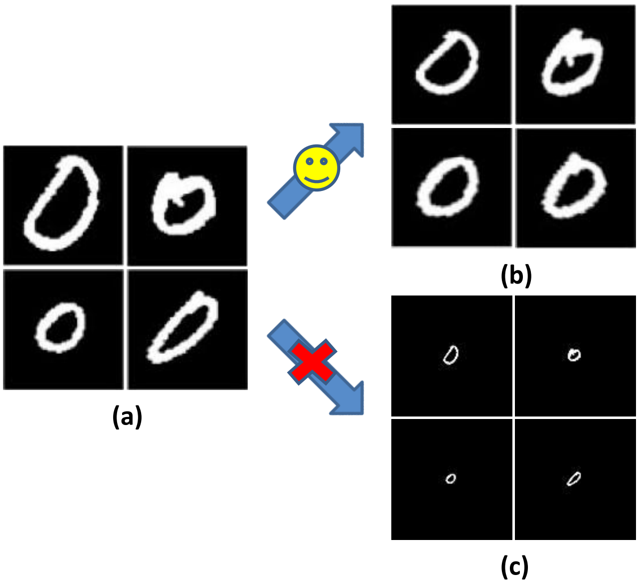


FIGURE 4.3 – Parameter drift phenomenon in joint alignment process. (a) Original images, (b) Desired aligned images, (c) Images suffering from parameter drift.

is introduced in 2.6), FERET and LFW are used to show the advantage of our algorithm for different face recognition methods. The video surveillance images are tested using average image as evaluation method.
(1) In order to evaluate the alignment performance *under illumination variations*, we first form two image sets from AR database : *Data 1* is a set of 120 images under neutral conditions with uniform illumination and *Data 2* contains 60 images under neutral conditions and 60 images under side illumination (see examples in Section 2.5.1.1). All the images are randomly selected and related to different identities. Thanks to the previous work [1], we can collect 22 landmarks per face of this database (see Figure 2.13).
(2) To prove the strength of our algorithm *in real-world conditions*, we also evaluate its performance on the SCface database. 130 images with *low resolution and noise* are selected from this database (see examples in Figure 4.4). For each image, there are 4 landmarks (two eye centers, nose tip, and mouth center) for the evaluation on this database.
(3) FERET is a classic database for face recognition experiments. 1,000 frontal face

[1] `http://personalpages.manchester.ac.uk/staff/timothy.f.cootes/data/tarfd_markup/tarfd_markup.html`

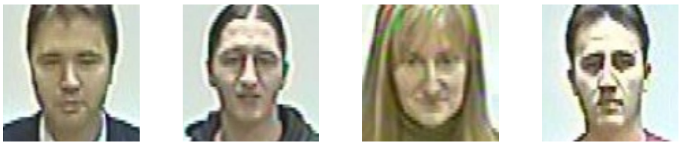


Figure 4.4 – Examples of test images from SCface database.

images of 500 persons are randomly selected, each person with 2 images : 1 “fa” image and 1 “fb” image (more information can be found in Section 2.5.1.2).

(4) Facial images from LFW database are also used for face recognition evaluation. In our experiments, we randomly selected 500 pairs of images from view 2 subset 1 for training, and 500 pairs from view 2 subset 2 for testing (250 matched and 250 mismatched, respectively). Details of LFW database are introduced in Section 2.5.2.2.

(5) We also extract a set of facial images from the surveillance videos collected by the laboratory LASMEA in France. Since general face detectors do not perform well on these video surveillance images, we roughly crop the face regions for evaluation. This test set contains 90 images (6 subjects) which undergo variations in poses and expression, low resolution, noise, and motion blur (several examples are shown in Figure 4.5).

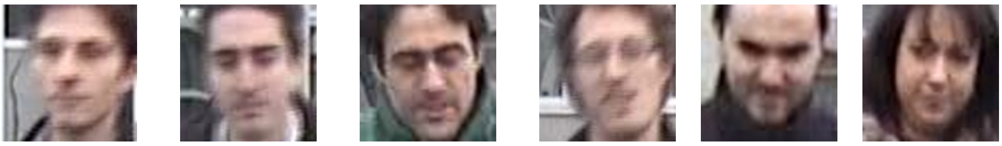


Figure 4.5 – Examples of faces extracted from video surveillance images.

4.2.4.2 Parameter settings

Parameters include the number of clusters, parameters of POEM and SIFT descriptors (the calculation of SIFT descriptor is introduced in Appendix A). The number of clusters in Gaussian mixture models is an important parameter for entropy Congealing. We evaluate the effect of cluster number on AR-face *Data 1*. Convergence rates with different numbers of clusters calculated using Equation (2.36) are shown in Figure 4.6. If the cluster number ≥ 10, convergence rates are very similar. That is, alignment performance is robust to cluster number in a certain range. In the following test, we adopt 10 clusters for Congealing methods.

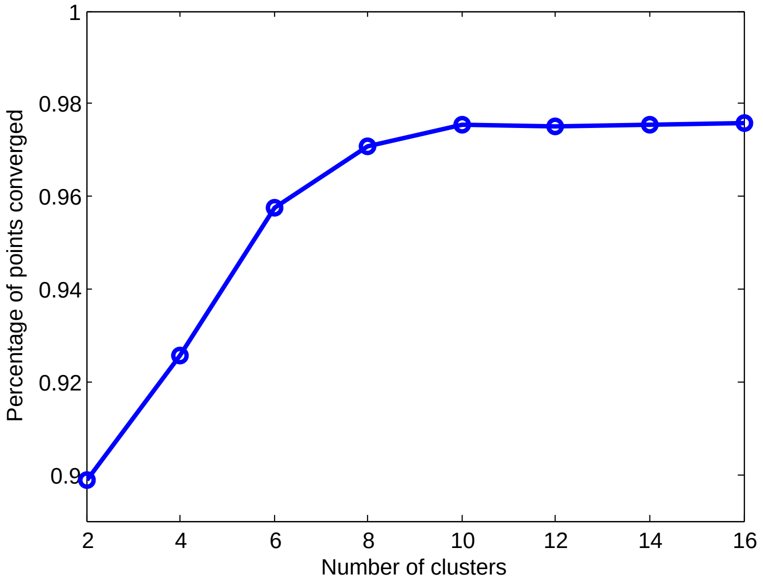


FIGURE 4.6 – Convergence rates with different numbers of clusters.

Concerning the parameters of POEM, we use here the parameters suggested in [119], that is, we adopt POEM feature with 3 unsigned orientations, a cell size of 7×7 and a radius of 5. Actually, according to our tests, the performance of proposed method is not sensitive to the selection of POEM parameters in a certain range. The code of original Learned-Miller Congealing is available on Internet [2], it uses 8 orientations for SIFT descriptor. For a fair comparison, we report here the results of LMC obtained with 4 orientation SIFT descriptors, rather than 8 orientations as in [53], because this results in better performance.

Here, we assume that the type of geometric mis-alignment is a similarity transformation, i.e., there are four parameters : x-translation, y-translation, in-plane rotation and scaling. Other transformations, e.g., affine transformation, are also adaptable for the proposed algorithm, but they may cause unwanted deformation of face.

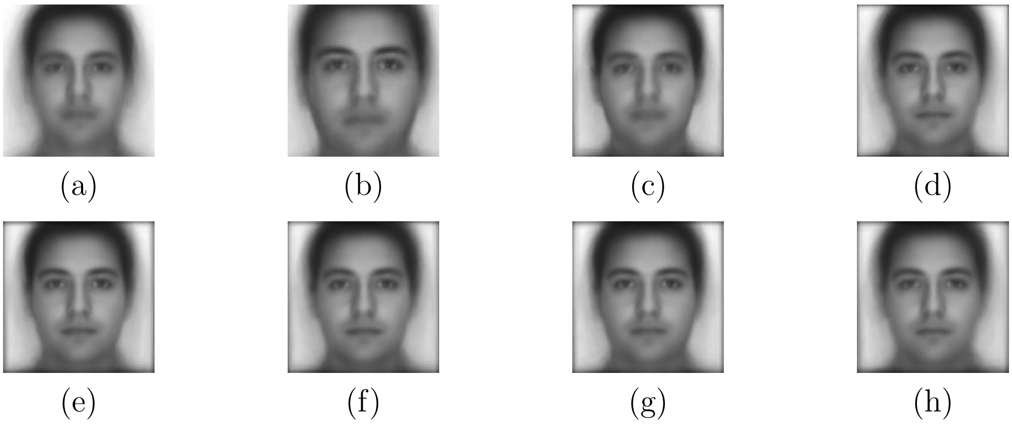
4.2.4.3 Average image

To compare the performance of alignment methods and feature descriptors, we align images using Learned-Miller Congealing (LMC) [53], forward compositional (fc-LKC) and inverse compositional Lucas-Kanade entropy Congealing (ic-LKC) com-

[2] `http://vis-www.cs.umass.edu/code/congealingcomplex/`

bined with POEM and SIFT descriptors respectively. Also, the results of RASL [93] are included for comparison [3].

The average images of AR-face *Data 1* after different alignments are shown in Figure 4.7. Intuitively, both face shape and facial features are unclear in the average of original images. The average of images aligned by LMC with SIFT has clearer face shape contours than original images, but is still blurry inside the face. We can see that the remaining five congealing based methods show similar performance in this visual evaluation. They all perform better than LMC with SIFT, since their average images have more distinct eye, nose and mouth contours. Similar conclusions are obtained from results on AR-face *Data 2* (see Figure 4.8) when compared with those on *Data 1*. This means that our methods are robust to illumination variations. RASL shows certain improvement on average images, e.g., eyebrows and eyes in Figure 4.7 (b) and lips in Figure 4.8 (b) are clearer than those of unaligned images. However, compared with the results of LKC methods, the average images generated by RASL are obviously blurrier.


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

(e) (f) (g) (h)

FIGURE 4.7 – Average images of AR-face *Data 1* aligned by (a) original (unaligned), (b) RASL (c) LMC with SIFT, (d) LMC with POEM, (e) fc-LKC with SIFT, (f) fc-LKC with POEM, (g) ic-LKC with SIFT, (h) ic-LKC with POEM.

We turn now to the alignment results obtained from the SCface database with more difficult images of low quality, e.g., low resolution and strong noise. The related average images are shown in Figure 4.9. As we can see from this figure, the face shape and features are blurry in the average of original images. LMC with SIFT

[3] The Matlab code of RASL is available at `http://perception.csl.uiuc.edu/matrix-rank/rasl.html`

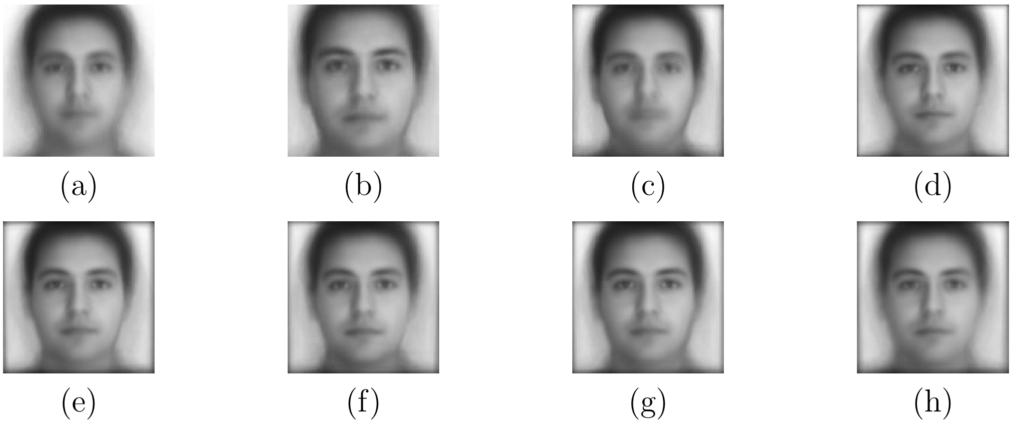


FIGURE 4.8 – Average images of AR-face *Data 2* aligned by (a) original (unaligned), (b) RASL (c) LMC with SIFT, (d) LMC with POEM, (e) fc-LKC with SIFT, (f) fc-LKC with POEM, (g) ic-LKC with SIFT, (h) ic-LKC with POEM.

just makes the face shape contour clearer, but the facial features are still unclear. Besides, RASL does not show obvious improvement on this image set. Intuitively, the remaining methods have no significant differences in performance and they yield more distinct average images with clearer facial feature contours, e.g., nostrils and mouth. Based on these results, we can conclude that the proposed methods work well on low quality images.

Figure 4.10 shows the average images of the challenging video surveillance data. The average of original images is very blurry. It is interesting to see that all considered joint alignment methods produce clearer average images (e.g., the regions of mouth and nose). Due to the motion blurs and noises in original images, it is difficult to make further comparison. Even though, these results prove that our LKC based methods are capable of aligning low quality video surveillance images.

4.2.4.4 Convergence rate

(1) With different thresholds

We further evaluate the alignment performance by calculating the convergence rates with different thresholds, using Equation (2.36). Here, the same approaches as those mentioned in Section 4.2.4.3 are compared. As can be seen from the results on the AR-face *Data 1* (Figure 4.11), all the alignment approaches significantly improve the convergence rates over original images, and the four LKC methods have higher convergence rates than the two LMC methods. The overall convergence rates of RASL

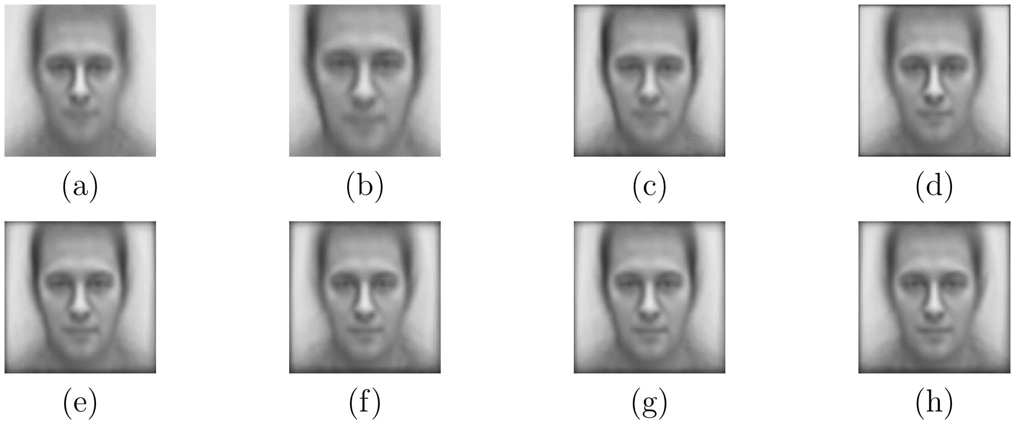


FIGURE 4.9 – Average images of SCface aligned by (a) original (unaligned), (b) RASL (c) LMC with SIFT, (d) LMC with POEM, (e) fc-LKC with SIFT, (f) fc-LKC with POEM, (g) ic-LKC with SIFT, (h) ic-LKC with POEM.

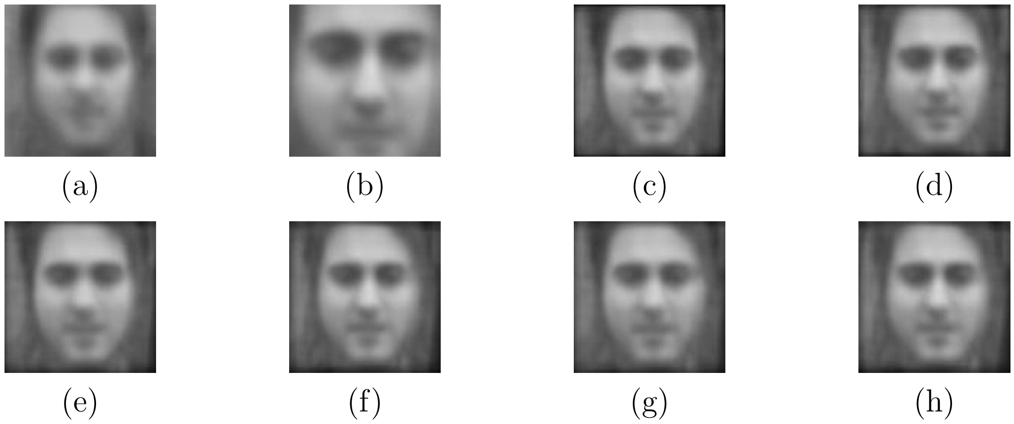


FIGURE 4.10 – Averages of video surveillance images aligned by (a) original (unaligned), (b) RASL (c) LMC with SIFT, (d) LMC with POEM, (e) fc-LKC with SIFT, (f) fc-LKC with POEM, (g) ic-LKC with SIFT, (h) ic-LKC with POEM.

are close to those of LMC with SIFT. When threshold ≤ 2, the LKC methods show similar performance ; when threshold ≥ 3, ic-LKC with POEM works better than other three LKC methods. It is also interesting to find that POEM improves alignment performance, compared to SIFT descriptor. More precisely, convergence rates of LMC with POEM are higher than those of LMC with SIFT ; convergence rates of fc-LKC with POEM are higher than those of fc-LKC with SIFT and convergence rates of ic-LKC with POEM are higher than those of ic-LKC with SIFT.

Results on AR-face *Data 2* can be seen in Figure 4.12. Ic-LKC with POEM always produces better alignment accuracy than other alignment methods. And there is a distinct decrease in the convergence rates of LMC with SIFT, which are close to original rates when threshold ≤ 2. Besides, RASL performs better than LMC with SIFT, but worse than the rest methods. Other conclusions are similar to those obtained from *Data 1*, except that the overall convergence rates are slightly lower (this is easy to understand since *Data 2* is more difficult than *Data 1*), but the performance of our methods is still satisfactory.

Concerning convergence rates of different alignment methods on SCface images, as can be seen from Figure 4.13, all methods still improve the convergence rates over original images, except LMC with SIFT which has lower rates when threshold ≥ 4. The reason is that the landmarks of SCface are all inside feature points, while LMC with SIFT just improves the consistency on face shape. Besides, the differences between results of other congealing based methods on this dataset are less than those on the AR-face images. This might be because the number of landmarks are smaller, only 4 landmarks are provided for calculation. Since the convergence rates generated by RASL are even lower than those of unaligned images, this method does not work at all on the low quality image set. Among all the methods, ic-LKC always performs the best.

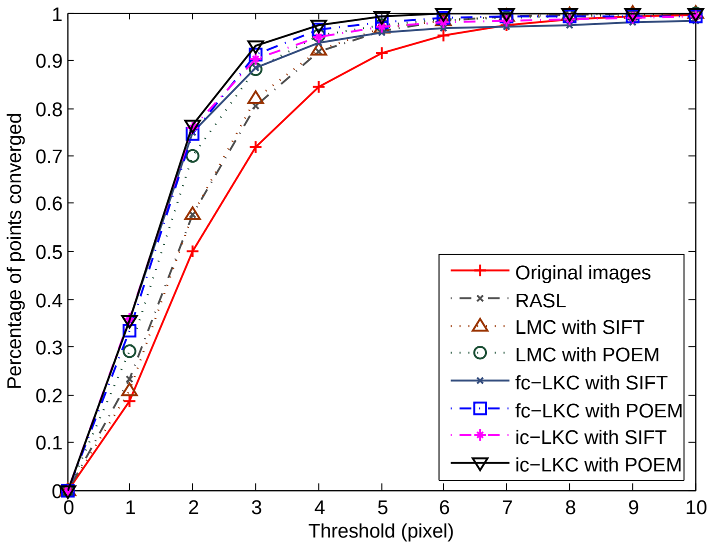


FIGURE 4.11 – Convergence rates with different thresholds on AR-face *Data 1*.

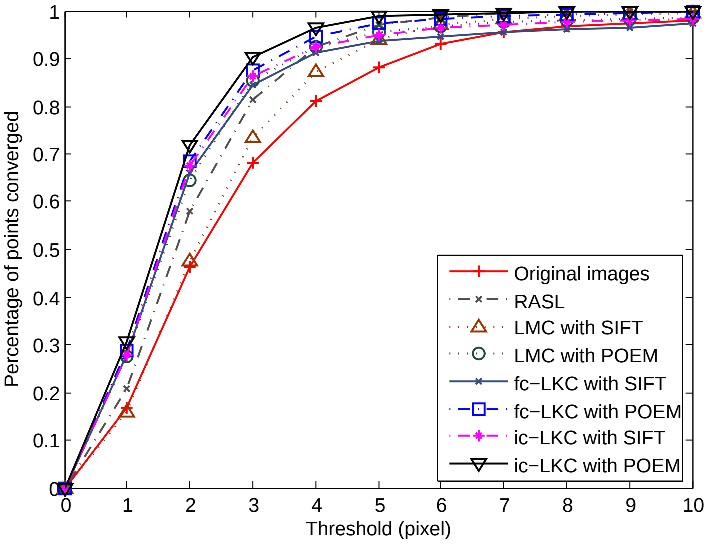


FIGURE 4.12 – Convergence rates with different thresholds on AR-face*Data 2*.

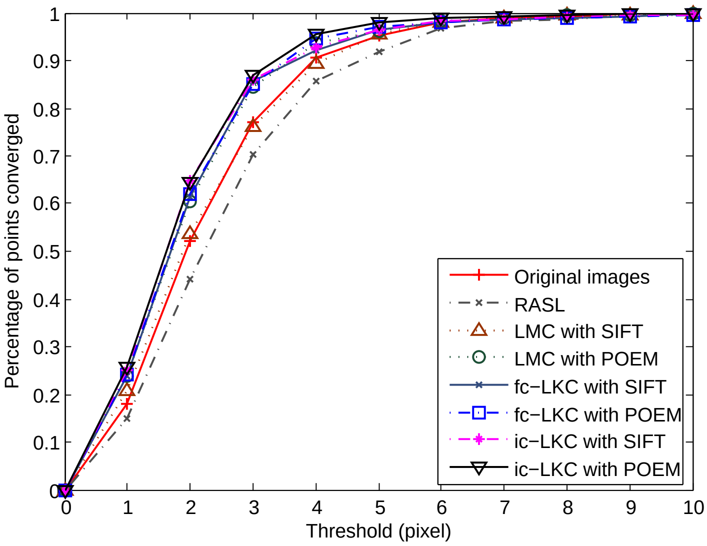


FIGURE 4.13 – Convergence rates on the SCface database with different thresholds.

(2) In different iterations

We also calculate the convergence rates of landmarks at different iterations to analyze the convergence ability of alignment methods. Landmarks are tracked throughout the Congealing process, and we calculate the convergence rates using Equation (2.36) at each iteration. The convergence curves of AR *Data 1*, *Data 2* and SCface are respectively shown in Figures 4.14, 4.15 and 4.16. It is clear that LKC methods require less iterations to reach a termination criterion than LMC algorithms and significantly increase the convergence rate especially during the first iterations of the process. This is because the updates of LKC methods have no certain limits, while there is a pre-defined step size for LMC. RASL shows a similar convergence speed to LMC methods, but this approach is not stable for aligning low quality SC images. It is also interesting to see that POEM based methods show more stable trends to convergence on images with lighting changes (AR *Data 2*, Figure 4.15) and images with low quality (SCface, Figure 4.16) when compared with SIFT based approaches. That is, POEM descriptor is more robust to variations in illumination and image quality.

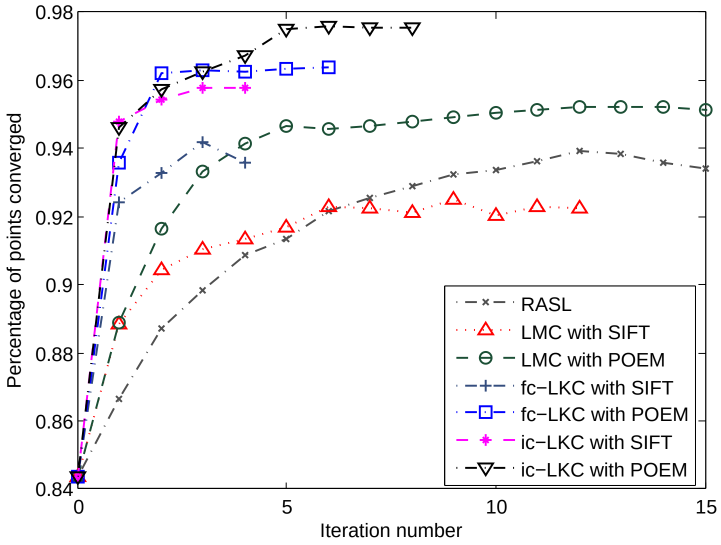


FIGURE 4.14 – Convergence curves of tracking landmarks on AR-face *Data 1*.

4.2.4.5 Improvement in face recognition

In this section, we evaluate the performance of alignment method regarding their improvement in face recognition accuracy. In order to reduce the impact of face

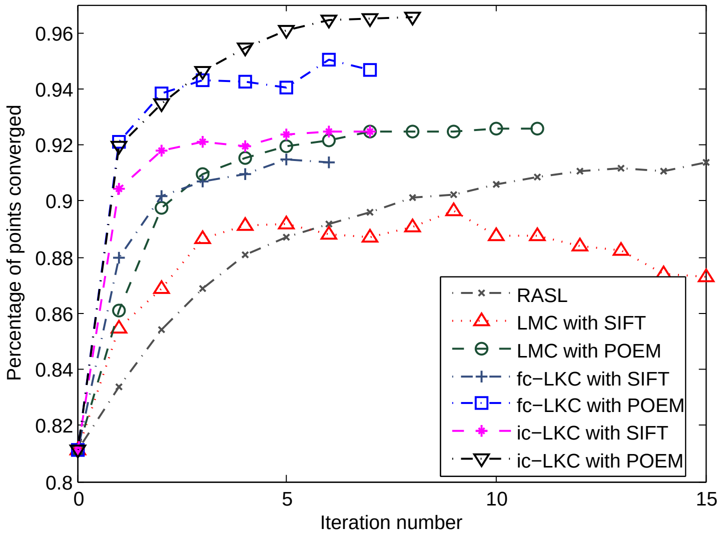


FIGURE 4.15 – Convergence curves of tracking landmarks on AR-face *Data 2*.

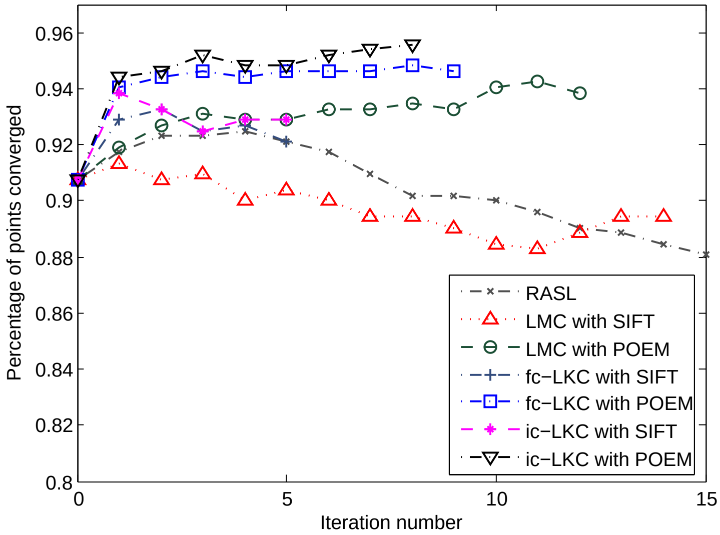


FIGURE 4.16 – Convergence curves of tracking landmarks on the SCface database.

recognition algorithm, results of two different face recognition methods respectively based on correlation coefficient and LBP (Local Binary Patterns) [1] are compared respectively.

Correlation coefficient measures the similarity between two face images, and a higher score implies a closer match. The correlation coefficient between images $\boldsymbol{I}$ and $\boldsymbol{J}$ is calculated by :

$$r = \frac{\sum_{\boldsymbol{x}}(\boldsymbol{I}(\boldsymbol{x}) - \bar{\boldsymbol{I}})(\boldsymbol{J}(\boldsymbol{x}) - \bar{\boldsymbol{J}})}{\sqrt{(\sum_{\boldsymbol{x}}(\boldsymbol{I}(\boldsymbol{x}) - \bar{\boldsymbol{I}})^2)(\sum_{\boldsymbol{x}}(\boldsymbol{J}(\boldsymbol{x}) - \bar{\boldsymbol{J}})^2)}} \tag{4.20}$$

where $\boldsymbol{x} = (x, y)$ is a vector of coordinates, $\boldsymbol{I}(\boldsymbol{x})$ and $\boldsymbol{J}(\boldsymbol{x})$ represent the pixel values at $\boldsymbol{x}$ in $\boldsymbol{I}$ and $\boldsymbol{J}$, respectively, $\bar{\boldsymbol{I}} = \sum_{\boldsymbol{x}} \boldsymbol{I}(\boldsymbol{x})$ and $\bar{\boldsymbol{J}} = \sum_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x})$ are the mean values of pixels.

LBP algorithm is widely used for face recognition. Detailed information about this descriptor is introduced in Appendix B. Here, we use the same LBP parameters as those in [1], and the face images are divided into 12 × 12 pixel subwindows. A face image is represented by concatenating histograms of LBP code estimated from all these subwindows. Chi-square distance of two histograms is used as similarity measure between two images.

(1) Results on FERET

In the FERET database, the “fa” face images are taken as the target set, and the “fb” face images form the query set. The target set contains known facial images, and the images in the query set are unknown facial images to be identified. We compute the face recognition accuracy using the output images of different alignment methods. We first consider three Congealing methods : Learned-Miller Congealing with SIFT, forward compositional and inverse compositional Lucas Kanade Congealing with POEM. We also include the results obtained by the alignment approach based upon BoRMaN facial point detector [115], whose code is available on Internet [4].

Here, we compute the rate that the correct answer is in the top n matches. The performance statistics are reported as cumulative match scores (see Figure 4.17). The horizontal axis of the graph is the rank and the vertical axis is the percentage of correct matches (face recognition rate). It is clear that using images aligned by the two LKC methods yields similar recognition accuracies which are obviously higher than the rate of using original unaligned images. The improvement of first-rank accuracy is around 10 percent. Besides, the results of our LKC methods are better than result LMC with SIFT within rank 10. This can be already predicted from the

[4] `http://www.doc.ic.ac.uk/~mvalstar/page5/page3/page3.html`

average images : if a query face is in a closer pose to a target one, face features can be better compared, and then a higher match score will be obtained. In our test, using images aligned by the BoRMaN algorithm does not give better recognition results than using unaligned images. This can be explained by the fact that transformation parameters are calculated from 2 simple mappings of points, the alignment performance is sensitive to point detection accuracy. That is, small errors produced by point detector will cause huge mis-alignment which has directly a great impact on recognition performance.

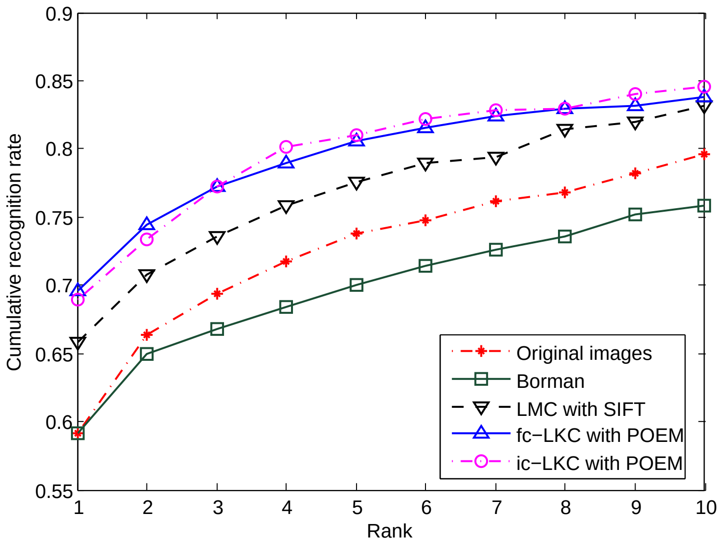


FIGURE 4.17 – Face recognition rates on the FERET database using correlation coefficient based algrithm.

The previous discussion has already shown the benefits of our alignment algorithm for face recognition regarding a very basic recognition method. We check now the performance of our alignment method regarding the more sophisticated LBP based face recognition approach. In fact, as pointed out in [1], LBP histogram is already robust to small mis-alignment, illustrating by a good recognition rate even on unaligned faces (see Figure 4.18). Even so, using our alignment algorithm, we still improve the final recognition rates : the 1-rank match rate is 97.8%. Also, our ic-LKC with POEM is slightly better than LMC with SIFT. Using images aligned by BoRMaN decreases the recognition accuracy as before, and this indicates that point detectors are not so reliable for face alignment.

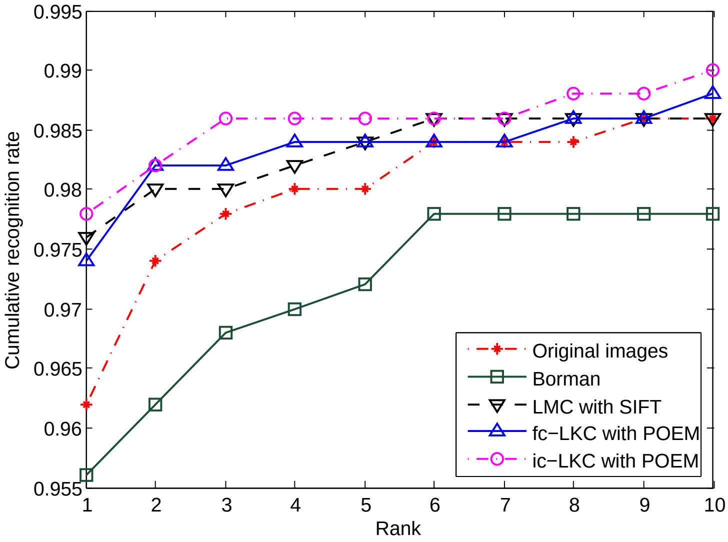


FIGURE 4.18 – Face recognition rates on the FERET database using LBP based algorithm.

(2) Results on LFW

In the previous experiment on FERET images, the point-based alignment method does not make expected effect. Besides, LKC in both forward and inverse formulation achieve similar performance. Therefore, here we concentrate on the comparison of ic-LKC and original LMC. The images aligned by LMC can be downloaded on the web[5]. The training set is used to find a threshold of distance which produces the highest correct classification rate. Using this threshold, we report the recognition rate on test image pairs. Also, correlation coefficient based and LBP based face recognition are respectively used[6], and the recognition rates are reported in Table 4.2. Compared with the results on unaligned images, the use of LMC improves the accuracy by 1% with correlation coefficient based algorithm, and 1.4% with LBP based algorithm, while the improvements by ic-LKC are respectively 2.6% and 4%. Hence, our alignment method is able to make significant positive effect in the task of identifying "uncontrolled" faces.

[5] `http://vis-www.cs.umass.edu/lfw/`

[6] When using correlation coefficient, a pair is matched (mismatched) if the value is higher (smaller) than the threshold. When using LBP, a pair is matched (mismatched) if the distance is smaller (higher) than the threshold.

	Unaligned	LMC	ic-LKC
Correlation coefficient	58.6%	59.6%	**61.2%**
LBP	66.2%	67.6%	**70.2%**

Table 4.2 – Correct classification rates on LFW images.

4.2.4.6 Complexity

Our alignment algorithm has three main steps : feature extraction, clustering and estimation of transformation parameters.

(1) Feature extraction

The complexity of this step obviously depends on the feature types. As pointed in [119], POEM and SIFT descriptors have similar complexity, but note that we use here only POEM codes not POEM histogram, that is why the extraction of POEM features in our LKC algorithms is faster than the extraction of SIFT in LMC.

(2) Clustering

The two factors affecting the time cost of this step are : feature dimension and number of clusters. While the dimension of SIFT feature in LMC is 16, POEM descriptor has 3 dimensions. Since we always adopt the same number of clusters, the clustering step in our LKC algorithms is considerably faster than that of LMC.

(3) Estimation of transformation parameters

The most important part of our alignment algorithm is the estimation of transformation parameters. Here, we compare the computing costs of parameter estimation with LMC, forward and inverse LKC on AR-face *Data 1*. This experiment was implemented in Matlab on a PC with $Core^{TM}$ 3.06G CPU, and the results are shown in Table 4.3. For each image at one iteration, LMC needs 0.4857s to estimate $\Delta\boldsymbol{v}$, while forward and inverse LKC respectively consume only 0.0944s and 0.0728s. The high computational cost of LMC is due to the searching procedure described in Section 2.4.2, in which deciding an update for one parameter requires recalculations of Equation (4.5). Between the two LKC approaches, inverse LKC is more computing efficient, it uses about 0.02s less than forward LKC, while the pre-computation time is only 0.02s for all images. That is, speed improvements of 20 percent are obtained by the inverse method. As discussed above, Least Squares Congealing is more computationally intensive than our algorithm, because at each iteration an estimation between every two images in the set is run. If running on a large number of images for multiple iterations, the proposed alignment method can significantly save time.

Moreover, the proposed ic-LKC costs less than five minutes to align 120 images of size 100 × 100, whereas RASL requires about ten minutes. Based on the above analysis, our method is more computational efficient than other approaches.

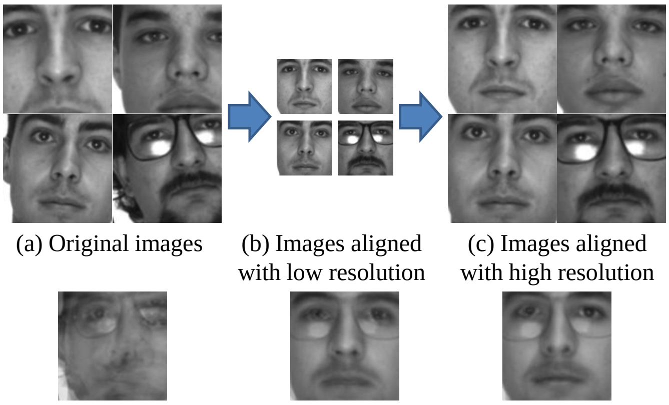
	Mean Time (seconds)		
	LMC	Forward LKC	Inverse LKC
Pre-computation	-	-	$\mathbf{0.0202 \pm 0.0017}$
Per image	0.4857 ± 0.0763	0.0944 ± 0.0208	$\mathbf{0.0728 \pm 0.0063}$
1000 images	485.7 ± 76.3	94.4 ± 20.8	$\mathbf{72.8 \pm 6.3}$

Table 4.3 – Mean time for estimation of $\Delta\boldsymbol{v}$ at each iteration.

4.3 Multi-resolution Lucas-Kanade Entropy Congealing

Most existing methods cast the joint alignment problem into a minimization of a discrepancy function calculated over all images, and then estimate transformation parameters by finding the local minimum of this function. For example, [53] defines an entropy function over images, and transformation parameters are estimated by iteratively searching updates which decrease the value of entropy. In [26], images are aligned by the minimization of a SSD function using gradient descent algorithms [10]. Unfortunately, large mis-alignment errors in faces (e.g., substantial translation and scaling, high-angle rotation in Figure 4.19 (a)) often lead to *undesirable* local minima in the cost function. The optimization process might get stuck in wrong minima, i.e., faces are not correctly aligned.

Section 4.2 presents a joint image alignment algorithm (LKC) where images are simultaneously aligned by minimizing an entropy function. This algorithm also suffers from the "local minima" problem when images undergo large mis-alignments. Here, we propose a multi-resolution solution to LKC where major mis-alignments are removed in coarse levels (images with lower resolution), and alignment is refined in finer levels (see an illustration in Figure 4.19). The new algorithm is referred to as multi-resolution LKC (MLKC). The idea of multi-resolution can also be applied to other joint alignment methods.



(a) Original images (b) Images aligned with low resolution (c) Images aligned with high resolution

(d) Averages of (a), (b), (c), respectively

FIGURE 4.19 – Illustration of multi-resolution joint alignment.

4.3.1 Multi-resolution joint alignment

The formulation in Section 4.2 works well on roughly aligned images cropped by Viola-Jones face detector [117]. Here, we focus on the alignment problem where faces undergo large mis-alignment errors, e.g., substantial translation and scaling, high-angle rotation as shown in Figure 4.19 (a). LKC estimates the update $\Delta \boldsymbol{v}$ using a first-order optimization algorithm which finds the local maximum of the cost function in Equation (4.5). However, large mis-alignment errors in faces probably cause unwanted local maxima of the cost function, i.e., images can not be aligned as desired.

In this situation, our interest is to increase the step size of update $\Delta \boldsymbol{v}$ in the first iterations to jump out of unwanted local maxima. Hence, a multi-resolution framework is applied to LKC : in coarse levels, images are processed with lower resolution to remove major mis-alignment errors ; in fine levels, alignment is refined using higher resolution. From the original images, we can easily obtain images of different smaller sizes using Gaussian smoothing and bicubic downsampling. Images in Level 0 are with original, high resolution. Level 1 consists of images with half the number of pixels along each axis. The downsampling rate for Level 2 is a quarter, and so on[7]. Similar coarse-to-fine method has been also used in other applications, e.g., facial point localization [25] and age simulation [106]. In our multi-resolution framework, the decision to change levels is made automatically, if the increase of cost function $\xi()$ is less than a pre-defined threshold T, or the maximum iteration

[7] Here, without loss of generality, we set the downsampling factor to 2, but it can take other values.

number $Iter_{max}$ is reached, the alignment process will change to next level with finer resolution. At the end of each level, the estimated parameters $\boldsymbol{v}$ are used as initial ones of the next level (since we use the downsampling factor of 2, translation parameters in $\boldsymbol{v}$ need to be doubled to fit larger images). The main steps of the multi-resolution ic-LKC are shown in Figure 4.20. The new algorithm is referred to as multi-resolution LKC (MLKC).

```
(1) Generate images for each level by downsampling ;
for $level = L$ to 0 do
  (2) Calculate POEM descriptor ;
  (3) Clustering and soft assignment ;
  repeat
    for $i = 1$ to $N$ do
      (4) Calculate $\Delta\boldsymbol{v}$ using Equation (4.16)
    end for
    (5) Update $\boldsymbol{v}$ using Equation (4.17)
    (6) Calculate new entropy $\xi_t$ using Equation (4.3)
  until $\xi_{t-1} - \xi_t < T_{level}$ or $Iter_{max}$ is reached
  (7) Calculate initial parameters for the next level
end for
```

FIGURE 4.20 – Main steps of proposed multi-resolution ic-LKC.

Another advantage of this multi-resolution framework is the improvement of computational speed. Suppose that N_x, N_v are respectively the numbers of pixels in original images and of warp components. In Level 0, the computational complexity for aligning N images at each iteration is $O(MNN_xN_v^2)$. The complexity of running an iteration in Level 1 decreases to $O(MNN_xN_v^2/4)$.

The idea of multi-resolution can also be applied to other joint alignment methods. Here, based on Least Squares Congealing (LSC) [26], we form multi-resolution LSC (MLSC) for comparison.

4.3.2 Experimental results

We evaluate the performance of proposed MLKC against four joint alignment methods including LKC, RASL [93], LSC [26], and MLSC. The maximum iteration number for all considered methods is 20, and we use two levels for multi-resolution alignment, i.e., 10 iterations for each level.

In our experiments, images are selected from two databases AR [87] and Yale B [40]. Face regions are first cropped manually from original images, and then aligned by transforming their eye centers to standard positions. Finally, these facial images are randomly warped for evaluating alignment algorithms (the acquisition of test data is similar to the way in [10]). The image size is 100×100. Here, we assume the type of mis-alignment is similarity, i.e., there are four transformation parameters : x-translation, y-translation, rotation, and scale.

4.3.2.1 Results on the AR database

100 AR facial images under neutral condition are selected for evaluation, and all faces are labeled by 22 landmarks (see Figure 2.13).

We calculate the new positions of landmarks in aligned images using estimated parameters. If the distance between a landmark and its relevant average position is smaller than a fixed threshold, the landmark is taken as converged. Regarding this criterion, a higher convergence rate stands for a better alignment. Considered alignment methods are run on images randomly warped with different standard deviations σ (mis-alignment magnitude increases with the value of σ), and the convergence rates are shown in Figure 4.21. RASL performs well on images with minor mis-alignments ($\sigma \leq 8$), but the convergence rates drop quickly when $\sigma \geq 9$. Although the overall convergence rates of LSC and MLSC are obviously lower than other three methods, we can still see the benefit of using a multi-resolution framework, i.e., MLSC works significantly better than LSC. Both LKC and MLKC achieve satisfactory results in a large range of mis-alignments ($\sigma \in [1, 14]$). However, there is a sharply decrease in the convergence rates of LKC when $\sigma \geq 15$ while the proposed MLKC is still able to perform stably.

To intuitively evaluate the alignment performance, averages of images aligned by different methods are shown in Figure 4.22, the standard deviation σ of random warp is 15 pixels. We can see that only the two LKC based methods are able to generate average images with distinct facial features, which means that faces are well aligned to a close pose. In some specific areas, e.g., mouth lips, the average image of MLKC is clearer than that of LKC.

The computational costs of these alignment methods on the AR images are shown in Table 4.3.2.1. It is clear that the proposed MLKC is more computationally efficient than other considered methods. The use of multi-resolution brings a significant speed improvement : MLSC and MLKC are respectively 20% and 30% faster than LSC

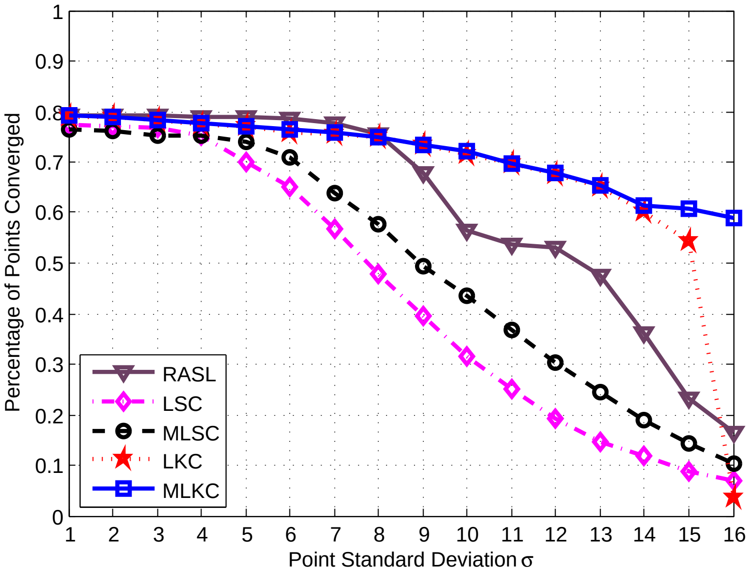


FIGURE 4.21 – Convergence rates with different standard deviations (AR).

and LKC. Two LSC based methods are computationally intensive because at each iteration an estimation between every two images in the set is run.

4.3.2.2 Results on the Yale B database

From the Yale B database [40] we select 100 frontal images of 10 subjects under 10 extremely difficult lighting conditions (examples are shown in Figure 4.23). Also, provided in this database, for each face, are the coordinates of three landmarks (centers of two eyes and mouth) which are used for calculating the convergence rates.

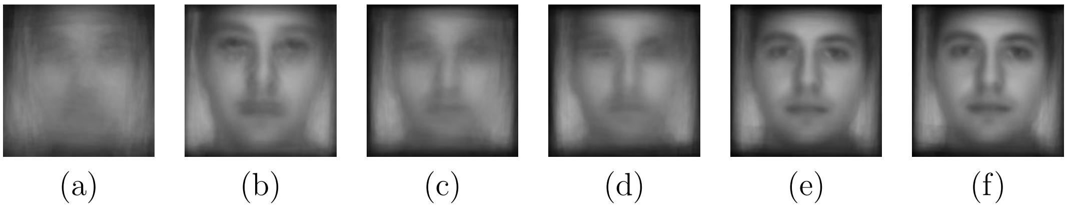


FIGURE 4.22 – Averages of AR images aligned by (a) original (unaligned), (b) RASL (c) LSC, (d) MLSC, (e) LKC, (f) MLKC.

	RASL	LSC	MLSC (level 1/0)	LKC	MLKC (level 1/0)
Per iter.	16.17	156.55	91.49/150.78	14.49	5.99/14.59
20 iter.	323.4	3131.0	2422.7	289.8	**205.8**

Table 4.4 – Computational cost of alignment methods (second).

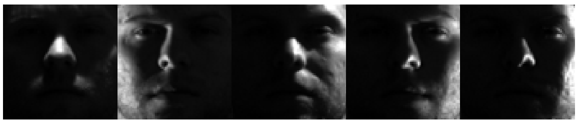


Figure 4.23 – Examples of Yale B images used in our experiments.

Since LSC and MLSC are both intensity-based methods which are sensitive to illumination changes, they are not capable of working on this set. The convergence rates of RASL, LKC, and MLKC are presented in Figure 4.24. RASL does not perform well on this challenging dataset, its overall convergence rates are clearly lower than those of the two LKC based methods. LKC and MLKC produce similar convergence rates when the standard deviation $\sigma \leq 9$. Then, MLKC shows its advantage when $\sigma \geq 10$. More precisely, there is a clear improvement when $\sigma = 10$, and LKC fails to achieve convincing results on extremely warped images ($\sigma \geq 11$) while MLKC still generates satisfactory convergence rates.

Averages of aligned images from this dataset are shown in Figure 4.25, and the standard deviation σ of random warp is 10 pixels. It is obviously that RASL fails to align these images with large mis-alignments. The result of LKC is acceptable, but the average image produced by the proposed MLKC has much clearer facial features (eyes, nose, and lips) than those in the average image of LKC.

4.4 Conclusion

This chapter first presents two Lucas-Kanade formulations of entropy congealing for joint image alignment (LKC), in which transformation parameters can be estimated simultaneously rather than in a sequential searching way as in Learned-Miller Congealing. In the forward formulation, the distribution field is taken as the template because it is invariant at each iteration. The inverse compositional LKC

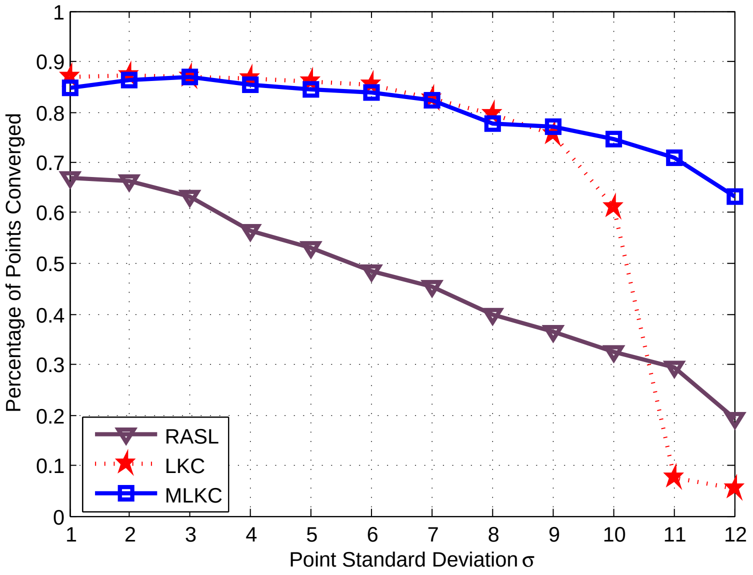
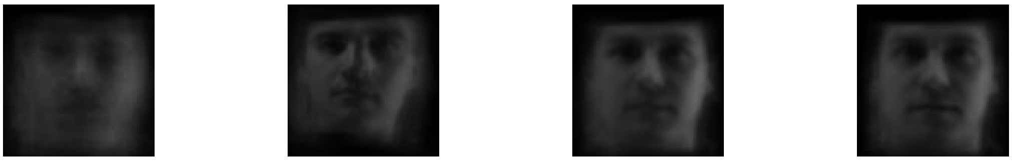


FIGURE 4.24 – Convergence rates with different standard deviations (Yale B).

method is obtained by switching the roles of template and test image. This yields constant parts of *Jacobian* and *Hessian*, which can be precomputed to decrease computing complexity. Moreover, we improve the alignment performance by combining Congealing with POEM descriptor, instead of SIFT. We conduct comparison experiments on five image sets respectively from AR, SCface, FERET, LFW databases, and surveillance videos. The benefits of LKC have extensively been verified using different evaluation protocols, including average image, convergence rate, face recognition performance, and complexity. The experimental results indicate the proposed



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

FIGURE 4.25 – Averages of Yale B images aligned by (a) original (unaligned), (b) RASL (c) LKC, (d) MLKC.

LKC shows better performance than other alignment methods, and it has certain robustness to variations in illumination and image quality. Concerning the complexity, the inverse LKC is more efficient than other considered approaches. Compared to the forward formulation, the inverse method produces a speed improvement of 20 percent.

In order to increase the robustness to large mis-alignments, this chapter then proposes a multi-resolution solution to entropy congealing. In coarse levels, images are processed with lower resolution to remove major mis-alignment errors, and then alignment is refined in finer levels. We test the proposed algorithm against several single resolution methods including LKC, Least Squares Congealing (LSC), RASL, and multi-resolution LSC. We conduct comparison experiments on images from two databases, AR and Yale B (under non-uniform illumination changes). Experimental results prove that the proposed algorithm outperforms other considered methods on images with large mis-alignment errors. The use of multi-resolution framework obviously improves both the robustness to mis-alignment and computational speed.

Up to this point, our LKC based methods are able to cope with most of the challenging factors for face alignment in video surveillance context mentioned in Section 1.2. More precisely, our methods are able to cope with a certain range of pose variation and expression changes, because they perform well on uncontrolled LFW images. The results on the AR *Data 2* and Yale B images prove the robustness of the LKC methods to illumination variations. The experiments on SCface and video surveillance images show that our methods are able to align low quality images where faces undergo low resolution, motion blur, and noise. In other words, the only remaining difficulty is occlusion. However, the LKC based methods require a clustering process for feature dimension reduction. When aligning images with occlusions, outlier pixels from occluded regions have negative impact on the clustering accuracy and may further lead to a decrease in performance of alignment. Hence, in the following chapter, we focus on finding a solution to align faces with occlusion.

Chapitre 5

Unsupervised Joint Face Alignment with Gradient Correlation Coefficient

5.1 Introduction

Faces usually suffer from occlusions and large lighting changes in video surveillance applications. However, most existing face alignment methods have just been evaluated in controlled environments without concerning occlusions nor lighting changes. For example, the LKC based methods presented in Chapter 4 require a clustering process for feature dimension reduction. When aligning images with occlusions, outlier pixels from occluded regions inevitably have negative impact on the clustering accuracy and further lead to a decrease in performance of alignment. Actually, common face detectors do not perform well under these challenging conditions, which makes face alignment phase more necessary. Recently, Roh et al. [97] proposed an occlusion-robust face alignment approach based on a set of feature detectors and the random sample consensus (RANSAC) strategy [37]. However, it is based on a shape model which is too complex for video surveillance applications. In the researches of [113, 114], gradient correlation coefficient, an efficient feature for face analysis under occlusions and illumination changes, was employed as a performance criterion for image-to-image alignment. Nevertheless, this approach requires pre-defined templates, and the experiment results are presented using templates related to the same subjects as the test images, i.e., the identities of test images are already known. This seems not so logical in a face recognition task, because the final objective of face recognition is to identify the input facial images.

In this chapter, the use of gradient correlation coefficient is extended to the *alignment of an image set*. For this purpose, we propose an unsupervised joint alignment framework, referred to as “Gradient Correlation Congealing (GCC)”, which aligns an image ensemble by maximizing a sum of gradient correlation coefficient function defined over all images. Two different formulations (GCC-1 and GCC-2) are developed respectively regarding the role of template. More precisely, GCC-1 uses the rest of images as templates within an iteration, while GCC-2 employs the held out image as template. The main advantages of GCC based methods are : (1) they work in an unsupervised manner, (2) they have no requirement of pre-defined templates, (3) they are robust to both *occlusions* and *non-uniform illumination changes*. The proposed algorithms are tested against four typical joint alignment methods including Least Squares Congealing [26], Learned-Miller Congealing [53], Lucas-Kanade entropy Congealing, and RASL (robust alignment by sparse and low-rank decomposition) [93] on images taken from two challenging face databases : AR [87] and Yale B [40]. Experimental results prove the efficiency of the proposed alignment approaches under different conditions, especially when faces are partially occluded, our algorithms perform much better than other considered methods. Compared with GCC-1, GCC-2 is more robust to large mis-alignment errors.

The remainder of the chapter is arranged as follows : we first introduce the gradient correlation coefficient and its application to image-to-image alignment in Section 5.2. The details of our joint alignment algorithm are presented in Section 5.3. The experimental results are analyzed in Section 5.4, and conclusions are finally given in Section 5.5.

5.2 Image-to-image Alignment with Gradient Correlation Coefficient

The feature we adopt in this work is gradient orientation which has been proved as an illumination-robust feature for face analysis in [19, 23, 51]. The work of [112] employed a FFT-based correlation of gradient orientation for image registration. Recently, this gradient correlation coefficient has been applied for the alignment of two facial images with occlusions [113, 114]. This section introduces the basic knowledge of gradient correlation coefficient (Section 5.2.1) and its application to image-to-image alignment (Section 5.2.2).

5.2.1 Gradient-based correlation coefficient

Given an image $\boldsymbol{I}_i(\boldsymbol{x})$ where $\boldsymbol{x} = [x, y]^T$ denotes a vector containing the pixel coordinates, its gradient can be calculated by :

$$\boldsymbol{G}_i(\boldsymbol{x}) = [\boldsymbol{G}_{i,x}(\boldsymbol{x}), \boldsymbol{G}_{i,y}(\boldsymbol{x})] = \left[\frac{\partial \boldsymbol{I}_i(\boldsymbol{x})}{\partial x}, \frac{\partial \boldsymbol{I}_i(\boldsymbol{x})}{\partial y}\right] \tag{5.1}$$

For representation convenience, we store $\boldsymbol{G}_{i,x}$, $\boldsymbol{G}_{i,y}$ in lexicographic ordering, then we have N-dimensional vectors $\boldsymbol{g}_{i,x}$, $\boldsymbol{g}_{i,y}$ where $\boldsymbol{g}_{i,x}(k) = \boldsymbol{G}_{i,x}(\boldsymbol{x}_k)$, $\boldsymbol{g}_{i,y}(k) = \boldsymbol{G}_{i,y}(\boldsymbol{x}_k)$, $k \in [1, N]$, N is the number of pixels in $\boldsymbol{I}_i$.

The gradient-based correlation coefficient between two images $\boldsymbol{I}_i$ and $\boldsymbol{I}_j$ can be defined by :

$$\psi(\boldsymbol{I}_i, \boldsymbol{I}_j) = \sum_{k=1}^{N} \boldsymbol{r}_i(k)\boldsymbol{r}_j(k) cos(\boldsymbol{\phi}_i(k) - \boldsymbol{\phi}_j(k)) \tag{5.2}$$

where $\boldsymbol{r}_i(k)$ and $\boldsymbol{\phi}_i(k)$ are respectively the gradient magnitude and orientation of the kth pixel in $\boldsymbol{I}_i$:

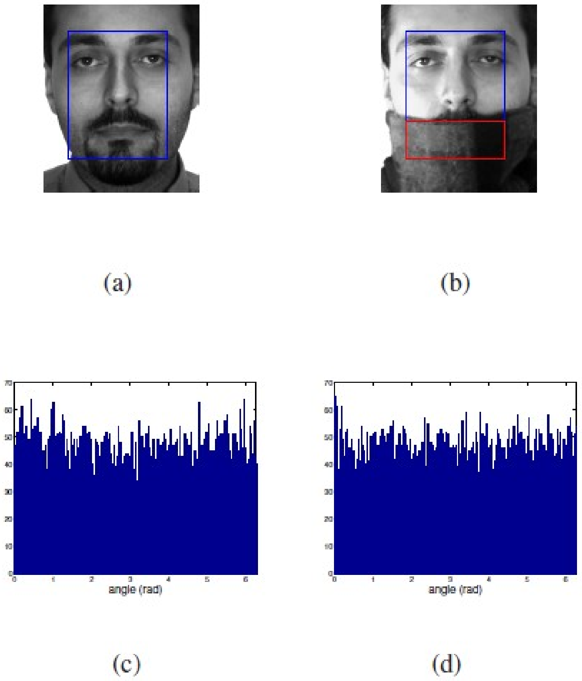
$$\boldsymbol{r}_i(k) = \sqrt{\boldsymbol{g}^2_{i,x}(k) + \boldsymbol{g}^2_{i,y}(k)} \tag{5.3}$$

$$\boldsymbol{\phi}_i(k) = arctan \frac{\boldsymbol{g}_{i,y}(k)}{\boldsymbol{g}_{i,x}(k)} \tag{5.4}$$

Chen et al. [19] pointed out that gradient magnitude varies drastically with the changes in illumination conditions. To improve the robustness of gradient-based correlation coefficient, normalized gradient $\widetilde{\boldsymbol{g}}_i = [\widetilde{\boldsymbol{g}}_{i,x}, \widetilde{\boldsymbol{g}}_{i,y}]$ can be used, where $\widetilde{\boldsymbol{g}}_{i,x}(k) = \boldsymbol{g}_{i,x}(k)/\boldsymbol{r}_i(k)$ and $\widetilde{\boldsymbol{g}}_{i,y}(k) = \boldsymbol{g}_{i,y}(k)/\boldsymbol{r}_i(k)$. Hence, normalized gradient correlation coefficient is $(\boldsymbol{r}_i(k) = 1, \forall i, k)$:

$$\psi(\boldsymbol{I}_i, \boldsymbol{I}_j) = \sum_{k=1}^{N} cos(\boldsymbol{\phi}_i(k) - \boldsymbol{\phi}_j(k)) \tag{5.5}$$

Now we discuss the properties of gradient correlation coefficient in Equation (5.5) between images with occlusions. Figure 5.1 (a) and (b) show a pair of facial images, one of which is partially occluded by a scarf. Since the occluded regions are mostly dissimilar to faces, it is not unreasonable to assume that the difference in gradient orientation $\Delta\boldsymbol{\phi}(k) = \boldsymbol{\phi}_i(k) - \boldsymbol{\phi}_j(k)$ can take any value in the range $[0, 2\pi]$ with equal probability. That is, we can assume that $\Delta\boldsymbol{\phi}(k)$ is generated by a uniform distribution defined over $[0, 2\pi]$. Hence, it is not difficult to find out that Equation (5.5) approximately equals to zero when it is calculated over occlusions. (As shown in Figure 5.1 (c) and (d), the distribution of $\Delta\boldsymbol{\phi}(k)$ in occluded region is very similar



(a)

(b)

(c)

(d)

FIGURE 5.1 – (a) is the template image, and (b) is the image to be aligned. (c) is the distribution of $\Delta\boldsymbol{\phi}(k)$ calculated over the occluded rectangle region of (b). (d) shows an uniform distribution of samples obtained with Matlab's rand function [114].

to an uniform distribution.) That is, occlusions do not bias the overall measure of similarity between two faces.

Using the normalized gradient $\widetilde{\boldsymbol{g}}_i$, we can easily obtain :

$$\psi(\boldsymbol{I}_i, \boldsymbol{I}_j) = \sum_{k=1}^{N} [\widetilde{\boldsymbol{g}}_{i,x}(k)\widetilde{\boldsymbol{g}}_{j,x}(k) + \widetilde{\boldsymbol{g}}_{i,y}(k)\widetilde{\boldsymbol{g}}_{j,y}(k)] \tag{5.6}$$

5.2.2 Image-to-image alignment with gradient correlation coefficient

Image-to-image alignment algorithms assume that a test image and a pre-defined template image (Figure 5.2) are related by a transformation :

$$\boldsymbol{I_j}(\boldsymbol{x}) = \boldsymbol{I}_i(\boldsymbol{W}(\boldsymbol{x}; \boldsymbol{p})) \tag{5.7}$$

where $\boldsymbol{I}_j$, $\boldsymbol{I}_i$ respectively denote the template and the test images, $\boldsymbol{p}$ is the vector of transformation parameters, and $\boldsymbol{W}(\boldsymbol{x}; \boldsymbol{p})$ stands for a warp.

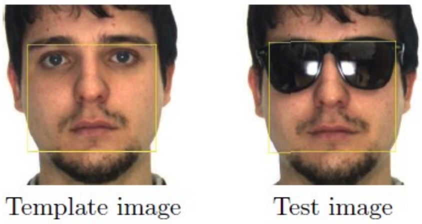


FIGURE 5.2 – An example of image-to-image alignment problem.

The goal of image-to-image alignment algorithms is to find the transformation which maximizes the similarities between two images. Section 5.2.2.1 introduces the formulation of alignment problem with gradient correlation coefficient, and Section 5.2.2.2 discusses an optimization algorithm of this alignment problem.

5.2.2.1 Alignment problem definition

In [114], gradient-based correlation coefficient was used as similarity measure, the alignment problem can be defined as the maximization of :

$$\psi(\boldsymbol{I}_i(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})), \boldsymbol{I}_j) \tag{5.8}$$

where $\boldsymbol{I}_j$, $\boldsymbol{I}_i$ respectively denote the template and test images, $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})$ stands for a warp.

Using the Lucas-Kanade algorithm [10], we assume that the current $\boldsymbol{p}$ is known, then $\boldsymbol{p}$ can be iteratively updated using $\Delta\boldsymbol{p}$:

$$\boldsymbol{p} \leftarrow \boldsymbol{p} + \Delta\boldsymbol{p} \tag{5.9}$$

However, a first order Taylor expansion of $\widetilde{\boldsymbol{g}}_i$ with respect to $\Delta\boldsymbol{p}$ yields a linear function of $\Delta\boldsymbol{p}$ which is maximized when $\Delta\boldsymbol{p} \to \infty$. To solve this problem, a new gradient correlation coefficient function which exactly equals to Equation (5.6) can be used :

$$\psi(\boldsymbol{I}_i, \boldsymbol{I}_j) = \frac{\sum_{k=1}^{N} \left[\widetilde{\boldsymbol{g}}_{i,x}(k)\widetilde{\boldsymbol{g}}_{j,x}(k) + \widetilde{\boldsymbol{g}}_{i,y}(k)\widetilde{\boldsymbol{g}}_{j,y}(k)\right]}{\sqrt{\sum_{k=1}^{N} \left[\widetilde{\boldsymbol{g}}_{i,x}^2(k) + \widetilde{\boldsymbol{g}}_{i,y}^2(k)\right]}} \tag{5.10}$$

Equation (5.10) can also be written in a vector expression :

$$\psi(\boldsymbol{I}_i, \boldsymbol{I}_j) = \frac{\widetilde{\boldsymbol{g}}_{i,x}^T \widetilde{\boldsymbol{g}}_{j,x} + \widetilde{\boldsymbol{g}}_{i,y}^T \widetilde{\boldsymbol{g}}_{j,y}}{\sqrt{\widetilde{\boldsymbol{g}}_{i,x}^T \widetilde{\boldsymbol{g}}_{i,x} + \widetilde{\boldsymbol{g}}_{i,y}^T \widetilde{\boldsymbol{g}}_{i,y}}} \tag{5.11}$$

5.2.2.2 Optimization of cost function

In Lucas-Kanade framework, the cost function Equation (5.8) becomes :

$$\begin{aligned}&\psi(\boldsymbol{I}_i(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}+\boldsymbol{\Delta p})),\boldsymbol{I}_j)\\ =&\frac{\widetilde{\boldsymbol{g}}_{i,x}^T\left[\boldsymbol{p}+\boldsymbol{\Delta p}\right]\widetilde{\boldsymbol{g}}_{j,x}+\widetilde{\boldsymbol{g}}_{i,y}^T\left[\boldsymbol{p}+\boldsymbol{\Delta p}\right]\widetilde{\boldsymbol{g}}_{j,y}}{\sqrt{\widetilde{\boldsymbol{g}}_{i,x}^T\left[\boldsymbol{p}+\boldsymbol{\Delta p}\right]\widetilde{\boldsymbol{g}}_{i,x}\left[\boldsymbol{p}+\boldsymbol{\Delta p}\right]+\widetilde{\boldsymbol{g}}_{i,y}^T\left[\boldsymbol{p}+\boldsymbol{\Delta p}\right]\widetilde{\boldsymbol{g}}_{i,y}\left[\boldsymbol{p}+\boldsymbol{\Delta p}\right]}}\end{aligned} \tag{5.12}$$

where the symbol $\boldsymbol{g}_{i,x}\left[\boldsymbol{p}\right]$ represents a vector obtained by writing $\boldsymbol{G}_{i,x}(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))$ in lexicographic ordering.

Because $\widetilde{\boldsymbol{g}}_{i,x}\left[\boldsymbol{p}\right]\equiv cos\boldsymbol{\phi}_i\left[\boldsymbol{p}\right]$, $\widetilde{\boldsymbol{g}}_{i,y}\left[\boldsymbol{p}\right]\equiv sin\boldsymbol{\phi}_i\left[\boldsymbol{p}\right]$, the first order Taylor expansion of $\widetilde{\boldsymbol{g}}_{i,x}\left[\boldsymbol{p}+\boldsymbol{\Delta p}\right](k)$ is :

$$\begin{aligned}\widetilde{\boldsymbol{g}}_{i,x}\left[\boldsymbol{p}+\boldsymbol{\Delta p}\right](k)&\approx cos\boldsymbol{\phi}_i\left[\boldsymbol{p}\right](k)+\frac{\partial cos\boldsymbol{\phi}_i\left[\boldsymbol{p}\right](k)}{\partial\boldsymbol{p}}\Delta\boldsymbol{p}\\ &=cos\boldsymbol{\phi}_i\left[\boldsymbol{p}\right](k)-sin\boldsymbol{\phi}_i\left[\boldsymbol{p}\right](k)\boldsymbol{j}\left[\boldsymbol{p}\right](k)\Delta\boldsymbol{p}\end{aligned} \tag{5.13}$$

where $\boldsymbol{j}_i\left[\boldsymbol{p}\right](k)$ is a $1\times n$ (n equals to the number of parameters of $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})$) vector defined by :

$$\boldsymbol{j}_i\left[\boldsymbol{p}\right](k)=\frac{cos\boldsymbol{\phi}_i\left[\boldsymbol{p}\right](k)\frac{\partial\boldsymbol{g}_{i,y}[\boldsymbol{p}](k)}{\partial\boldsymbol{p}}-sin\boldsymbol{\phi}_i\left[\boldsymbol{p}\right](k)\frac{\partial\boldsymbol{g}_{i,x}[\boldsymbol{p}](k)}{\partial\boldsymbol{p}}}{\sqrt{\boldsymbol{g}_{i,x}^2\left[\boldsymbol{p}\right](k)+\boldsymbol{g}_{i,y}^2\left[\boldsymbol{p}\right](k)}} \tag{5.14}$$

Equation (5.13) can be rewritten in a vector expression :

$$\widetilde{\boldsymbol{g}}_{i,x}\left[\boldsymbol{p}+\boldsymbol{\Delta p}\right]\approx cos\boldsymbol{\phi}_i\left[\boldsymbol{p}\right]-\boldsymbol{S}_\phi\left[\boldsymbol{p}\right]\odot\boldsymbol{J}_i\left[\boldsymbol{p}\right]\Delta\boldsymbol{p} \tag{5.15}$$

where $\boldsymbol{S}_\phi\left[\boldsymbol{p}\right]$ stands for a $N\times n$ matrix whose kth row has n elements all equal to $sin\phi_i\left[\boldsymbol{p}\right](k)$, $\boldsymbol{J}_i$ denotes a $N\times n$ matrix whose kth row equals to $\boldsymbol{j}_i\left[\boldsymbol{p}\right](k)$, and $\odot$ is the Hadamard product. Similarly, $\widetilde{\boldsymbol{g}}_{i,y}\left[\boldsymbol{p}+\boldsymbol{\Delta p}\right](k)$ can be written as :

$$\widetilde{\boldsymbol{g}}_{i,y}\left[\boldsymbol{p}+\boldsymbol{\Delta p}\right]\approx sin\boldsymbol{\phi}_i\left[\boldsymbol{p}\right]+\boldsymbol{C}_\phi\left[\boldsymbol{p}\right]\odot\boldsymbol{J}_i\left[\boldsymbol{p}\right]\Delta\boldsymbol{p} \tag{5.16}$$

where $\boldsymbol{C}_\phi\left[\boldsymbol{p}\right]$ stands for a $N\times n$ matrix whose kth row has n elements all equal to $cos\phi_i\left[\boldsymbol{p}\right](k)$.

By plugging Equations (5.15) and (5.16) into (5.12), we have a target function of $\Delta\boldsymbol{p}$:

$$\psi(\Delta \boldsymbol{p}) = \frac{q_p + \boldsymbol{S}_{i,j}^T \boldsymbol{J}_i \Delta \boldsymbol{p}}{\sqrt{N + \Delta \boldsymbol{p}^T \boldsymbol{J}_i^T \boldsymbol{J}_i \Delta \boldsymbol{p}}} \tag{5.17}$$

where $q_p = cos\boldsymbol{\phi}_j^T cos\boldsymbol{\phi}_i + sin\boldsymbol{\phi}_j^T sin\boldsymbol{\phi}_i$ and $\boldsymbol{S}_{i,j}$ represents a $N \times 1$ vector whose kth element equals to $sin(\boldsymbol{\phi}_j(k) - \boldsymbol{\phi}_i\left[\boldsymbol{p}\right](k))$.

The maximization of Equation (5.12) can be achieved by using the result in [36]. In particular, the maximum value is attained when [114] :

$$\Delta \boldsymbol{p}_{i,j} = \lambda_{i,j}(\boldsymbol{J}_i^T \boldsymbol{J}_i)^{-1} \boldsymbol{J}_i^T \boldsymbol{S}_{i,j} \tag{5.18}$$

where $\lambda_{i,j}$ is the reciprocal of normalized gradient correlation coefficient between $\boldsymbol{I}_i(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))$ and $\boldsymbol{I}_j$.

5.3 Proposed Algorithm : Gradient Correlation Congealing

Most image-to-image face alignment approaches are based on pre-defined templates. In [114], templates related to the same subjects as the test images are used to present experiment results, i.e., the identities of test images are already known. This seems not so logical in facial biometrics, because the final objective of face recognition is to identify the input facial images. To circumvent this template selection stage, we formulate an unsupervised joint alignment method called "Gradient Correlation Congealing (GCC)" which uses gradient correlation coefficient as the measure of alignment. Section 5.3.1 introduces the definition of GCC. We then respectively present two formulations to solve the alignment problem : GCC-1 in Section 5.3.2 and GCC-2 in Section 5.3.3.

5.3.1 Problem formulation

Congealing can be defined as a maximization/minimization of a similarity/discrepancy function $\xi()$ which is calculated over a set of images. The researches in [71, 53] use a sum-of-entropy as the cost function and in [26], a SSD function is employed as the measure of misalignment. Since we use correlation coefficient as the cost function where higher value means that images are more correlated, the joint alignment problem is written as :

$$arg \max_{\mathbb{P}} \xi(\mathbb{P}) \tag{5.19}$$

where $\mathbb{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_M\}$ stands for the set of warp parameters for different images, M is the number of images.

In the present work, the cost function is defined as a sum of separate functions, and the alignment problem in Equation (5.19) is solved by maximizing $\xi_i(\boldsymbol{p}_i)$ for each image in the set, that is :

$$\xi(\mathbb{P}) = \sum_{i=1}^{M} \xi_i(\boldsymbol{p}_i) \tag{5.20}$$

Intuitively, there are two ways to formulate the cost function $\xi_i(\boldsymbol{p}_i)$ depending on the definition of "template". More precisely, since joint alignment methods require no pre-defined template, at each time we can assume one image (or some images) as the template(s), and apply transformation to the rest. During the alignment process, the role of template is always changing. To make a clear distinction, the two different formulations are respectively referred to as GCC-1 (Section 5.3.2) and GCC-2 (Section 5.3.3).

5.3.2 GCC-1

In the first formulation, we try to find a warp $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}_i)$ for the held out image $\boldsymbol{I}_i$ and the rest of images are taken as templates. Hence, the cost function is defined as :

$$\xi_i(\boldsymbol{p}_i) = \sum_{\substack{j=1 \\ j\neq i}}^{M} \psi(\boldsymbol{I}_i(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}_i)), \boldsymbol{I}_j) \tag{5.21}$$

where $\psi()$ stands for gradient correlation coefficient function in Equation (5.10).

Since Equation (5.21) is non-linear, we can solve the maximization problem in the Lucas-Kanade framework. Supposing the current $\boldsymbol{p}_i$ is known, our goal is to iteratively find an update $\Delta\boldsymbol{p}_i$ which maximizes :

$$\xi_i(\Delta\boldsymbol{p}_i) = \sum_{\substack{j=1 \\ j\neq i}}^{M} \psi(\boldsymbol{I}_i(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}_i + \Delta\boldsymbol{p}_i)), \boldsymbol{I}_j) \tag{5.22}$$

Here, we obtain $\Delta\boldsymbol{p}_i$ by averaging the warps estimated between $\boldsymbol{I}_i$ and each image in the rest of the set :

$$\Delta\boldsymbol{p}_i = \frac{1}{M-1} \sum_{\substack{j=1 \\ j\neq i}}^{M} \Delta\boldsymbol{p}_{i,j} \tag{5.23}$$

where $\Delta\boldsymbol{p}_{i,j}$ denotes the update in Equation (5.18) which maximizes $\psi(\boldsymbol{I}_i(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}+\boldsymbol{\Delta p})),\boldsymbol{I}_j)$.

At each iteration, an estimated $\Delta\boldsymbol{p}_i$ is updated to the current transformation parameter $\boldsymbol{p}_i$:

$$\boldsymbol{p}_i \leftarrow \boldsymbol{p}_i + \Delta\boldsymbol{p}_i \tag{5.24}$$

The main steps of GCC-1 are shown in Figure 5.3. Note that during the calculation of $\Delta\boldsymbol{p}_i$ across the rest of image set at each iteration, there is an invariant part $(\boldsymbol{J}_i^T\boldsymbol{J}_i)^{-1}\boldsymbol{J}_i^T$ which can be pre-computed at the beginning of each iteration in order to reduce the computing time (Step (5) in Figure 5.3). Moreover, in our implementation, Steps (4) and (8) are optimized by directly calculating $\boldsymbol{J}_i$ and $\boldsymbol{S}_{i,j}$ from the gradients of images. This can improve the speed of our algorithm, because the inverse trigonometric functions in these two steps are slow and not necessary.

```
repeat
  for $i = 1$ to $M$ do
    (1) Warp $\boldsymbol{I}_i$ with $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}_i)$ to calculate $\boldsymbol{I}_i(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}_i))$
    (2) Calculate the gradient of $\boldsymbol{I}_i(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}_i))$
    (3) Calculate the gradient orientation $\boldsymbol{\phi}_i$ using Equation (5.4)
    (4) Calculate the Jacobian matrix $\boldsymbol{J}_i$ using Equation (5.14)
    (5) Calculate $(\boldsymbol{J}_i^T\boldsymbol{J}_i)^{-1}\boldsymbol{J}_i^T$
    for $j = 1$ to $M$, $j \neq i$ do
      (6) Warp $\boldsymbol{I}_j$ with $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}_j)$ to calculate $\boldsymbol{I}_j(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}_j))$
      (7) Calculate the gradient of $\boldsymbol{I}_j(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}_j))$
      (8) Calculate the gradient orientation $\boldsymbol{\phi}_j$ using Equation (5.4)
      (9) Calculate $\boldsymbol{S}_{i,j}$ and $\lambda_{i,j}$
      (10) Calculate $\Delta\boldsymbol{p}_{i,j}$ using Equation (5.18)
    end for
    (11) Calculate $\Delta\boldsymbol{p}_i$ using Equation (5.23)
    (12) Update $\boldsymbol{p}_i$ using Equation (5.24)
  end for
until $\xi()$ has converged
```

Figure 5.3 – Main steps of GCC-1.

However, the formulation of GCC-1 may not work well when images undergo large mis-alignment errors, e.g., substantial translation and scaling, high-angle rotation.

More precisely, the estimation of $\Delta\boldsymbol{p}_i$ in GCC-1 has similar effect to align $\boldsymbol{I}_i$ to the average image $\frac{1}{M-1}\sum_{\substack{j=1\\j\neq i}}^{M}\boldsymbol{I}_j$, since the rest of images are invariant within an iteration. Larger mis-alignments in $\boldsymbol{I}_j$ mean a blurrier average image, hence aligning $\boldsymbol{I}_i$ to this average image may result in a worse alignment performance. To circumvent this limitation, GCC-2 is proposed in Section 5.3.3.

5.3.3 GCC-2

In contrast to GCC-1, the alternative solution is to use the held out $\boldsymbol{I}_i$ as the template. Hence, the new cost function is formulated by :

$$\xi_i(\Delta\boldsymbol{p}_{-i}) = \sum_{\substack{j=1\\j\neq i}}^{M} \psi(\boldsymbol{I}_i, \boldsymbol{I}_j(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}_j + \Delta\boldsymbol{p}_{-i}))) \tag{5.25}$$

In Equation (5.25), the objective is to find a transformation $\Delta\boldsymbol{p}_{-i}$ which is applied to the rest of images. In this way, the alignment process is able to use more details of the image ensemble. Similarly, $\Delta\boldsymbol{p}_{-i}$ is calculated using :

$$\Delta\boldsymbol{p}_{-i} = \frac{1}{M-1}\sum_{\substack{j=1\\j\neq i}}^{M}\Delta\boldsymbol{p}_{j,i} \tag{5.26}$$

Then, the update process is defined by :

$$\boldsymbol{p}_j \leftarrow \boldsymbol{p}_j + \Delta\boldsymbol{p}_{-i} \tag{5.27}$$

The main steps of GCC-2 are shown in Figure 5.4. One major difference from GCC-1 is that the Jacobian matrix of GCC-2 is calculated in the inner loop. That is, GCC-2 is more computational intensive than GCC-1, but the increment of running time (around 20%) is acceptable comparing to the whole cost. Also, the update of transformation is different in Step (12). Note that $\Delta\boldsymbol{p}_{j,i}$, $\boldsymbol{S}_{j,i}$, and $\lambda_{j,i}$ are not equal to the symbols $\Delta\boldsymbol{p}_{i,j}$, $\boldsymbol{S}_{i,j}$, and $\lambda_{i,j}$ in Figure 5.3.

5.4 Experimental Results

This section evaluates the performance of proposed Gradient Correlation Congealing (GCC). It is compared with three joint alignment methods including Least Square Congealing (LSC) [26], Learned-Miller Congealing (LMC) [53], Lucas-Kanade

```
repeat
  for $i = 1$ to $M$ do
    (1) Warp $\boldsymbol{I}_i$ with $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}_i)$ to calculate $\boldsymbol{I}_i(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}_i))$
    (2) Calculate the gradient of $\boldsymbol{I}_i(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}_i))$
    (3) Calculate the gradient orientation $\boldsymbol{\phi}_i$ using Equation (5.4)
    for $j = 1$ to $M$, $j \neq i$ do
      (4) Warp $\boldsymbol{I}_j$ with $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}_j)$ to calculate $\boldsymbol{I}_j(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}_j))$
      (5) Calculate the gradient of $\boldsymbol{I}_j(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}_j))$
      (6) Calculate the gradient orientation $\boldsymbol{\phi}_j$ using Equation (5.4)
      (7) Calculate the Jacobian matrix $\boldsymbol{J}_j$ using Equation (5.14)
      (8) Calculate $(\boldsymbol{J}_j^T\boldsymbol{J}_j)^{-1}\boldsymbol{J}_j^T$
      (9) Calculate $\boldsymbol{S}_{j,i}$ and $\lambda_{j,i}$
      (10) Calculate $\Delta\boldsymbol{p}_{j,i}$ using Equation (5.18)
    end for
    (11) Calculate $\Delta\boldsymbol{p}_{-i}$ using Equation (5.26)
    for $j = 1$ to $M$, $j \neq i$ do
      (12) Update $\boldsymbol{p}_j$ using Equation (5.27)
    end for
  end for
until $\xi()$ has converged
```

FIGURE 5.4 – Main steps of GCC-2.

entropy Congealing (LKC), and with RASL (robust alignment by sparse and low-rank decomposition) [93]. Section 5.4.1 first introduces the databases, parameters, and evaluation methods used in the comparison experiments. Section 5.4.2 to Section 5.4.5 respectively present and discuss comparison results on images under different conditions.

5.4.1 Experiment Settings

1. Databases

We use the images from AR [87], Yale B [40] databases, and surveillance videos to design five scenarios for conducting comparison experiments :

- Test 1 : Images are taken under the neutral condition.
- Test 2 : Images are taken under extremely large illumination variations.
- Test 3 : Images are taken with occlusion under neutral illumination conditions.

- Test 4 : Images undergo both occlusion and non-uniform illumination variations.
- Test 5 : Video surveillance images undergo low resolution, motion blur, noise, and variations in pose and expression.

Here, we form three image sets from the AR database : *Data 1* consists of 100 frontal facial images corresponding to different subjects with neutral expression and homogeneous illumination (see an example in Figure 5.5 (a)) ; *Data 2* contains 100 frontal facial images of 50 subjects, and each subject has two images under the neutral illumination : one with sun glasses (Figure 5.5 (b)) and the other without occlusion (Figure 5.5 (a)) ; *Data 3* is composed of 100 frontal facial images of 25 subjects, and each subject has one image under the same neutral condition as in *Data 1* and three images with occlusion (sunglass) and non-uniform illumination changes (examples are shown in Figure 5.5 (b)∼(d)). In the following experiments, we use *Data 1* for Test 1, *Data 2* for Test 3, and *Data 3* for Test 4.

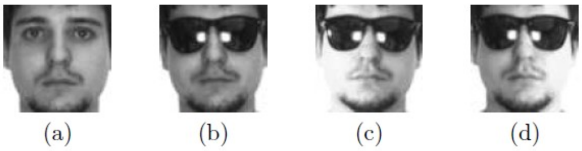


FIGURE 5.5 – Examples of test images from AR-face database. (a) is a face under neutral conditions and (b)∼(d) are faces with occlusion under different lighting conditions.

We also select 100 frontal images of 10 subjects under 10 extremely difficult lighting conditions from the Yale B database (three from the subset 4 and seven from the subset 5), images of one subject are shown in Figure 5.6 as example. These images are used for Test 2.

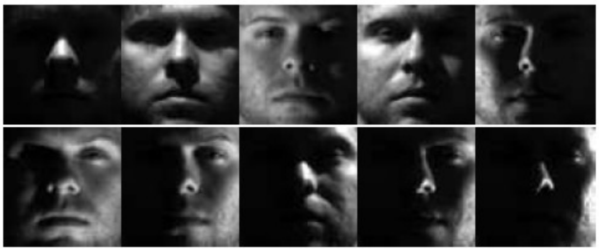


FIGURE 5.6 – Yale B images under 10 different illumination conditions used in our experiments.

The video surveillance image set used in Section 4.2.4 is also adopted here to evaluate the alignment performance on images of low resolution, motion blur and noise as well as variations in pose and expression (see examples in Figure 4.5.

In AR database, three feature points (two eyes and nose tip) have been labeled manually. Also, provided in the Yale B database, for each face, are the coordinates of three landmarks : two eye centers and mouth. These landmarks are used for evaluation in our experiments.

Occlusion is one of the most difficult factors in face analysis. A direct influence of occlusion is that there is no useful information in occluded regions. In fact, some lighting conditions may also result in this kind of influence. Take the image in Figure 5.5 (c) as example, gray values of pixels around the left cheek all equal to 255, i.e., no information can be caught in the overexposed region. This information losing problem makes our test data more challenging.

Because common face detectors do not work on these challenging images, face regions are cropped manually and then the eye centers are aligned to standard positions. In our experiments, the image size is 100 × 100. As shown in Figure 5.2, the gradient correlation coefficient is calculated over a center region of images. The size of this region of interest (ROI) is 60 × 60.

Similar to [10], randomly warped images are used as test data in our experiments. The left-top and right-bottom corner points of the ROI are selected as canonical points which are randomly perturbed with additive white Gaussian noise of a certain deviation σ. Then we can fit a similarity warp which transforms these canonical points to their perturbed positions. Finally, test images are warped using these randomly generated warps.

2. Parameter settings

We compare the performance of Gradient Correlation Congealing (GCC) with four joint alignment methods including Least Square Congealing (LSC) [26], Learned-Miller Congealing (LMC) [53], Lucas-Kanade entropy Congealing (LKC), and RASL [1] [93]. The code of original LMC is available on Internet [2], it uses 8 orientations for SIFT descriptor. For a fair comparison, we report here the results of LMC obtained with 4 orientations SIFT descriptors [82], because this results in better performance. In our experiments, results of all approaches are presented after 20 iterations unless they can converge within fewer iterations.

Here, we assume that the type of geometric mis-alignment is a similarity transformation, i.e., there are four parameters : x-translation, y-translation, in-plane rotation and scaling. Other transformations, e.g., affine transformation, are also adaptable for

[1] The Matlab code of RASL is available at `http://perception.csl.uiuc.edu/matrix-rank/rasl.html`

[2] `http://vis-www.cs.umass.edu/code/congealingcomplex/`

the proposed algorithm, but they may cause unwanted deformations in cropped (aligned) faces.

Here, we use standard deviations $\sigma = [1, 5]$ to generate random warps for evaluation.

3. Evaluation methods

By applying alignment algorithms to the randomly warped images, we evaluate the performance based on the two following criteria : average of cropped faces and convergence rates of landmarks (details of these evaluation methods are presented in Section 2.6). The facial images used here are difficult for face recognition because they undergo extreme illumination conditions or occlusion. In other words, face recognition accuracy may not be improved even after faces are well aligned. Hence, we do not use face recognition result as evaluation criterion in this chapter.

In the following experiments, all convergence rates are computed using a fixed threshold of two pixels. More precisely, we present the convergence rates of considered methods in two different ways : one is to track the convergence rates at each iteration to verify the convergence ability during alignment procedure ; the other option is to compare the convergence rates produced by aligning images warped with different deviations $\sigma = [1, 5]$.

5.4.2 Scenario 1 : Facial images under the neutral condition

We first assess the alignment performance on AR face *Data 1* which contains images taken under neutral conditions.

The averages of unaligned and aligned image sets are shown in Figure 5.7, here the standard deviation σ of randomly generated warps is three pixels. Intuitively, both face contour and facial features are unclear in the average of unaligned images. All the six alignment methods show the similar performance regarding this visual evaluation criterion, they yield average images with clearer facial features, e.g., eyes, nostrils, and lips. For further evaluation, the convergence rates of these alignment approaches at different iterations are drawn in Figure 5.8. It is obvious that the proposed GCC-2 is stable across the alignment process and it achieves the highest convergence rate at the end. The results of GCC-1 on this data are also satisfactory, its convergence rates are higher than those of other four methods. RASL works well within the first iterations, but there is a clear decrease in its performance after three iterations. The rest three methods are also capable of aligning images under this less challenging condition and their final convergence rates are sightly lower than that of GCC-1 and GCC-2.

We also test the alignment methods on images randomly warped with standard

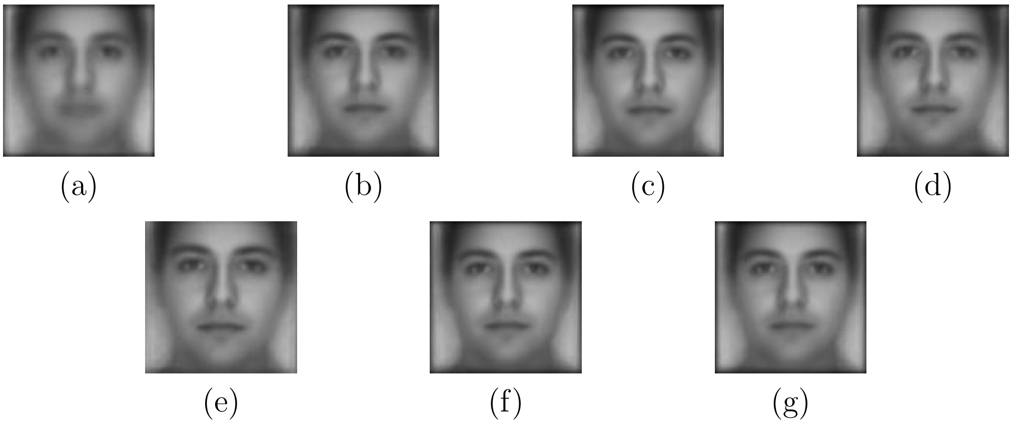


FIGURE 5.7 – Average images of AR face *Data 1* which are (a) unaligned, (b) aligned by LSC, (c) aligned by LMC, (d) aligned by LKC, (e) aligned by RASL, (f) aligned by GCC-1, (g) aligned by GCC-2.

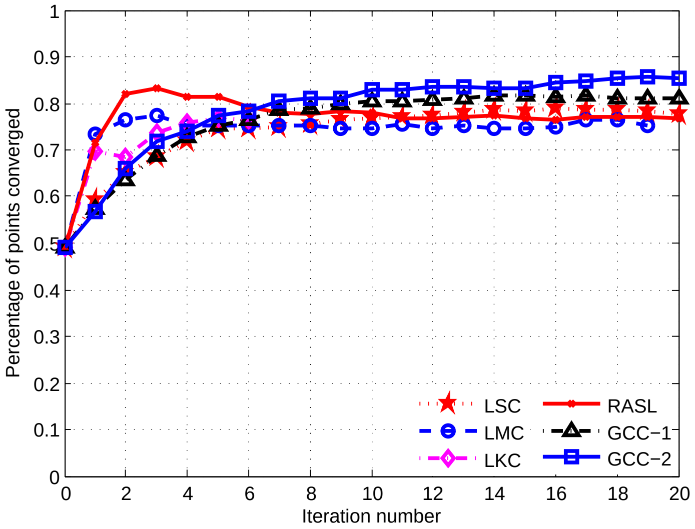


FIGURE 5.8 – Convergence rates at different iterations on the AR face *Data 1* ($\sigma = 3$).

deviations which range from 1 to 5. The convergence rates of landmarks can be seen in Figure 5.9. Our GCC-2 obviously produces higher convergence rates than other considered alignment methods over different values of standard deviation. GCC-1 performs well on images with minor mis-alignment errors ($\sigma \leq 2$), but the convergence rates decrease quickly with the increasing of mis-alignment magnitude. This can be explained by the discussion in Section 5.3.2. LKC, LSC, and RASL achieve similar convergence rates which are slightly higher than those of LMC. All these results prove the advantage of GCC-2 algorithm over other methods under the neutral condition.

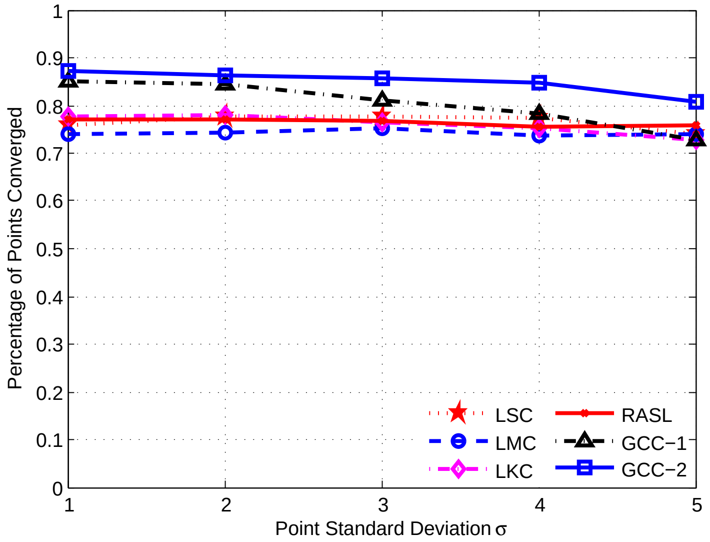


FIGURE 5.9 – Convergence rates with different standard deviations on the AR face *Data 1*.

5.4.3 Scenario 2 : Facial images under non-uniform illumination variations

In order to show the robustness of our algorithm to illumination changes, this section evaluate the alignment performance on the challenging Yale B images taken under non-uniform lighting variations (see Figure 5.6).

Using test images warped with a standard deviation of three pixels, the averages of unaligned and aligned images are shown in Figure 5.10. Due to the random warping, the average of unaligned images yields blurs around the facial features. It is clear that LSC does not work in this case. This can be predicted because LSC is an intensity-based method which is very sensitive to illumination changes. LMC and RASL seem not to be suitable for images under these extreme lighting changes, because the left and right parts of their average faces are not symmetrical. LKC and the two GCC based algorithms produce similar average images with clear inside and outside facial contours. The corresponding convergence rates of landmarks at different iterations are presented in Figure 5.11. As can be predicted from the average images, LSC, LMC, and RASL can not work stably on this image set. LKC is able to converge using fewer iterations and reaches a satisfying result. The two GCC based methods always show a stable convergence trend during the alignment procedure and it generates a final convergence rate similar to LKC.

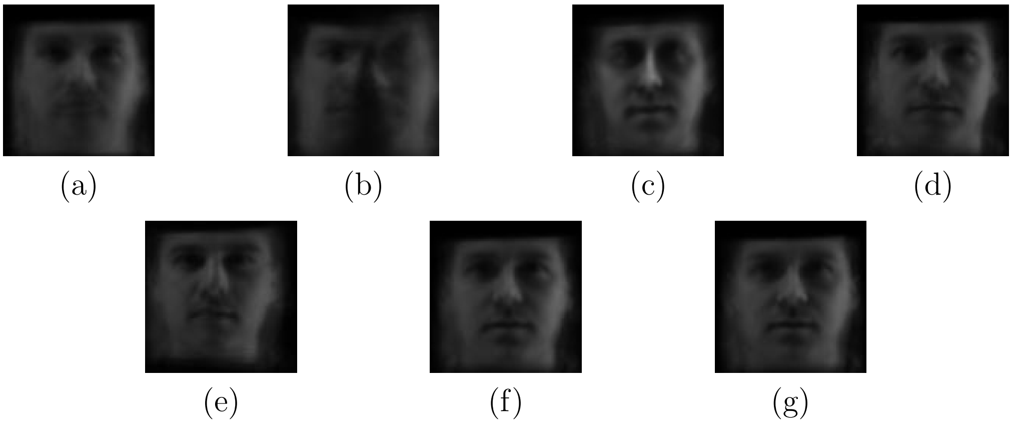

(a) (b) (c) (d)
(e) (f) (g)

FIGURE 5.10 – Average images of Yale B sets which are (a) unaligned, (b) aligned by LSC, (c) aligned by LMC, (d) aligned by LKC, (e) aligned by RASL, (f) aligned by GCC-1, (g) aligned by GCC-2.

Stated in Figure 5.12 are the convergence rates obtained by aligning randomly warped images with different standard deviations. It is clear that LSC and LMC still do not perform well on images warped with other standard deviations. The results of RASL are better than those of LSC and LMC, but obviously worse than the other three methods. In this case, LKC achieves the best performance over different deviations. This is because LKC adopts an effective feature descriptor POEM [119] which is robust to large variations in lighting conditions. The two GCC based algorithms produce similar convergence rates which are slightly lower than these of LKC, but still satisfying, i.e., our method performs well under non-uniform illumination changes.

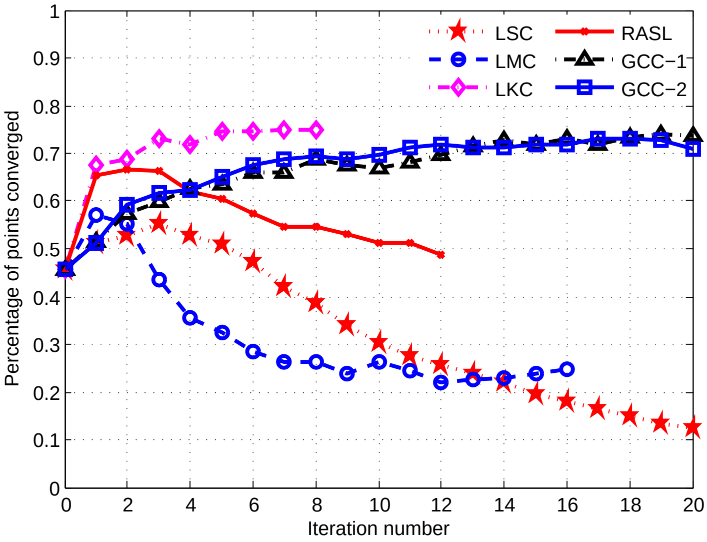


FIGURE 5.11 – Convergence rates at different iterations on Yale B Set ($\sigma = 3$).

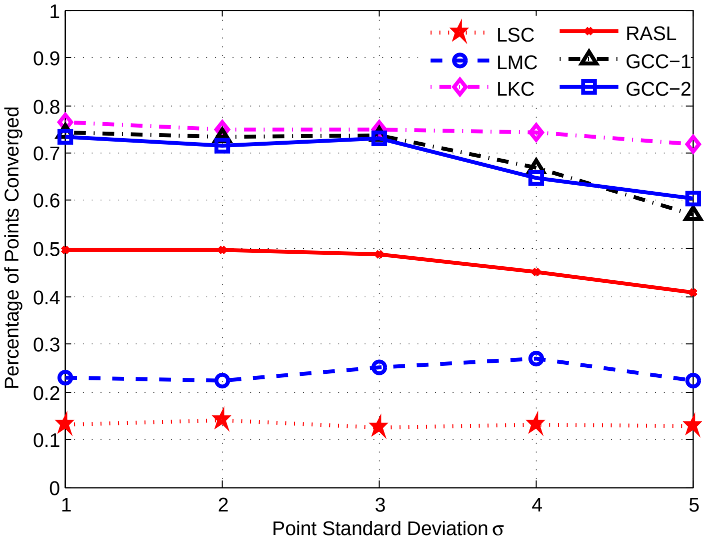
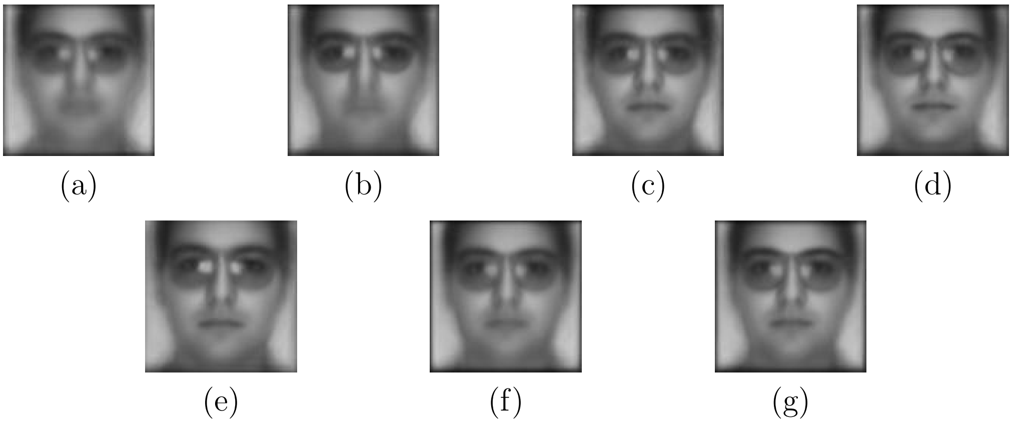


FIGURE 5.12 – Convergence rates with different standard deviations on Yale B set.

5.4.4 Scenario 3 : Facial images with occlusion under neutral illumination conditions

This section evaluates the performance of proposed algorithms on face with occlusion (sun glasses) taken under neutral lighting conditions.

Figure 5.13 shows average images of unaligned and aligned AR *Data 2*. After the random warps of a three-pixel deviation, the unaligned set has a blurry average image. LSC does not make an obvious improvement on these occluded images. The rest five methods are able to generate clearer average images which are visually similar. For further evaluation, the corresponding convergence rates of facial feature points at different iterations are presented in Figure 5.14. GCC-1 and GCC-2 show stable convergence trend, and their final convergence rates are higher than those of rest methods. Between the two proposed methods, GCC-2 performs better than GCC-1. RASL, LMC and LKC generate similar final results, but they do not work stably during the alignment procedure. Take RASL for example, it improves the convergence rates within first iterations, but there is an obvious decrease after four iterations. It seems that LSC is not able to work on these images, its final convergence rate is close to that of the unaligned images.


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

(e) (f) (g)

FIGURE 5.13 – Average images of AR face *Data 2* which are : (a) unaligned, (b) aligned by LSC, (c) aligned by LMC, (d) aligned by LKC, (e) aligned by RASL, (f) aligned by GCC-1, (g) aligned by GCC-2.

Convergence rates obtained by aligning randomly warped images with different standard deviations are shown in Figure 5.15. LSC are not able to align images warped with other values of standard deviation. RASL, LMC and LKC produce similar performance. GCC-1 is able to generate satisfactory results on images with minor

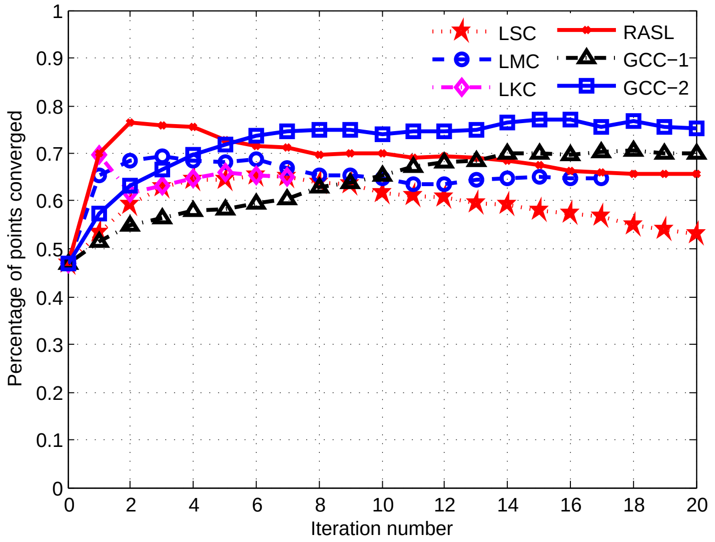


FIGURE 5.14 – Convergence rates at different iterations on the AR face *Data 2* ($\sigma = 3$).

mis-alignments, but its performance decreases sharply when $\sigma \geq 4$. The convergence rates of GCC-2 are satisfactory over all values of standard deviation.

5.4.5 Scenario 4 : Facial images with occlusion under non-uniform illumination variations

This section aims at proving the strength of our algorithm in the most challenging scenario where images undergo both occlusion (sunglass) and non-uniform illumination changes. Indeed, there is scarcely any specific results reported on joint image alignment with occlusions.

The averages of unaligned and aligned images sets are presented in Figure 5.16, and the standard deviation of randomly generated warps is three pixels. After the random warping, the average of unaligned images shows unclear face shape and inside features. It is clear that LSC does not work in this case, the average image becomes blurrier. Hence, this kind of intensity-based method is not capable of aligning images with occlusions. LMC has certain improvement in the average image, but there is still some blurs around the mouth and nose. The rest four alignment methods yield clear average images, and they perform similarly according to this visual evaluation.

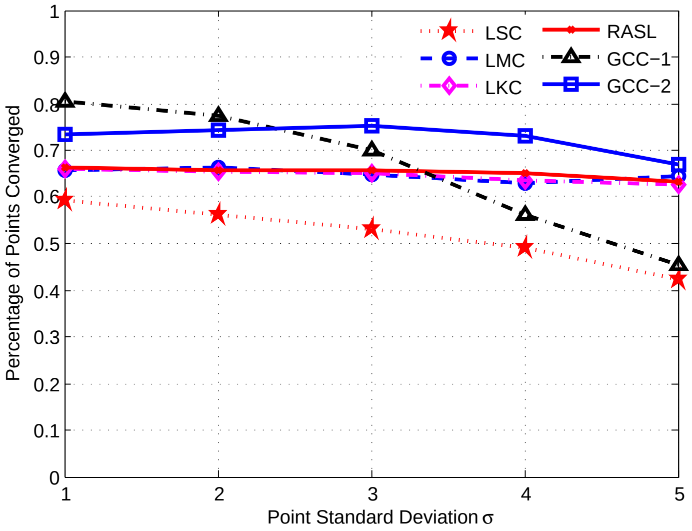
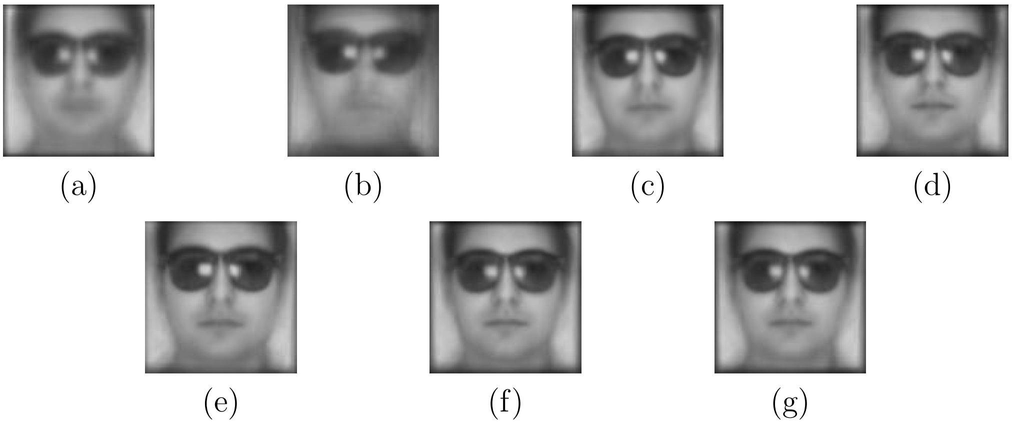


FIGURE 5.15 – Convergence rates with different standard deviations on the AR face *Data 2*.

The corresponding convergence rates of landmarks at different iterations are shown in Figure 5.17. Obviously LSC is not able to align this image set. The convergence rates of LMC and LKC decrease after the first few iterations, this instability might be because both methods are based on a clustering stage which is sensitive to outlier pixels. RASL does improve the convergence rates within the first few iterations, but it shows a clear decrease after three iterations. In contrast, the proposed GCC-2 always shows a stable convergence trend during the alignment procedure and it reaches the highest convergence rate. The results of GCC-1 are slightly worse than those of GCC-2. Here, the good performance of proposed algorithms is owing to the use of gradient correlation coefficient which is robust to occlusion (analyzed in Section 5.2.1).

Figure 5.18 illustrates the convergence rates obtained by aligning randomly warped images with different standard deviations. LSC still does not work on images warped with other values of standard deviation. LMC and LKC produce similar performance, and the convergence rates of RASL are clearly higher than these two methods. GCC-1 is able to generate satisfactory results on images with minor misalignments, but its performance decreases sharply when $\sigma = 5$. The convergence rates of GCC-2 are higher than those of the others over all values of standard deviation. That is, GCC-2 algorithm performs better than other considered methods on images



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

(e) (f) (g)

FIGURE 5.16 – Average images of AR face *Data 3* which are : (a) unaligned, (b) aligned by LSC, (c) aligned by LMC, (d) aligned by LKC, (e) aligned by RASL, (f) aligned by GCC-1, (g) aligned by GCC-2.

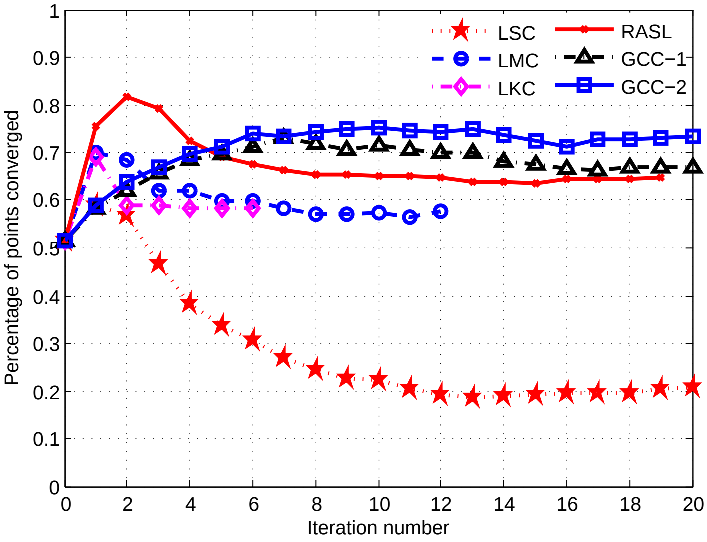


FIGURE 5.17 – Convergence rates at different iterations on the AR face *Data 3* ($\sigma = 3$).

undergoing both occlusion and non-uniform illumination variations.

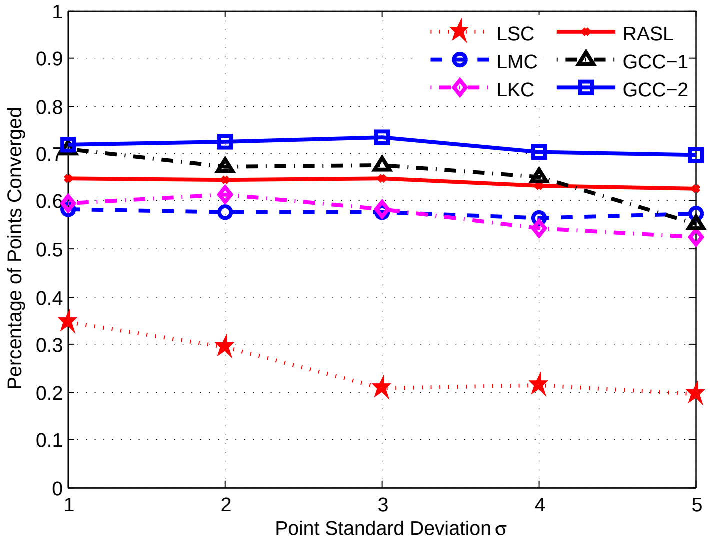


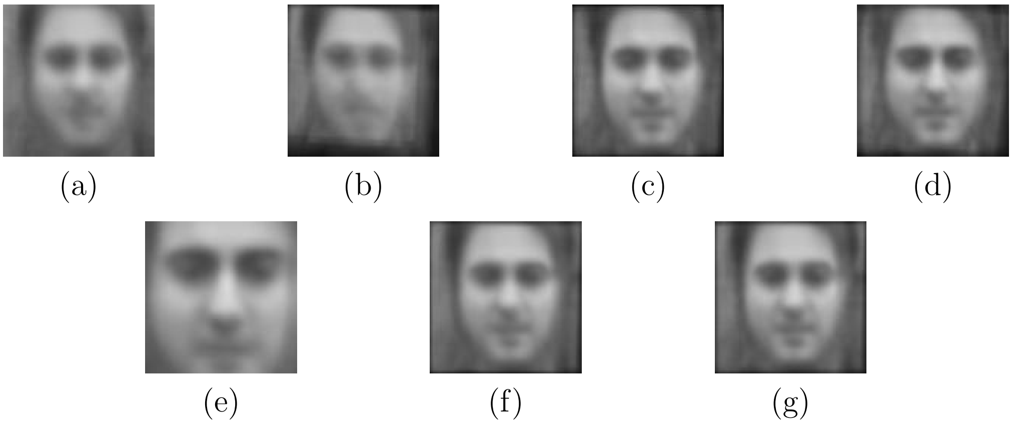
FIGURE 5.18 – Convergence rates with different standard deviations on the AR face *Data 3*.

5.4.6 Scenario 5 : Facial images extracted from surveillance videos

Here we evaluate the performance of these alignment methods on video surveillance images. Averages of unaligned and aligned images are shown in Figure 5.19. We can see that LSC does not work at all on this challenging set. It is interesting to see that the rest five joint alignment methods produce clearer average images than that of the original set (e.g., the regions of mouth and nose). Due to the motion blurs and noises in original images, it is difficult to make further comparison. Even though, these results prove that our GCC based methods are capable of working on video surveillance images.

5.5 Conclusion

In this chapter, we propose an unsupervised joint alignment framework where an image ensemble is aligned by maximizing a sum of gradient correlation coefficient function defined over all images. Two different formulations (GCC-1 and GCC-2) are developed respectively regarding the role of template. The novel algorithms are tested against four typical joint alignment methods including Least Squares Congealing, Learned-Miller Congealing, Lucas-Kanade entropy congealing, and RASL on images



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

(e) (f) (g)

Figure 5.19 – Averages of video surveillance images which are : (a) unaligned, (b) aligned by LSC, (c) aligned by LMC, (d) aligned by LKC, (e) aligned by RASL, (f) aligned by GCC-1, (g) aligned by GCC-2.

taken from two challenging face databases (AR and Yale B) and surveillance videos. Experimental results prove the efficiency of the proposed alignment approaches under different conditions, especially when faces are partially occluded, our algorithms perform much better than the other considered methods. Compared with GCC-1 which uses the rest of images as templates, GCC-2 (employing the held out image as template) is more efficient when faces undergo large mis-alignment errors. Although LKC performs slightly worse than GCC-2 on occluded images, the performance of LKC is still satisfactory.

Up to this point, the proposed methods (LKC and GCC-2) have shown certain robustness to all the challenging factors mentioned in Section 1.2. More precisely, our methods are able to cope with low resolution, motion blur and noise as well as a certain range of pose variation and expression changes, because they perform well on video surveillance images. The results on the Yale B images prove the robustness of LKC and GCC-2 methods to illumination variations (even extremely difficult ones). The experiments on the AR images illustrate that our algorithms are robust to partial occlusion.

Since face alignment and face acquisition (i.e., face detection or face tracking) are interrelated, in the following chapter we combine our joint alignment method with a face tracker for accurate face extraction in videos.

Chapitre 6

Adaptive Appearance Face Tracking with Alignment Feedbacks

6.1 Introduction

In a video-based face recognition system, simply applying face detection in each frame is not stable. Usually, face detectors are used to find the initial position of a face, then this target face is continuously located by the tracker across different frames. Since face alignment and face tracking are interrelated, we trace the source of mis-alignments (i.e., face tracking errors) rather than limiting our work in the face alignment stage.

Numerous approaches have been proposed for visual tracking [127]. These algorithms are conventionally developed on object representation methods and prediction schemes. Typical representation methods include contours [57], view-based appearance models [17], integration of shape and color [14], mixture models [60], histograms [21], etc. Prediction schemes are usually based on optical flow [83], kernel-based filters [21], particle filters [57], support vector machines (SVM) [6], etc. Since most existing tracking methods use fixed appearance models of the target, they often fail in uncontrolled environments where objects undergo large variations in pose, scale and illumination.

To solve these problems, trackers have been combined with adaptive appearance models. Jepson et al. [60] employed three components to account for appearance changes during tracking. Incremental Visual Tracker (IVT) [98] represents objects in a low-dimensional incremental subspace with a mean update. Babenko et al. [8] proposed an online instance learning approach to deal with appearance ambiguity. Recently, Kwon et al. [68] tracked the target by sampling trackers. However, these approaches usually lack direct mechanisms for correcting spatial mis-alignments (e.g., translation, scaling and rotation errors) existing in their tracking outputs. The unwanted errors are then accumulated in the target's appearance model. This inevitably has negative effects on tracking performance, even leads to lose-target. The work of [65] introduced pose and alignment constraints to the target model, in which alignment confidences are determined by a two-class SVM classifier. Same as IVT, this method predicts the target's state using a fixed dynamic model which is sensitive to gross changes in target's state such as scale and speed.

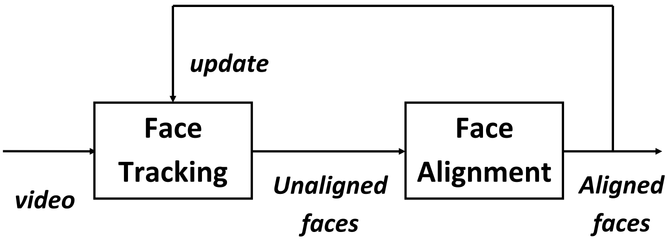


Figure 6.1 – Pipeline of the proposed algorithm

This chapter presents an adaptive appearance method for tracking faces in uncontrolled environments (the pipeline of our approach is shown in Figure 6.1). Section 6.2 first introduces an incremental update algorithm for target's appearance model. Section 6.3 discusses the details of our adaptive appearance face tracking method using alignment feedbacks. We first apply a self-adaptive dynamical model to predict target candidates in a particle filtering framework (Section 6.3.1). This improvement brings two main advantages : our tracker (1) is able to work with identical parameters for various situations, (2) is more robust to large changes in target's scale. In order to decrease the impact of mis-alignment, we then employ a multi-view joint face alignment method which performs well on low resolution, noisy video frames (Section 6.3.2). Aligned faces are further used as feedbacks to update the appearance model of target face (Section 6.3.3). In Section 6.4, we test the proposed algorithm on outdoor surveillance videos and real-world YouTube videos. Experimental results prove the effectiveness of the proposed algorithm in tracking faces undergoing large variations in pose, expression and scale. Compared to IVT, faces tracked by our algorithm are clearly better aligned.

6.2 Incremental update of appearance model

In a video sequence, the appearance of the target may change drastically due to both intrinsic (e.g., pose variation and shape deformation) and extrinsic (e.g., illumination change, camera viewpoint, and occlusion) factors. Therefore, it is necessary to update the appearance model online to produce a tracker which is robust to these changes. In this chapter, we choose an eigenbasis as appearance model. Typically, in order to learn an eigenbasis from a set of training images $\{\boldsymbol{I}_1, ..., \boldsymbol{I}_n\}$ (for convenience, let $\boldsymbol{I}_i$ be an N-dimensional image vector, where N is the number of pixels in each image), we first calculate a $N \times N$ covariance matrix :

$$C = \frac{1}{n-1} \sum_{i=1}^{n} (\boldsymbol{I}_i - \bar{\boldsymbol{I}})(\boldsymbol{I}_i - \bar{\boldsymbol{I}})^T \tag{6.1}$$

where $\bar{\boldsymbol{I}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{I}_i$ is the mean of training images.

Then, eigenvectors $U = [u_1 ... u_N]$ and eigenvalues $\{\lambda_1, ..., \lambda_N\}$ of the covariance matrix are computed by :

$$C\boldsymbol{u}_k = \lambda_k \boldsymbol{u}_k \tag{6.2}$$

In the eigenspace, an image $\boldsymbol{I}_i$ can be represented by :

$$\boldsymbol{Y}_i = U_r^T(\boldsymbol{I}_i - \bar{\boldsymbol{I}}) \tag{6.3}$$

where U_r consists of r eigenvectors corresponding to the largest eigenvalues.

Equivalently, this principal component analysis (PCA) procedure can be done by computing the singular value decomposition (SVD) of the centered data matrix. More precisely, we attain eigenvectors U using :

$$[(\boldsymbol{I}_1 - \bar{\boldsymbol{I}}), ..., (\boldsymbol{I}_n - \bar{\boldsymbol{I}})] = U\Sigma V^T \tag{6.4}$$

where Σ is an $N \times n$ diagonal matrix and V is an $n \times n$ matrix.

Updating the appearance model to deal with novel changes of the target can be taken as retraining the eigenbasis with an additional image set $\{\boldsymbol{I}_{n+1}, ..., \boldsymbol{I}_{n+m}\}$. Of course this update can be performed by computing the singular value decomposition $U'\Sigma'V'^T$ of the matrix $[(\boldsymbol{I}_1 - \bar{\boldsymbol{I}}'), ..., (\boldsymbol{I}_{n+m} - \bar{\boldsymbol{I}}')]$, where $\bar{\boldsymbol{I}}'$ is the average of all $n+m$ images. Unfortunately this procedure is not satisfactory for online visual tracking, due to its storage and computational requirements.

Numerous algorithms have been proposed to efficiently update an eigenbasis as more images arrive [41, 45, 73, 18, 46]. Here we employ an incremental PCA algorithm

1. Calculate the mean vectors $\bar{\boldsymbol{I}}_B = \frac{1}{m}\sum_{i=n+1}^{n+m} \boldsymbol{I}_i$, and $\bar{\boldsymbol{I}}_C = \frac{n}{n+m}\bar{\boldsymbol{I}}_A + \frac{m}{n+m}\bar{\boldsymbol{I}}_B$.

2. Form the matrix $\hat{B} = [(\boldsymbol{I}_{m+1} - \bar{\boldsymbol{I}}_B), ..., (\boldsymbol{I}_{n+m} - \bar{\boldsymbol{I}}_B), \sqrt{\frac{nm}{n+m}}(\bar{\boldsymbol{I}}_B - \bar{\boldsymbol{I}}_A)]$.

3. Compute $\tilde{B} = orth(\hat{B} - UU^T\hat{B})$ and $R = \begin{bmatrix} \Sigma & U^T\hat{B} \\ 0 & \tilde{B}(\hat{B} - UU^T\hat{B}) \end{bmatrix}$, where orth() performs orthogonalization.

4. Compute the SVD of R : $R \overset{\text{SVD}}{=} \tilde{U}\tilde{\Sigma}\tilde{V}^T$.

5. Finally $U' = [U\tilde{B}]\tilde{U}$ and $\Sigma' = \tilde{\Sigma}$.

Figure 6.2 – The incremental PCA algorithm with mean update [98].

with mean update [98] which is developed based on the Sequential Karhunen-Loeve (SKL) algorithm [73]. Let $A = \{\boldsymbol{I}_1, ..., \boldsymbol{I}_n\}$, $B = \{\boldsymbol{I}_{n+1}, ..., \boldsymbol{I}_{n+m}\}$ respectively be the existing data matrix and new data matrix and $C = [A\ B]$ be their concatenation. $\bar{\boldsymbol{I}}_A$, $\bar{\boldsymbol{I}}_B$, $\bar{\boldsymbol{I}}_C$ represent the mean vectors of A, B, C. Suppose that we have already computed the SVD : $(A - \bar{\boldsymbol{I}}_A) = U\Sigma V^T$, where the notation $(A - \bar{\boldsymbol{I}}_A)$ stands for the matrix $[(\boldsymbol{I}_1 - \bar{\boldsymbol{I}}_A), ..., (\boldsymbol{I}_n - \bar{\boldsymbol{I}}_A)]$. Our goal is to compute the SVD : $(C - \bar{\boldsymbol{I}}_C) = U'\Sigma' V'^T$. The main steps of the employed update algorithm are listed in Figure 6.2.

6.2.1 Forgetting factor

In a time-varying scene, the appearance of a target is more related to recently-acquired images than to earlier observations. Hence, a *forgetting factor* [73, 98] can be used to down-weight the contribution of earlier images during the tracking procedure. To do this, at each update the previous singular values are multiplied by a scalar factor $f \in [0, 1]$, where $f = 1$ means that no forgetting is to occur. In order to use a forgetting factor, two main modifications are made to the algorithm in Figure 6.2 : (1) In Step 1, the mean update becomes $\bar{\boldsymbol{I}}_C = \frac{fn}{fn+m}\bar{\boldsymbol{I}}_A + \frac{m}{fn+m}\bar{\boldsymbol{I}}_B$; (2) In Step 3, the calculation of R changes to $R = \left[\begin{smallmatrix} f\Sigma & U^T\hat{B} \\ 0 & \tilde{B}(\hat{B} - UU^T\hat{B}) \end{smallmatrix}\right]$.

6.3 Face tracking with alignment feedbacks

In a typical face recognition system, face alignment is the following stage of face acquisition (e.g., detection, tracking). Normally, face alignment and face acquisition are run in an open-loop manner i.e., there is no feedback to restrict the output of face acquisition methods. For an adaptive appearance method, spatial mis-alignments (e.g., translation, scaling and rotation errors) existing in faces will be accumulated in the target's appearance model. This inevitably has negative effects on tracking performance, even leads to lose-target. Here we present a closed-loop solution to adaptive appearance face tracking where aligned faces are used as feedbacks to update the appearance model of target. Section 6.3.1 introduces a sequential inference model for tracking where a self-adaptive dynamical model is applied to predict target candidates. Section 6.3.2 discusses a multi-view joint face alignment method. Aligned faces are further used as feedbacks to update the appearance model of target face (see Section 6.3.3).

6.3.1 Sequential inference model for tracking

Based on Markov process and Bayesian recursion, the tracking problem can be formulated as an inference task :

$$p(\boldsymbol{u}_t|\mathbb{F}_t) \propto \ p(\boldsymbol{F}_t|\boldsymbol{u}_t) \int p(\boldsymbol{u}_t|\boldsymbol{u}_{t-1})p(\boldsymbol{u}_{t-1}|\mathbb{F}_{t-1})d\boldsymbol{u}_{t-1} \tag{6.5}$$

in which $\boldsymbol{F}_t$, $\boldsymbol{u}_t$ are respectively the video frame and target state at time t, and $\mathbb{F}_t = \{\boldsymbol{F}_1, ..., \boldsymbol{F}_t\}$. Suppose that the initial state $\boldsymbol{u}_0$ is known, the aim is to estimate the hidden state variable $\boldsymbol{u}_t$. Here we use similarity transformation parameters $\boldsymbol{u}_t = [x_t, y_t, s_t, \theta_t]$, where (x_t, y_t) is the center position of tracking box, s_t, θ_t are respectively scale and rotation angle.

Two major elements in Equation (6.5) are the observation model $p(\boldsymbol{F}_t|\boldsymbol{u}_t)$ and the dynamical model $p(\boldsymbol{u}_t|\boldsymbol{u}_{t-1})$. In a particle filter framework, candidate particles predicted by $p(\boldsymbol{u}_t|\boldsymbol{u}_{t-1})$ are weighted according to $p(\boldsymbol{F}_t|\boldsymbol{u}_t)$.

(1) Self-adaptive dynamic model

Be different from [98] where the dynamic model is relied on fixed variances, we introduce a self-adaptive dynamical model based on a Gaussian distribution :

$$p(\boldsymbol{u}_t|\boldsymbol{u}_{t-1}) = \mathcal{N}(\boldsymbol{u}_t; \boldsymbol{u}_{t-1}, \boldsymbol{\Psi_t}) \tag{6.6}$$

where $\boldsymbol{\Psi_t}$ is a diagonal covariance matrix defined by :

$$\boldsymbol{\Psi_t} = \varphi(\boldsymbol{u}_{t-1}) \tag{6.7}$$

Using this dynamical model, the prediction of particles is adaptive to target's recent state. Hence, our tracker is able to work with identical parameters for all situations, and it is more robust to large changes in target's state. A simple and reliable example of Equation (6.7) is to adjust translation variances regarding target's scale :

$$\boldsymbol{\Psi_t} = diag(s_{t-1} \cdot \psi_x, s_{t-1} \cdot \psi_y, \psi_s, \psi_\theta) \tag{6.8}$$

in which ψ_x, ψ_y, ψ_s, ψ_θ stand for constant values.

(2) Observation model

The cropped face image $\boldsymbol{I}_t$ can be obtained by (see an illustration in Figure 6.3) :

$$\boldsymbol{I}_t = w(\boldsymbol{F}_t, \boldsymbol{u}_t) \tag{6.9}$$

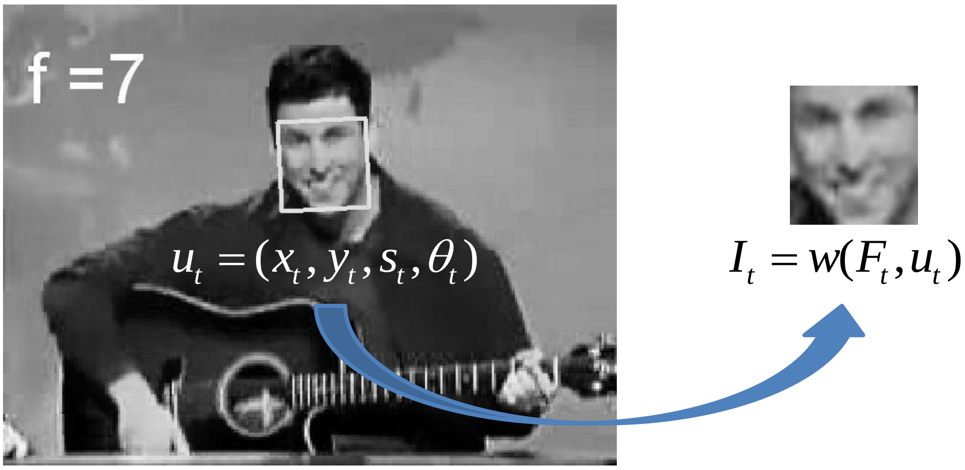


FIGURE 6.3 – Crop image $\boldsymbol{I}_t$ from video frame $\boldsymbol{F}_t$ using state parameters $\boldsymbol{u}_t$.

Similar to [98], we build a low-dimensional subspace to represent the target :

$$\boldsymbol{M}(\boldsymbol{I}_1, ..., \boldsymbol{I}_{t-1}) = (\boldsymbol{\mu}, \boldsymbol{U}) \tag{6.10}$$

where $\boldsymbol{\mu}$, $\boldsymbol{U}$ respectively denote the mean and the basis of subspace.

The likelihood of $\boldsymbol{I}_t$ being generated from this target model is defined as :

$$p(\boldsymbol{F}_t | \boldsymbol{u}_t) = \mathcal{N}(\boldsymbol{I}_t; \boldsymbol{\mu}, \boldsymbol{U}\boldsymbol{U}^T) \tag{6.11}$$

According to [99], the likelihood in Equation (6.11) is proportional to the negative exponential distance from $\boldsymbol{I}_t$ to the subspace, i.e., we can use the following equation to estimate the likelihood :

$$p(\boldsymbol{F}_t|\boldsymbol{u}_t) \propto exp(-\left\|(\boldsymbol{I}_t - \boldsymbol{\mu}) - \boldsymbol{U}\boldsymbol{U}^T(\boldsymbol{I}_t - \boldsymbol{\mu})\right\|^2) \tag{6.12}$$

6.3.2 Multi-view face alignment

This section discusses (1) the basic knowledge of the employed joint image alignment method, (2) a solution to align new images efficiently, and (3) the estimation of face poses.

(1) Joint image alignment

Since video images are usually with low resolution, blur and noise, here we employ the joint face alignment approach, referred to as "Lucas-Kanade entropy congealing", in which the alignment problem is defined as a minimization problem of a sum-of-entropy function. First, POEM descriptor [119] is used to represent faces. The transformation parameters are then iteratively estimated using a Newton optimization method (see details in Chapter 4). Also, we assume the type of geometric mis-alignment is a similarity transformation.

(2) Alignment of new images

As shown in Figure 6.1, tracked faces are output to the alignment stage. If we simply insert new tracked faces into the image set and repeat the joint alignment procedure on all images, it will be very computationally intensive. Inspired by [53], we first selected hundreds of face images from different videos to form a training set. Then, transformation parameters related to the training images are recorded at each iteration of the alignment procedure. To align new face images, we insert them into the training set and "re-run" the alignment algorithm. Actually, by using the saved transformation parameters, there is no more computational cost for the training images. The main steps of this alignment process are shown in Figure 6.4.

All face regions are expanded by 1.4 before alignment. There are two reasons for expanding the face region : (1) some facial features may be outside the original tracking box (e.g., bad crops), and expanding allows including these facial features into the new face region ; (2) alignment often causes outlier (black) pixels at the border of images, cropping the expanded margins after alignment will decrease the impact of outlier pixels.

(3) Multi-view model

Considering the large changes in target's pose, we build a multi-view model for face alignment. More precisely, we detect about 3,500 facial images of different poses from Youtube videos [65] to estimate a PCA-based pose subspace model : $\boldsymbol{M}_i =$

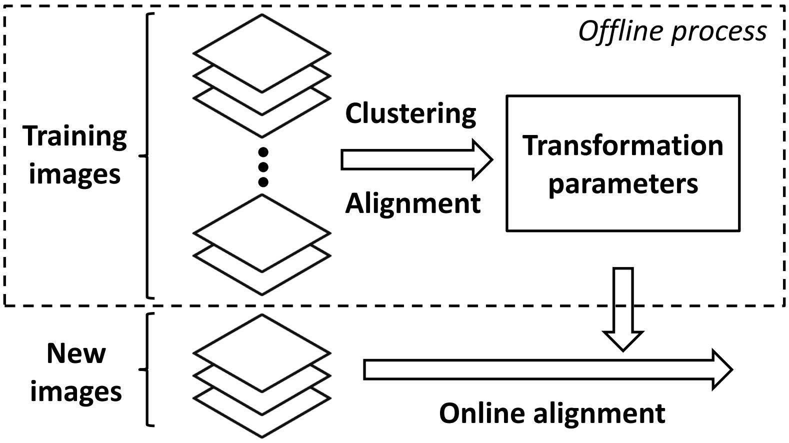


FIGURE 6.4 – Overview of alignment process.

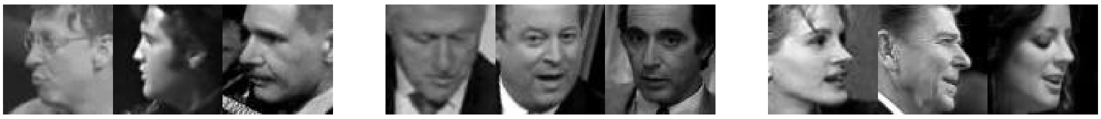
$(\boldsymbol{\mu}_i, \boldsymbol{U}_i)$, where i means a pose. Since our alignment method performs well on frontal and near-frontal faces, here we roughly categorize the poses into three clusters : near-frontal, left and right profile (see some examples in Figure 6.5). The pose of a new face image $\boldsymbol{I}_t$ can be estimated by :

$$pose = arg\ min_i d(\boldsymbol{I}_t, (\boldsymbol{\mu}_i, \boldsymbol{U}_i)) \tag{6.13}$$

where the distance is calculated using :

$$d(\boldsymbol{I}_t, (\boldsymbol{\mu}_i, \boldsymbol{U}_i)) = \left\|(\boldsymbol{I}_t - \boldsymbol{\mu}_i) - \boldsymbol{U}_i\boldsymbol{U}_i^T(\boldsymbol{I}_t - \boldsymbol{\mu}_i)\right\|^2 \tag{6.14}$$

For each pose cluster, we can form a training set and record their transformation parameters during joint alignment (the offline process in 6.4). After pose estimation, new faces are aligned according to the related training set.



(a) Left profile (b) Near-frontal (c) Right profile

FIGURE 6.5 – Examples of faces across different poses.

6.3.3 Appearance model updating

Let $\boldsymbol{v}_t$ represent the vector of the transformation parameters for $\boldsymbol{F}_t$ estimated by the alignment stage, the final cropped face image $\boldsymbol{I}_t^{'}$ is calculated by :

$$\boldsymbol{I}_t^{'} = w(\boldsymbol{F}_t, \boldsymbol{u}_t \circ \boldsymbol{v}_t) \tag{6.15}$$

where "$\circ$" stands for the composition of two warps.

Then, the aligned faces $\left\{\boldsymbol{I}_{t1}^{'}, ..., \boldsymbol{I}_{t2}^{'}\right\}$ are used as feedbacks to update the appearance model of target defined in Equation 6.10 every n frames ($n = t_2 - t_1 + 1$). Here, we incrementally update eigenbasis and mean of target model using the solution proposed in [98] (see details in Section 6.2).

6.4 Experimental results

We test the proposed method against Incremental Visual Tracker (IVT) [1] [98] on surveillance videos (Section 6.4.1) and real-world YouTube videos (Section 6.4.2). The frame sizes range from (180 × 240) to (240 × 320). Manually marked target states in the first frames were used as the initial states for tracking. For target representation, tracked faces were resized to 48×48, and the appearance models were updated every 10 frames. For simplicity, here we only track one face in the video. The tracking of multiple targets can be achieved by using multiple appearance templates. Our algorithm implemented in Matlab is able to process 2-3 frames per second with 600 particles.

6.4.1 Tracking on surveillance videos

We first tested the proposed method on surveillance videos [2] which were collected in uncontrolled outdoor environments to mimic the real-world conditions. In these videos, people walk from far to near, accompanying with large variations in the face's scale and translation speed.

Both IVT and the proposed tracker were run on these surveillance videos, and some example results are shown in Figures 6.6 to 6.8. It is clear that our algorithm is able to stably track both objects while IVT turned to lose-target (object 1 in Figure 6.6 (a)) or bad-crop (object 2 in Figure 6.7 (b)). This is because IVT uses a fixed dynamic model to predict target's position, whereas our algorithm is based on a self-adaptive dynamic model which is more robust to the changes in target's state.

[1] The code of IVT can be found at `http://www.cs.toronto.edu/~dross/ivt/`

[2] Collected by the laboratory LASMEA in France.

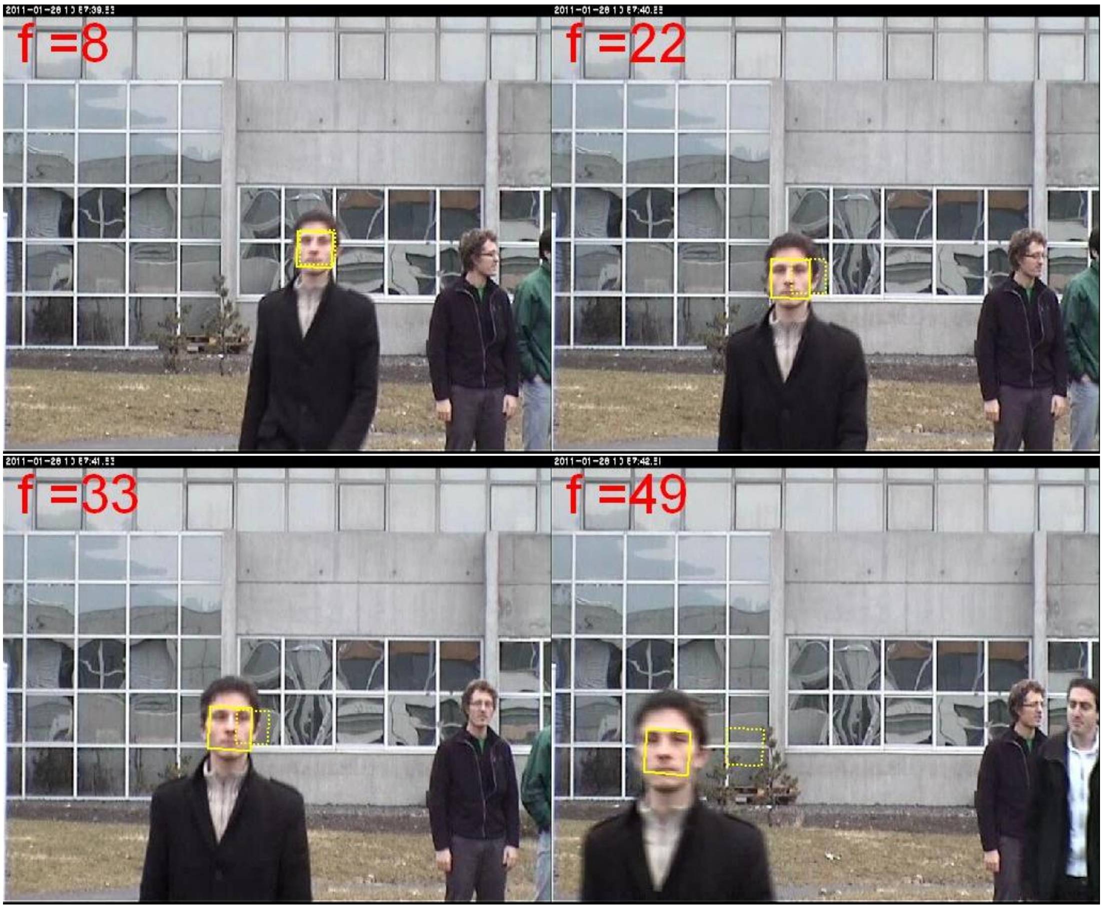


FIGURE 6.6 – Tracking results of IVT (dotted line) and our method (solid line) on surveillance videos : Object 1.

According to the target states estimated by our tracker, we can obtain a set of cropped faces using Equation (6.15). Similarly, the cropped faces of IVT can be acquired using Equation (6.9). The tracked faces of object 3 in Figure 6.8 cropped from frame 20 and 33 are shown in Figure 6.9. In these frames, both IVT and the proposed approach are able to track the face region, but the cropped faces contain clear spatial mis-alignments while faces generated by our method are much better aligned.

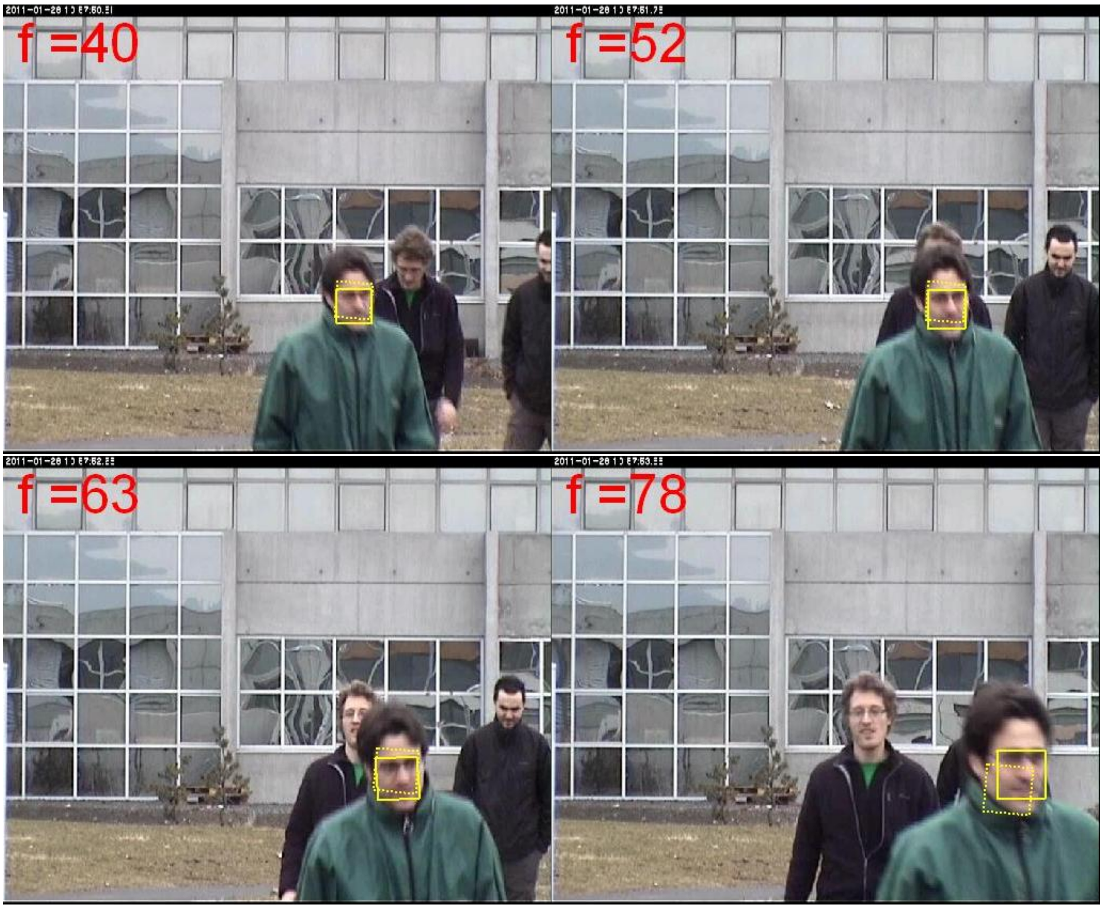


FIGURE 6.7 – Tracking results of IVT (dotted line) and our method (solid line) on surveillance videos : Object 2.

6.4.2 Tracking on YouTube videos

Our algorithm was also evaluated on a large set of noisy real-world videos containing 1910 video clips of 47 celebrities from YouTube [65]. We used identical parameters for all these data, and our approach successfully tracked faces in 85% of these video clips. Here, we consider lose-target and obviously bad crops as failed cases. Figure 6.10 shows several examples of tracking faces with different poses. We can see that there exist many mis-alignments in the outputs of IVT. More precisely, the faces tracked by IVT have large variations in poses (e.g., in Figure 6.10 (c), from frame 61 to 85), and they are often bad cropped (e.g., in Figure 6.10 (b), frame 78). Compared to IVT, the proposed approach is much more stable.

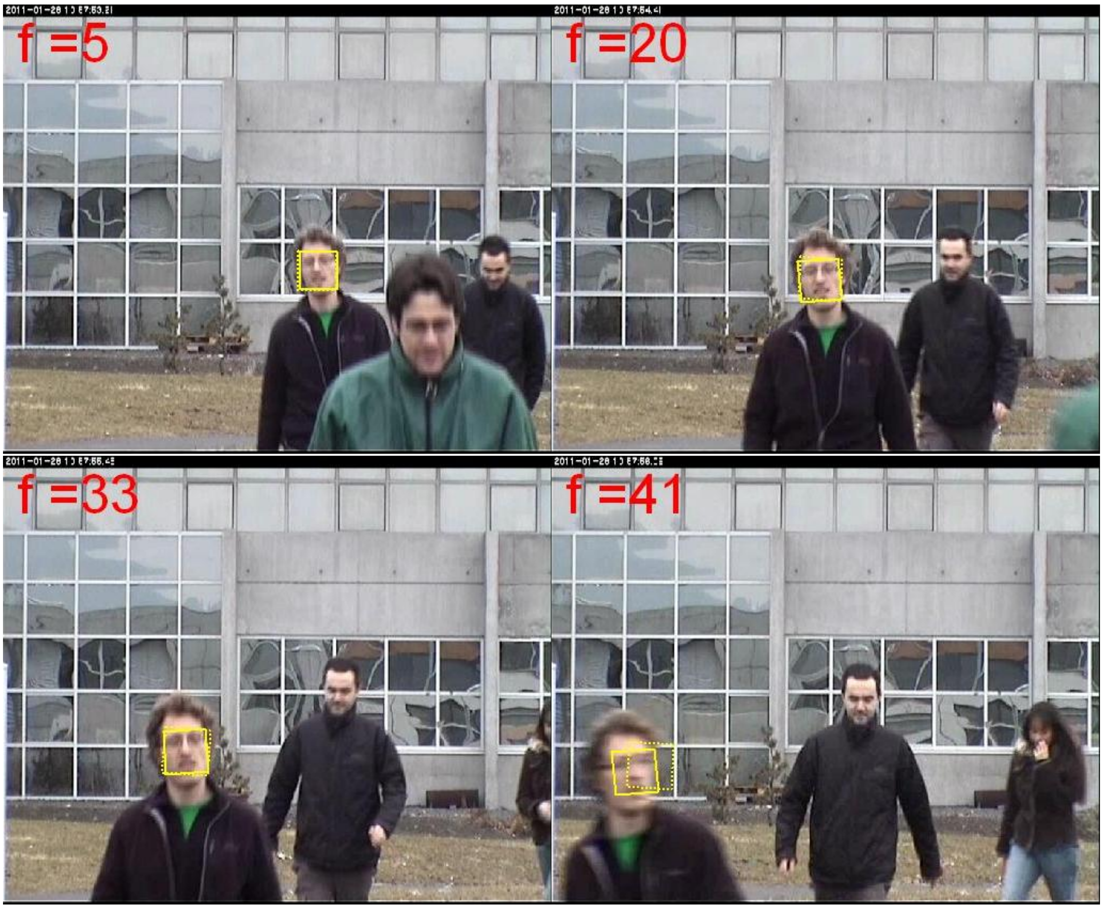


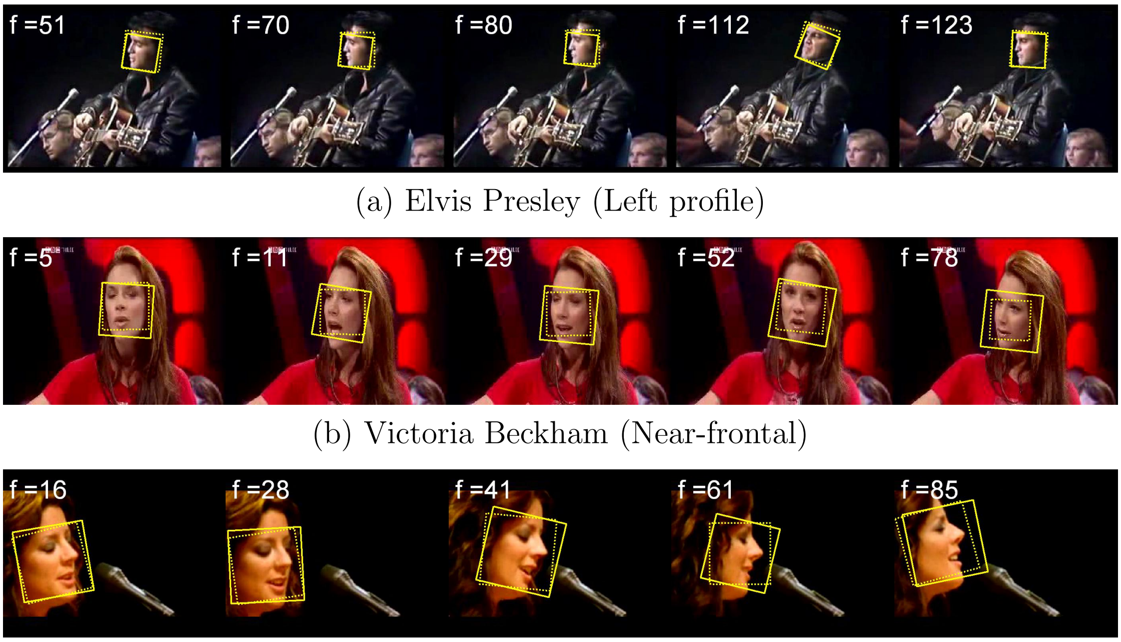
FIGURE 6.8 – Tracking results of IVT (dotted line) and our method (solid line) on surveillance videos : Object 3.

(a) frame 20

(b) frame 33

FIGURE 6.9 – Cropped faces of object 3. For each sub-figure, the result of IVT is on the left, and the result of our method is on the right.

For better evaluation, we respectively calculated the average images of cropped faces produced by IVT and our tracker. Since well-aligned faces are in a standard



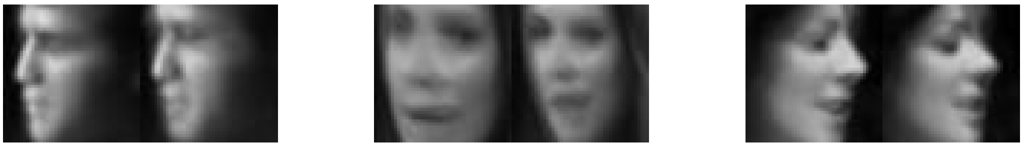
(a) Elvis Presley (Left profile)

(b) Victoria Beckham (Near-frontal)

(c) Sarah Mclachlan (Right profile)

FIGURE 6.10 – Tracking results of IVT (dotted line) and our method (solid line) on YouTube videos with different poses.

pose, their average image should have clearer facial features. Figure 6.11 presents the average face images related to the same videos as in Figure 6.10. It is clear that the average images of our algorithm have clearer facial features (e.g., eyes, mouth) than those of IVT.



(a) Left profile

(b) Near-frontal

(c) Right profile

FIGURE 6.11 – Average images of cropped faces. For each sub-figure, the result of IVT is on the left, and the result of our method is on the right.

Besides these visual evaluations, we also respectively calculated the convergence rates of facial landmarks in two sets of cropped faces. Five facial landmarks (eyes, nose tip and mouth corners) were manually labeled in all the frames of "Victoria

Beckham" video. New positions of landmarks in cropped faces were computed using the target's states. We can easily obtain the average positions of landmarks over a set of face images. If the distance between a landmark and its relevant average position is smaller than the threshold, the landmark is taken as converged. The convergence rates are shown in Figure 6.12, the size of cropped faces is 48×48. We can find that our algorithm significantly improves the convergence rates over all threshold values.

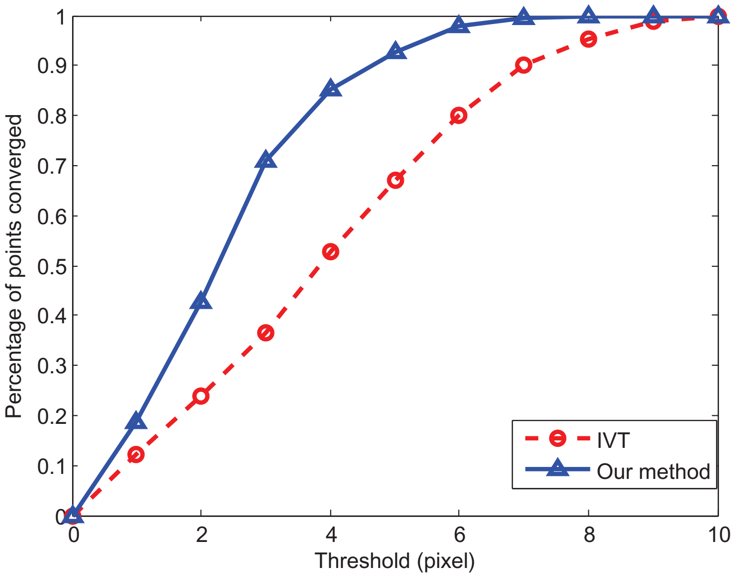


FIGURE 6.12 – Convergence rates of landmarks in cropped faces.

Real-world face tracking often encounters occlusions of target. Figure 6.13 is an example of face tracking with temporary occlusions. It is clear that our algorithm performs well in this situation and the tracked faces are better aligned than those of IVT (e.g., frame 61 and 63).

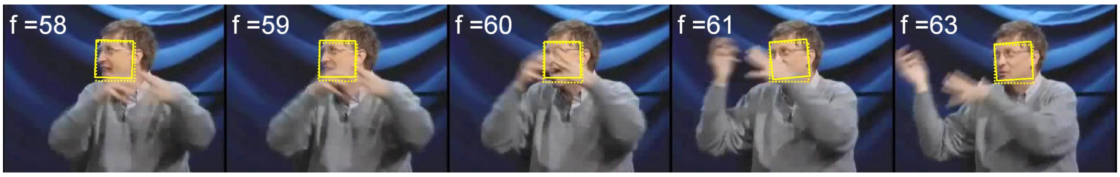


FIGURE 6.13 – Tracking results of IVT (dotted line) and our method (solid line) on YouTube videos with temporary occlusions.

6.5 Conclusion

This chapter describes an adaptive appearance method for tracking faces in uncontrolled environments. We first apply a self-adaptive dynamical model to predict target candidates in a particle filtering framework. This makes our tracker able to work with identical parameters for different situations, and be more robust to large changes in target's state. In order to decrease the impact of mis-alignment, we then employ a multi-view joint face alignment method named "Lucas-Kanade entropy congealing" which performs well on low resolution, noisy video frames. Instead of re-running the congealing process on all images, we propose a solution to align new faces efficiently based on the transformation parameters recorded in the training phase. Aligned faces are further used as feedbacks to update the appearance model of target. We tested the proposed algorithm on outdoor surveillance videos and real-world YouTube videos. Experimental results prove the effectiveness of the proposed algorithm in tracking faces undergoing large variations in pose, expression and scale. Compared to IVT, faces tracked by our algorithm are clearly better aligned. This indicates that the use of alignment feedback is able to dramatically improve the performance of visual tracker.

Since face alignment and face tracking are interrelated, our work is not limited in the face alignment stage. Here we use aligned faces as feedbacks to update the face tracker, i.e., our method runs in a closed-loop manner. A key idea of this strategy is that we trace the source of mis-alignment errors rather than just passively aligning images.

Chapitre 7

Conclusions and Perspectives

7.1 Conclusions

This thesis studies face alignment algorithms which are capable of working in the context of video surveillance. We address our work as a part of an automatic face recognition system. The main challenging factors of this research include the low quality of images (e.g., low resolution, motion blur, and noise), uncontrolled illumination conditions, pose variations, expression changes, and occlusions. In order to deal with these problems, we propose several face alignment methods using different strategies. Our methods are evaluated on image sets which are selected from different databases under different conditions for mimicing the real-world video context. We conclude our main contributions as follows :

• *Feature point detection :* We first present a three-stage method to locate facial feature points in Chapter 3. While existing algorithms mostly rely on a priori knowledge of facial structure and on a training phase, our approach works in an online mode without requirements of pre-defined constraints on feature distributions. Instead of training specific detectors for each facial feature, a generic method is used to extract a set of interest points from test images in the first step. Then, using POEM descriptor, a smaller set of these points are picked as candidates. In the final step, we apply a game-theoretic technique to select facial points from the candidates. The experimental results prove that (1) our method achieves satisfactory performance on AR face images under expression and lighting variations, (2) the use of game-theoretic technique is able to preserve the global geometric of face, and (3) POEM descriptor is efficient for facial feature representation.
• *Joint alignment with entropy :* A key contribution of this work is the development

of an unsupervised joint face alignment algorithm, referred to as “Lucas-Kanade entropy congealing” (LKC), where an image ensemble is aligned by minimizing a sum-of-entropy function defined over all images. As discussed in Chapter 4, we solve this minimization problem using both forward and inverse Lucas-Kanade formulations. Unlike the canonical entropy congealing which estimates transformation parameters sequentially, our LKC based algorithms are able to estimate all the transformation parameters at the same time. In the comparison experiments on images of different illumination conditions and qualities, our algorithms outperform other considered joint alignment methods regarding both alignment performance and computational speed. Moreover, using images aligned by LKC-based algorithms is able to significantly improve the accuracy of generic face recognition methods.

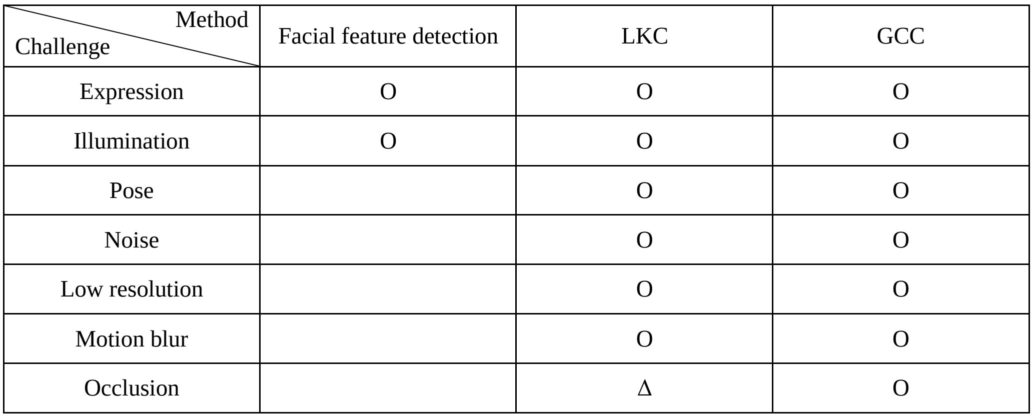
• *Joint alignment with gradient correlation coefficient :* In Chapter 5, we also propose another efficient unsupervised joint face alignment algorithm named “gradient correlation congealing” (GCC) which uses gradient correlation coefficient as similarity measure. While most existing face alignment methods suffer from outliers, e.g., occlusions, GCC is able to align faces undergoing partial occlusions. Moreover, our algorithm can cope with non-uniform illumination changes (even extremely difficult ones from the Yale B database).

• *Combination of face tracking and face alignment :* Our work is not limited in the face alignment stage. Since face alignment and face acquisition are interrelated, we developed an adaptive appearance face tracking method with alignment feedbacks in Chapter 6. We first apply a self-adaptive dynamical model to predict target candidates in a particle filtering framework. Hence, our tracker is able to work with identical parameters for various situations and it is robust to large changes in target’s state. Then, we employ a multi-view joint face alignment phase based on LKC. Aligned faces are further used as feedbacks to update the appearance model of target. This significantly decreases the mis-alignment errors in tracked faces.

In order to show the abilities of our algorithms clearly, we respectively list the names of the three algorithms and the main challenges for face alignment in Figure 7.1. The notation “O” means that one method has certain robustness to the corresponding challenge. For the LKC based algorithms, there is no specific design to cope with occlusion, but they are still able to achieve acceptable performance. Hence, we use the notation “Δ” for this case.

7.2 Perspectives

In this thesis, we mainly investigate joint alignment algorithms for a set of images. In order to expand the applications of our work, it is interesting to find a solution



Method Challenge	Facial feature detection	LKC	GCC
Expression	O	O	O
Illumination	O	O	O
Pose		O	O
Noise		O	O
Low resolution		O	O
Motion blur		O	O
Occlusion		Δ	O

FIGURE 7.1 – The abilities of our face alignment methods regarding different challenging factors. "O" means that one method has certain robustness to the corresponding challenge. "Δ" stands for that one method is able to achieve acceptable performance under the corresponding challenge.

to align a single image. Of course, we can insert the new image into the set, then re-run the joint alignment process. Unfortunately, this is time consuming and not suitable for video surveillance application. One possible solution has been presented in Chapter 6, we align a small set of images using transformation parameters recorded in the training stage with less computational cost. But this strategy still does not work well on separate images. Hence, the alignment of a single image using joint algorithms can be further investigated.

Face analysis methods including face alignment suffer from expression and pose variation in uncontrolled environments. Chapter 6 roughly divides faces into three pose clusters using a PCA based subspace. To improve the performance of face alignment, more sophisticated algorithms, e.g., Kernel Entropy Component Analysis [59] and a multi-view face detector [62], can be used to estimate the pose and expression of faces. Similar to [79], another possible extension is to combine the proposed method with a clustering function. In this way, pose/expression estimation and face alignment can be achieved simultaneously.

There is also a notable problem of face alignment regarding the face recognition application. In general, face alignment methods first estimate geometric transformations which correct the mis-alignment errors in faces. Then, aligned images are produced by using an interpolation method with respect to the estimated transformations. If the faces to be processed are of low quality (with low resolution and noise), the interpolation procedure will introduce undesired noises into the aligned images. In this case, even if faces are “well” aligned, face recognition accuracy will not be improved due to the image differences caused by new noises. One possible solution to this problem is using super-resolution algorithms to improve the quality of images. Otherwise, we may try to avoid directly applying interpolation methods to low quality images. For example, if gallery images are of high quality whereas probe images are of low quality, we can apply inverse transformations to gallery images, then compare these warped gallery images with unaligned probe images.

This work focuses on the alignment of facial images. Actually, our alignment methods (LKC and GCC) can be also used in other applications. For example, we can align MR (magnetic resonance) images of brains for medical applications, or align images of cars in a traffic control system.

Chapitre 8

Résumé en Français

8.1 Introduction

Dans un système typique de reconnaissance automatique de visage, la détection de visage et la reconnaissance de visage sont les deux étapes principales. Les algorithmes de détection de visage visent à localiser les régions du visage dans les images d'entrée. Les méthodes de reconnaissance de visage identifient ensuite les visages détectés en les comparant avec les images de visages de référence selon certains critères. Au cours de la dernière décennie, les chercheurs ont proposé de nombreuses méthodes de détection et de reconnaissance de visage. Malheureusement, les détecteurs de visage ne sont toujours pas exempts d'erreurs, autrement dit, il y a des erreurs d'alignement dans les visages détectés, telles que des erreurs de translation, d'échelle et de rotation. Plusieurs études ont démontré que ces erreurs d'alignement conduisent inévitablement à une dégradation des performances des méthodes de reconnaissance de visage. En effet, toutes ces méthodes procèdent à base de comparaison d'images qui n'a de sens que si les différents visages à comparer sont alignés.

Une solution possible au non-alignement est de développer des méthodes de reconnaissance de visage robustes aux erreurs d'alignement en faisant intervenir des invariants à ces erreurs ou en essayant de modéliser les erreurs d'alignement. Cependant, ces approches nécessitent un nombre important de données d'apprentissage, et elles ne peuvent pas éliminer complètement l'effet du non-alignement.

Une autre solution au non-alignement est d'introduire l'alignement de visage comme une étape de traitement supplémentaire, intermédiaire entre la détection et la reconnaissance de visage. L'objectif de l'alignement de visage est de transformer les visages détectés en une pose standard. Plusieurs recherches ont montré que l'application de méthodes d'alignement de visage peut améliorer considérablement la précision de l'étape de reconnaissance de visage.

Par voie de conséquence, nous avons choisi de travailler sur le développement d'algorithmes d'alignement de visage pour traiter le problème du non-alignement dans un système de reconnaissance de personnes.

8.1.1 Impact du non-alignement sur la reconnaissance de visage

En vision artificielle, la reconnaissance de visage peut être considérée comme un problème de classification ou de mise en correspondance de caractéristiques d'images de visage. Pratiquement, un visage inconnu (dénommé visage à reconnaître) est comparé avec tous les visages enregistrés dans une base de données (visages de référence) et il sera affecté à l'identité (classe) conduisant à la mesure de similarité la plus élevée. De toute évidence, un non-alignement spatial engendre des différences artificielles entre les visages à comparer. Et si les différences engendrées par le non-alignement prévalent sur les différences entre deux individus, il en résultera des erreurs de classification. Au cours des dernières années, quelques algorithmes efficaces de reconnaissance de visage ont été proposés sur la base de l'extraction de caractéristiques pertinentes du visage (par exemple, Local Binary Pattern (LBP), Histogram of Oriented Gradients (HOG), et Patterns of Oriented Edge Magnitudes (POEM)). Malgré tout , l'effet du non-alignement est encore loin d'être négligeable.

8.1.2 Les principaux défis pour l'alignement de visage

Ce travail se focalise sur l'alignement de visage en conditions non contrôlées d'acquisition des images de visage. En particulier, nous sommes très intéressés par la reconnaissance de visage en cas d'applications de vidéosurveillance. Dans ce contexte, nous avons à faire face aux problèmes suivants :

- Variations de la pose du sujet ;
- Occultation du visage ;
- Variations des conditions d'éclairement ;
- Variations de l'expression ;
- Faible résolution ;
- Flou ;
- Bruit engendré par le capteur.

8.1.3 Contributions principales

Le travail de cette thèse porte sur l'alignement de visage pour des applications de vidéosurveillance. Spécifiquement, nous proposons les contributions suivantes :

(1) La première contribution importante de ce travail est le développement d'un algorithme non supervisé d'alignement conjoint de visage, dénommé "Lucas-Kanade entropy congealing"(LKC), où une image entière est alignée en minimisant une fonction d'entropie définie sur toutes les images de la base d'apprentissage. Nous résolvons ce problème de minimisation en utilisant à la fois les formules directes et inverses de Lucas-Kanade. Contrairement au congealing canonique de l'entropie qui estime séquentiellement la transformation des paramètres, nos algorithmes LKC sont en mesure d'estimer tous les paramètres de transformation en même temps. D'ailleurs, dans la méthode d'alignement conjoint, il n'est pas nécessaire d'avoir des modèles prédéfinis.

(2) Nous proposons également un autre algorithme efficace non supervisé d'alignement conjoint de visage dénommé "Gradient Correlation Congealing" (GCC), qui utilise le coefficient de corrélation de gradient comme mesure de similarité. Alors que la plupart des méthodes existantes d'alignement de visage souffrent de valeurs aberrantes, le GCC permet d'aligner les visages qui subissent des occultations partielles. De plus, notre algorithme peut s'affranchir des changements non-uniformes

d'illumination (même ceux très difficiles).

(3) Dans le but d'aligner des images comportant des erreurs importantes de non-alignement, nous proposons une solution multi-résolution à l'alignement conjoint de visage : les images sont d'abord traitées à basses résolutions pour supprimer les erreurs importantes d'alignement ; ensuite, l'alignement est affiné avec les résolutions plus hautes.

(4) Comme l'alignement et l'acquisition de visage sont liés, nous développons une méthode adaptative de suivi du visage avec les retours d'alignement. Nous appliquons d'abord un modèle dynamique d'auto-adaptation pour envisager les cibles candidates dans un cadre de filtrage à particules. Notre tracker est capable de fonctionner avec des paramètres identiques pour les diverses situations. Afin de diminuer l'impact du non-alignement, nous mettons en place une phase d'alignement de visage basée sur le LKC. Les visages alignés sont ensuite utilisés comme retour pour mettre à jour le modèle d'apparence du visage cible.

(5) En dehors de ces algorithmes d'alignement conjoint, une méthode en trois étapes est proposée pour la localisation de points caractéristiques sur le visage. Alors que les algorithmes existants s'appuient principalement sur les connaissances acquises de la structure du visage et sur une phase d'apprentissage, notre approche fonctionne dans un mode en ligne sans contraintes prédéfinies sur la distribution des points recherchés. Au lieu de développer un détecteur spécifique pour chaque trait du visage, une méthode générique est d'abord utilisée pour extraire un ensemble de points d'intérêt des images de test. En utilisant l'histogramme des POEM, un plus petit nombre de ces points sont repris en tant que candidats. Ensuite, nous appliquons la théorie des jeux pour sélectionner les points du visage parmi les candidats, tout en préservant les propriétés géométriques globales du visage.

8.2 Classification des méthodes d'alignement de visage

Nous divisons les algorithmes d'alignement de visage en trois catégories, à savoir, ceux basés sur l'extraction de points caractéristiques, ceux qui procèdent par alignement direct et enfin ceux qui procèdent par alignement conjoint.

8.2.1 Alignement de visage basé sur l'extraction de points caractéristiques

Un alignement parfait de visages est généralement issu d'une phase d'alignement basée sur des repères extraits et mis en correspondant manuellement, par exemple, les centres des yeux, les narines et les coins de la bouche. Par conséquent, une approche intuitive au non-alignement est de détecter des points caractéristiques du visage et de les transformer en position standard. La détection de points caractéristiques du visage peut être divisée en deux catégories : les méthodes basées sur la texture et celles basées sur la forme. Les approches à base de texture modélisent la texture locale autour d'un point donné du visage, par exemple, les valeurs de luminance dans une petite zone autour du centre d'un œil. Les méthodes à base forme considèrent tous les repères du visage comme un modèle de forme, qui est développé à partir des visages marqués, et essaient de trouver la bonne forme pour des visages inconnus.

8.2.2 Alignement direct de visage

Les approches d'alignement direct de visage visent à déformer une image test de visage pour faire la paire avec une image prédéfinie. Les éléments importants de l'alignement direct de visage comprennent une mesure de similarité et une méthode d'optimisation. Plus précisément, l'alignement de visage est réalisé par optimisation d'une fonction de coût basée sur la similarité entre les deux images considérées.

8.2.3 Alignement conjoint de visage

Contrairement aux approches à base de points caractéristiques ou aux approches directes, les méthodes d'alignement conjoint de visages fonctionnent en alignant simultanément une série d'images faciales. Les méthodes récentes d'alignement conjoint comprennent le Learned-Miller Congealing (LMC) et le Least Square Congealing (LSC). Dans le cas du congealing, la seule hypothèse est le type de non-alignement géométrique, et les paramètres de transformation sont obtenus en minimisant une fonction de coût.

8.2.4 Discussion

Dans cette thèse, nous ne travaillons pas sur les méthodes à base de forme car elles sont trop compliquées et coûteuses en calcul pour des applications de vidéosurveillance. Comme les méthodes d'alignement direct de visage ne conviennent pas pour des applications de reconnaissance de visage en raison de l'exigence d'un modèle prédéfini, nous n'adoptons pas ce type de méthodes dans notre travail. Ainsi, dans cette thèse, nous nous concentrons sur l'étude de la détection de point à base de texture et sur l'alignement conjoint de visage.

8.3 Une méthode en trois étapes pour la localisation de points caractéristiques du visage

Trouver des points caractéristiques sur un visage expressif ou non uniformément éclairé est toujours un défi. Une solution populaire pour améliorer les performances de localisation est d'utiliser la relation spatiale entre les positions des différents traits du visage. Les algorithmes existants comptent essentiellement sur une connaissance acquise de la structure du visage et sur une phase d'apprentissage. Cependant, la construction des données d'apprentissage pour ces méthodes est compliquée et coûteuse en calcul.

Pour contourner l'étape d'apprentissage, nous proposons une approche en ligne sans exigences prédéfinies sur la distribution des points caractéristiques à extraire. Au lieu d'appliquer un détecteur spécifique pour chaque trait du visage, une méthode générique est utilisée pour extraire un groupe de points d'intérêt à partir des images de test. Avec un descripteur robuste de traits dénommé l'histogramme Patterns Oriented Edge Magnitude (POEM), le nombre de points candidats est réduit. Ensuite, nous appliquons la théorie des jeux pour sélectionner les points de visage parmi les candidats, tout en conservant les propriétés géométriques globales du visage.

8.3.1 Étape 1 : Détection des points d'intérêt

Contrairement à certaines approches nécessitant des détecteurs calibrés pour les traits spécifiques du visage, nous utilisons une méthode générique pour extraire un ensemble de points d'intérêt qui sont invariants à l'échelle, à la rotation et à la translation et qui sont également robustes aux changements d'éclairage. Un sous-ensemble de ces points est sélectionné en tant que candidats dans l'étape suivante. L'idée fondamentale derrière est que nous croyons que certains traits du visage, tels les coins des yeux, les coins de la bouche et les narines sont invariants aux transformations de similarité relativement aux changements d'identité, d'expression et d'éclairage. Nous avons testé les détecteurs de points d'intérêt suivants : Laplacian-of-Gaussian (LoG), Difference-of-Gaussian (DoG), Hessian-Laplacian et Harris-Laplacian. Après plusieurs séries de tests, nous adoptons ici le détecteur Harris-Laplacien, car il permet trouver plus de points caractéristiques du visage, tels que les coins de la bouche, les coins des yeux et les narines.

8.3.2 Étape 2 : Cribalge de points candidats

Après l'extraction de points d'intérêt, la localisation des points de visage se transforme en un problème de correspondance entre les points cibles du modèle et les points d'intérêt de l'image de test. Compte tenu de l'efficacité de la reconnaissance, pour chaque point cible, seulement K (par exemple, K ¡= 10) points dans l'image de test avec le descripteur le plus proche sont pris comme candidats. Un descripteur robuste est nécessaire pour distinguer les points d'intérêt. Nous proposons ici d'utiliser le descripteur POEM. En fonction de la distance entre les histogrammes, les K points d'intérêt associés au descripteur le plus proche sont pris en tant que candidats pour chaque point cible.

8.3.3 Etape 3 : Adaptation multi-modèle de la théorie des jeux

Jusqu'ici, il y a plusieurs points candidats dans l'image de test pour chaque point cible dans le modèle. A ce stade, nous cherchons à trouver les paires correspondantes pour chaque point cible. Puisque les traits de visage ont une certaine structure géométrique, il existe une transformation compatible pour toutes ces paires. Le processus de sélection peut être considéré comme un jeu d'adaptation, les paires de points avec des probabilités élevées sont pris comme paires correspondantes.

Comme les traits de visage des images de test varient en fonction du changement d'identité et d'expression, la question d'adaptation souffrirait de l'erreur de criblage des candidats. Plus précisément, la correspondance d'un point cible peut ne pas être incluse dans le groupe de candidats. Dans ce cas, toutes les paires contenant ce point cible obtiendront une faible probabilité après le jeu d'adaptation, et donc ce point de visage est mal localisé. Pour augmenter la robustesse de l'adaptation de la théorie des jeux, nous appliquons plusieurs modèles pour correspondre aux images de test. C'est seulement si l'un de ces modèles donne un point de correspondance au point cible que ce point de visage peut être localisé avec succès. Par conséquent, la probabilité de "mauvaise localisation" est très faible. Si un point de visage est localisé par plusieurs modèles, la localisation moyenne est utilisée comme résultat final.

8.3.4 Résumé

Nous avons présenté d'abord une approche en ligne pour localiser les points caractéristiques du visage, qui ne nécessite pas de contraintes prédéfinies sur la distribution de traits. Nous modélisons le problème de localisation en un jeu d'adaptation qui préserve la cohérence géométrique globale des points du visage. Les résultats expérimentaux montrent que l'algorithme proposé permet d'obtenir des performances satisfaisantes sur les images AR qui présentent des varations d'expressions et d'illumination. Comme indiqué dans la section précédente, il existe d'autres facteurs difficiles pour l'alignement de visage en contexte de vidéosurveillance : l'occultation, la pose, le bruit, le flou, et la basse résolution. Malheureusement, les méthodes à base de point ne sont pas performantes pour les images de vidéosurveillance de mauvaise qualité, parce que les méthodes à base de point dépendent des caractéristiques locales qui sont sensibles au bruit et au flou. En outre, le problème des occultations et des variations de pose sont encore des facteurs difficiles pour les détecteurs à base de texture. Dans cette perspective, le travail suivant est focalisé sur les méthodes d'alignement conjoint qui sont plus adaptées aux images de faible qualité.

8.4 Lucas-Kanade Entropy Congealing pour l'alignement conjoint de visage

Le congealing à base d'entropie a d'abord été proposé par Learned-Miller pour l'alignement conjoint d'images binaires et d'images de résonance magnétique. Cette méthode d'alignement conjoint a ses propres avantages : elle est automatique et insensible à la qualité d'image. Toutefois, la minimisation de la fonction d'entropie est basée sur une estimation de recherche séquentielle avec une taille d'incrément prédéfinie pour la mise à jour. Ceci entraîne généralement une faible vitesse de convergence. Afin de surmonter ces limitations, nous proposons l'utilisation de l'algorithme d'optimisation de Lucas-Kanade pour estimer en même temps tous les paramètres de transformation, plutôt que de façon séquentielle comme pour le congealing de Learned-Miller.

8.4.1 Lucas-Kanade entropy congealing

Dans cette méthode, une image est alignée via la minimisation d'une fonction d'entropie. Ici, nous combinons le congealing avec le descripteur facial robuste POEM. Cependant, il est difficile de calculer directement les probabilités du descripteur POEM. Par conséquent, tous les descripteurs de caractéristiques sont modélisés comme étant générés par une combinaison de modèles gaussiens en utilisant l'algorithme des k-moyennes. De cette manière, chaque pixel est représenté par un vecteur de probabilités qui peut être utilisé pour calculer la valeur d'entropie parmi les images.

Au lieu d'utiliser la méthode canonique de congealing, nous employons l'algorithme de Lucas-Kanade pour optimiser la fonction de coût. Plus précisément, nous calculons d'abord la matrice Jacobienne et Hessienne de la fonction de coût. Ensuite, on peut facilement obtenir un incrément de transformation qui est utilisé pour mettre à jour de manière itérative la transformation courante. Dans ce travail, nous avons respectivement développé deux formules de congealing, c'est-à-dire, les méthodes directe et inverse, qui se distinguent par le rôle du modèle. Lorsque le "champ de la distribution" est invariant au cours d'une itération, il est pris pour modèle dans la formule directe. Cependant, il existe un énorme coût de calcul en recalculant la matrice Jacobienne et Hessienne à chaque itération. Certaines études font remarquer que généralement la clé pour l'efficacité est d'inverser les rôles du modèle et des données de test. Par conséquent, nous inversons les rôles du modèle et de l'image de test. Cette formule de composition inverse permet de pré-calculer les matrices Jacobienne et Hessienne ce qui diminue la complexité de calcul. Nous menons une étude comparative de performances sur cinq groupes d'image des bases de données AR, SCface, FERET, LFW et vidéos de surveillance respectivement. Les avantages de nos méthodes ont été évalués à l'aide de différents protocoles : comparaison visuelle à l'image moyenne, estimation du taux de convergence, taux de reconnaissance de visage et complexité. Les résultats expérimentaux indiquent que le LKC proposé montre de meilleures performances que d'autres méthodes d'alignement, et qu'il a une certaine robustesse aux variations d'éclairage et de qualité d'image. En ce qui concerne la complexité, le LKC inverse est plus efficace que d'autres approches considérées. Par rapport à la formule directe, la méthode inverse produit une amélioration de la vitesse de 20%.

8.4.2 Multi-resolution Lucas Kanade entropy congealing

La plupart des méthodes existantes transforment le problème d'alignement conjoint en un problème de minimisation d'une fonction de coût calculée sur toutes les images, et estiment ensuite les paramètres de transformation en trouvant le minimum local de cette fonction. Malheureusement, les erreurs importantes d'alignement de visages conduisent souvent à une convergence de la fonction de coût dans un minimum local car la fonction de coût est a priori non convexe. Le processus d'optimisation peut se coincer dans un minimum local, ce qui signifie que les visages ne sont pas correctement alignés.

L'algorithme LKC proposé souffre également du problème de minimum local. Pour y remédier, nous proposons une solution multi-résolution pour laquelle les non-alignements les plus importants sont éliminés dans les hauts niveaux (images de résolution inférieure), et l'alignement est affiné dans des niveaux plus fins.

Les résultats expérimentaux sur les images des bases de données AR et Yale B prouvent que l'algorithme proposé est (1) robuste aux grandes erreurs de non-alignement, (2) plus efficace que les méthodes de résolution unique au niveau du coût de calcul (convergence plus rapide).

8.4.3 Résumé

Jusqu'ici, nos méthodes à base de LKC sont en mesure de traiter la plupart des cas difficiles pour l'alignement de visage en contexte de vidéosurveillance. Plus précisément, nos méthodes sont capables de traiter une certaine gamme de variation de pose et changements d'expression, parce qu'ils fonctionnent bien sur les images non contrôlées de la base de données LFW. Les résultats sur les images des bases de données AR et Yale B prouvent la robustesse des méthodes LKC aux variations d'éclairage. Les expérimentations sur les images de la base de données SCface et de vidéosurveillance montrent que nos méthodes sont capables d'aligner les images de mauvaise qualité où les visages ont une mauvaise résolution, sont flous ou bruités. Autrement dit, la seule difficulté restante est le cas d'occultation partielle des visages. Or les méthodes à base de LKC requièrent d'un processus de clustering pour réduire la dimension des vecteurs de caractéristiques. Lors de l'alignement d'images avec occultations, les pixels aberrants provenant de régions cachées ont un impact négatif sur la précision du clustering ce qui conduit à une baisse de performance de l'alignement. Ainsi, dans le travail suivant, nous nous concentrons sur la recherche d'une solution pour aligner les visages partiellement occultés.

8.5 Alignement conjoint de visage par coefficient de corrélation de gradient

En cas d'occultation partielle des visages, les détecteurs de visage ne fonctionnent pas très bien dans ces conditions difficiles, ce qui rend la phase d'alignement de visage plus nécessaire encore.

Dans ce travail, nous proposons un cadre d'alignement conjoint non supervisé, appelé "Gradient Correlation Congealing (GCC)", qui aligne un ensemble d'images en maximisant la somme des coefficients de corrélation de gradient définie sur toutes les images. Deux formulations différentes (GCC-1 et GCC-2) sont respectivement développées en fonction du rôle du modèle. Plus précisément, le GCC-1 utilise le reste des images comme modèle au sein d'une itération, tandis que le GCC-2 utilise l'image sélectionnée comme modèle. Les avantages principaux des méthodes à base du GCC sont : (1) elles travaillent de manière non supervisée, (2) elles ne nécessitent aucun modèle prédéfini, (3) elles sont robustes à la fois aux occultations et aux variations non-uniformes d'éclairage.

8.5.1 Gradient Correlation Congealing

La caractéristique que nous utilisons dans ce travail est l'orientation du gradient car il a été prouvé que c'est une caractéristique robuste à l'éclairage pour l'analyse de visage. Le travail de Tzimiropoulos et al. a utilisé une corrélation d'orientation du gradient à base de FFT pour l'enregistrement d'images. Récemment, ce coefficient de corrélation du gradient a été appliqué pour l'alignement de deux images faciales avec occultation. Mais cette approche nécessite des modèles prédéfinis, et les résultats expérimentaux sont présentés à l'aide de modèles relatifs aux mêmes sujets que les images de test, autrement dit, les identités des images de test sont déjà connues. Cette démarche ne semble pas logique en reconnaissance de visage, puisque l'objectif final de la reconnaissance de visage est justement d'identifier les visages à l'entrée du système. Pour contourner cette étape de sélection de modèle, nous créons une méthode d'alignement conjoint non supervisée appelée "Gradient Correlation Congealing (GCC)". Puisque nous utilisons le coefficient de corrélation comme fonction de coût pour laquelle une valeur élevée signifie que les images sont plus corrélées, le problème d'alignement conjoint est défini par la maximisation d'une fonction de similarité qui est calculée sur un ensemble d'images.

Intuitivement, il y a deux façons pour formuler la fonction de coût en fonction de la définition du modèle. Plus précisément, puisque les méthodes d'alignement conjoint ne nécessitent pas de modèle prédéfini, à chaque fois nous pouvons choisir une image (ou des images) pour modèle(s) et effectuer une transformation pour le reste. Pendant le processus d'alignement, le choix du modèle change tout le temps. Pour établir une distinction claire, les deux formulations différentes sont respectivement nommées GCC-1 et GCC-2.

Dans l'algorithme GCC-1, nous essayons de trouver une déformation pour l'image sélectionnée, le reste des images étant prises comme modèle. Cependant, cette formulation ne fonctionne pas correctement lorsque les images subissent de grandes erreurs d'alignement. Plus précisément, l'estimation des paramètres de transformation dans la méthode GCC-1 a un effet similaire à aligner une image sur la moyenne de l'ensemble des images, puisque le reste des images est invariant dans une itération donnée. Des non-alignements plus importants engendrent une image moyenne plus floue et aligner une image à cette image moyenne plus floue entraîne un alignement de moins bonne qualité.

Pour remédier à cette limitation, nous proposons alors l'algorithme GCC-2. Contrairement au GCC-1, la solution alternative est d'utiliser l'image sélectionnée comme modèle. Dans ce travail, nous utilisons une méthode d'optimisation similaire pour les deux formulations : une mise à jour des paramètres de transformation est obtenue en calculant la moyenne des déformations estimées entre l'image sélectionnée et chaque image du reste de l'ensemble.

Les algorithmes proposés sont testés et comparés à quatre méthodes typiques d'alignement conjoint : LSC, LMC, LKC et RASL et ce sur trois groupes d'images qui sont respectivement tirées des bases de données AR, Yale B et des vidéos de surveillance. Les résultats expérimentaux prouvent l'efficacité des approches proposées d'alignement dans les conditions différentes. En particulier, lorsque les visages sont partiellement cachés, nos algorithmes sont bien plus performants que les autres méthodes considérées. En comparaison avec le GCC-1, le GCC-2 est plus robuste aux grandes erreurs d'alignement.

8.5.2 Résumé

Les méthodes proposées (LKC et GCC-2) ont montré une certaine robustesse face à l'ensemble des facteurs de défi. Plus précisément, nos méthodes sont capables de traiter la basse résolution, le flou et le bruit ainsi qu'une certaine gamme de variations de pose et de changements d'expression. Les résultats sur les images Yale B prouvent la robustesse des méthodes LKC et GCC-2 aux variations d'éclairage (même les cas très difficiles). Les expérimentations sur les images AR montrent que nos algorithmes sont robustes aux occultations partielles.

8.6 Tracking adaptatif de visage avec boucle de retour d'alignement

Dans un système de reconnaissance de visage à base de la vidéo, la détection de visage dans chaque image n'est pas stable. Habituellement, les détecteurs de visage sont utilisés pour trouver la position initiale d'un visage, ce visage cible est ensuite localisé en permanence par suivi à travers les images successives. Comme l'alignement et le suivi de visage sont interdépendants, nous proposons de coupler un système de suivi et d'alignement de visages.

De nombreuses approches ont été proposées pour le suivi visuel. Ces algorithmes sont conventionnellement développés suivant les méthodes de représentation des objets et les systèmes de prédiction. La plupart des méthodes existantes de suivi utilisent des modèles figés d'apparence de la cible, d'où des erreurs en environnements non contrôlés où les objets subissent de grandes variations d'apparence (pose, échelle et éclairage). Pour résoudre ces problèmes, les trackers ont été combinés avec des modèles adaptatifs d'apparence. Cependant, ces approches manquent souvent de mécanisme direct pour corriger les non-alignements spatiaux qui peuvent survenir lors du suivi. Les erreurs indésirables s'accumulent ensuite dans le modèle d'apparence de la cible. Cela a inévitablement des effets négatifs sur la performance du suivi et cela engendre une perte de la cible. Nous présentons ici une méthode adaptative d'apparence pour le suivi de visages en environnement non contrôlé. Nous appliquons d'abord un modèle dynamique d'auto-adaptation pour prédire les cibles candidates dans un cadre de filtrage de particules. Afin de diminuer l'impact du non-alignement, nous utilisons ensuite une méthode d'alignement conjoint de visage qui fonctionne bien en basse résolution. Les visages alignés sont d'ailleurs utilisés comme information pour mettre à jour le modèle d'apparence du visage cible.

8.6.1 Suivi du visage

Basé sur le processus Markov, le problème de suivi peut être formulé comme une tâche d'inférence. Les deux éléments majeurs de cette tâche sont le modèle d'observation et le modèle dynamique. Dans un cadre de filtre à particules, les particules candidates prédites par le modèle dynamique sont pondérées selon le modèle d'observation.

Ici, nous présentons un modèle dynamique d'auto-adaptation basé sur une distribution gaussienne dans lequel la prédiction des particules est adaptée à l'état récent de la cible. Par conséquent, notre tracker est capable de fonctionner avec des paramètres identiques pour toutes les situations, et il est aussi plus robuste aux grands changements d'état de la cible. Un exemple simple et fiable de notre modèle dynamique d'auto-adaptation consiste à ajuster les variations de translation en fonction de l'échelle de la cible.

Pour le modèle d'observation, nous construisons un sous-espace de faible dimension pour représenter la cible. La probabilité pour un candidat d'être généré à partir de ce modèle cible est définie par une fonction exponentielle négative de la distance.

8.6.2 Alignement de visage multi-vues

Comme les images de la vidéo sont souvent de faible résolution, floues et bruités, nous utilisons ici l'approche d'alignement conjoint de visage, LKC, dans lequel le problème d'alignement est défini comme un problème de minimisation d'une fonction d'entropie. Tout d'abord, le descripteur POEM est utilisé pour représenter les visages. Les paramètres de transformation sont ensuite estimés de façon itérative en utilisant une optimisation de Newton. De plus, nous supposons que le type de non-alignement géométrique est une transformation de similarité.

Durant le processus de suivi, les visages suivis sont envoyés en sortie pour la phase d'alignement. Si nous insérons simplement les nouveaux visages suivis dans l'ensemble d'images et répétons la procédure d'alignement conjoint sur toutes les images, le coût de calcul sera très important. Dans ce travail, nous avons d'abord sélectionné quelques centaines d'images de visage issues de différentes vidéos pour former un ensemble d'apprentissage. Ensuite, les paramètres de transformation liés aux images d'apprentissage sont enregistrés à chaque itération de la procédure d'alignement. Pour aligner les nouvelles images de visage, nous les insérons dans l'ensemble d'apprentissage et relançons l'algorithme d'alignement. En fait, en utilisant les paramètres de transformation enregistrés, il n'y a plus de surcoût de calcul pour les images d'apprentissage.

Compte tenu des changements importants dans la pose de la cible, nous construisons un modèle multi-vues pour l'alignement de visage. Plus précisément, nous détectons environ 3500 images faciales de différentes poses dans les vidéos sur Youtube afin d'estimer un modèle de sous-espace de poses à base de PCA. Etant donné que notre méthode d'alignement fonctionne bien sur les visages frontaux, nous classons ici de manière approximative les poses en trois catégories : frontal, profil gauche et profil droite.

Pour chaque catégorie de poses, nous pouvons former un ensemble d'apprentissage et enregistrer leurs paramètres de transformation au cours de l'alignement conjoint. Après l'estimation de la pose, les nouveaux visages sont alignés selon leur propre ensemble d'apprentissage.

8.6.3 Mise à jour du modèle d'apparence

Dans un système typique d'analyse de visage, l'alignement de visage est l'étape suivante au suivi de visage. Normalement, l'alignement et le suivi de visage s'enchaînent en boucle ouverte, sachant qu'il n'y a pas de retour pour restreindre la sortie des méthodes de suivi de visage. Pour une méthode adaptative d'apparence, les non-alignements spatiaux existants dans les visages seront accumulés dans le modèle d'apparence de la cible. Cela aura inévitablement des effets négatifs sur les performances de suivi. Nous présentons ici une solution en boucle fermée pour le suivi adaptatif de l'apparence de visage où les visages alignés sont utilisés comme retour pour mettre à jour le modèle d'apparence de la cible.

8.6.4 Résumé

Comme l'alignement et le suivi de visage sont liés entre eux, notre travail ne se limite pas à la phase d'alignement de visage. Nous utilisons ici les visages alignés en tant que retour pour mettre à jour le suivi de visage, c'est-à-dire, notre méthode s'exécute en boucle fermée. Nous avons testé l'algorithme proposé sur des vidéos de surveillance extérieure et sur des vidéos d'actualités sur YouTube. Les résultats expérimentaux démontrent l'efficacité de l'algorithme proposé pour le suivi de visages subissant de grandes variations de pose, d'expression et d'échelle. Par rapport à un tracker en boucle ouverte, les visages suivis par notre algorithme sont clairement mieux alignés. Cela indique que l'utilisation du retour d'alignement permet d'améliorer considérablement la performance du suivi visuel.

8.7 Conclusions et perspectives

8.7.1 Conclusions

Cette thèse s'intéresse au développement d'algorithmes d'alignement de visage capables de fonctionner en contexte de vidéosurveillance. Cette phase d'alignement s'intègre directement dans les systèmes de reconnaissance de personnes par analyse de visage. Nous proposons plusieurs méthodes d'alignement de visage en utilisant des stratégies différentes. Nos méthodes sont évaluées par des ensembles d'images qui sont sélectionnées à partir de différentes bases de données pour leurs conditions spécifiques d'acquisition proches d'une acquisition réelle. Nous concluons nos principales contributions comme suit :

Détection de points caractéristiques : Nous présentons d'abord une méthode en trois étapes pour localiser les points caractéristiques du visage. Alors que les algorithmes existants s'appuient principalement sur les connaissances acquises de la structure de visage et sur une phase d'apprentissage, notre approche fonctionne dans un mode en ligne sans contraintes prédéfinies sur la distribution des caractéristiques. Au lieu d'appliquer les détecteurs spécifiques pour chaque caractéristique de visage, une méthode générique est utilisée pour extraire un ensemble de points d'intérêt à partir d'images de test dans la première étape. Ensuite, en utilisant le descripteur POEM, un sous-ensemble de ces points est sélectionné comme candidats. Dans la dernière étape, nous appliquons une technique de la théorie des jeux pour sélectionner les points du visage parmi les points candidats. Les résultats expérimentaux montrent que (1) notre méthode permet d'obtenir une performance satisfaisante sur les images avec variations d'expression et d'éclairage, (2) l'utilisation de la technique de théorie des jeux est en mesure de préserver la géométrie globale du visage, et (3) le descripteur POEM est efficace pour la représentation des caractéristiques du visage.

L'alignement conjoint avec l'entropie : Une contribution essentielle de ce travail est le développement d'un algorithme non supervisé d'alignement conjoint de visage, dénommé "Lucas-Kanade entropy congealing" (LKC), dans lequel un ensemble d'images est aligné en minimisant une fonction de la somme des entropies définies sur toutes les images. Nous résolvons ce problème de minimisation en utilisant à la fois les formulations directe et inverse d'optimisation de Lucas-Kanade. Contrairement au congealing classique qui estime les paramètres de transformation de façon séquentielle, nos algorithmes basés sur LKC permettent d'estimer tous les paramètres de transformation en même temps. Dans les expérimentations comparatives sur les images de différentes conditions d'éclairage et de différentes qualités, nos algorithmes surclassent les autres méthodes d'alignement conjoint considérées au niveau de la performance d'alignement et de la vitesse de calcul. En outre, l'utilisation des images alignées par les algorithmes à base de LKC permet d'améliorer considérablement la précision des méthodes génériques de reconnaissance de visage.

L'alignement conjoint avec coefficient de corrélation du gradient : Nous proposons également un autre algorithme efficace non supervisé d'alignement conjoint de visage dénommé "gradient correlation congealing" (GCC), qui utilise le coefficient de corrélation du gradient comme niveau de similarité. Alors que la plupart des méthodes existantes d'alignement de visage souffrent de valeurs aberrantes en cas d'occultation, le GCC permet d'aligner les visages qui subissent les occulations partielles. De plus, notre algorithme peut traiter le cas des images avec des variations d'éclairage non-uniformes (même celles extrêmement difficiles de la base de données Yale B).

La combinaison du suivi et de l'alignement de visage : Notre travail ne se limite pas à la phase d'alignement de visage. Comme l'alignement et l'acquisition de visage sont interdépendants, nous avons développé une méthode adaptative de suivi de l'apparence de visage. Nous avons d'abord appliqué un modèle dynamique d'auto-adaptation pour prédire les candidats cibles dans un cadre du filtrage de particules. Ainsi, notre tracker est capable de fonctionner avec des paramètres identiques pour diverses situations, et il résiste aux grands changements dans l'état de la cible. Ensuite, nous appliquons une phase multi-vues d'alignement conjoint de visage à base du LKC. Les visages alignés sont ensuite utilisés comme retour pour mettre à jour le modèle d'apparence de la cible. Cela réduit considérablement les erreurs d'alignement dans les visages suivis.

8.7.2 Perspectives

Dans cette thèse, nous nous intéressons principalement à des algorithmes d'alignement conjoint pour une série d'images. Afin d'élargir les applications de notre travail, il est intéressant de trouver une solution pour aligner une seule image. Bien sûr, on peut insérer la nouvelle image dans l'ensemble, puis relancer le processus d'alignement conjoint. Malheureusement, cela prend du temps et ne convient pas pour les applications de vidéosurveillance. Une solution possible a été présentée dans notre travail, nous alignons une petite série d'images en utilisant les paramètres de transformation enregistrés pendant l'étape d'apprentissage avec un coût inférieur de calcul. Mais cette stratégie ne fonctionne pas vidéosurveillance bien sur les images séparées. Par conséquent, l'alignement d'une seule image en utilisant les algorithmes conjoints peut être davantage étudié.

Les méthodes d'analyse de visage, y compris l'alignement de visage, souffrent des variations d'expression et de pose dans les environnements non contrôlés. Nous divisons grossièrement les visages en trois catégories à l'aide d'un sous-espace à base de PCA. Pour améliorer la performance d'alignement de visage, des algorithmes plus sophistiqués, tels que Kernel Entropy Component Analysis peuvent être utilisés pour estimer la pose et l'expression des visages. Une autre extension possible est de combiner la méthode proposée avec une fonction de clustering. De cette façon, l'estimation de la pose / de l'expression et l'alignement de visage peuvent être réalisés simultanément.

En général, les méthodes d'alignement de visage estiment d'abord les transformations géométriques qui corrigent les erreurs de non-alignement dans les visages. Ensuite, les images alignées sont produites en utilisant une méthode d'interpolation par rapport aux transformations estimées. Si les visages à traiter sont de faible qualité, la procédure d'interpolation induira du bruit indésirable sur les images alignées.

Dans ce cas, même si les visages sont “ bien” alignés, la précision de reconnaissance de visage ne sera pas améliorée en raison des différences d’images engendrées par le nouveau bruit. Une solution possible à ce problème est l’utilisation d’algorithmes de super-résolution pour améliorer la qualité des images. Sinon, nous pouvons essayer d’éviter l’application directe des méthodes d’interpolation sur les images de faible qualité. Par exemple, si les images de référence sont de haute qualité alors que les images à reconnaître sont de faible qualité, nous pouvons appliquer les transformations inverses sur les images de référence, puis comparer ces images de référence déformées avec les images non alignées à reconnaître.

Bien que ce travail ait porté sur l’alignement d’images de visage, nos méthodes d’alignement (LKC et GCC) peuvent également être utilisées dans d’autres applications. Par exemple, nous pouvons aligner des images RM (résonance magnétique) du cerveau pour les applications médicales, ou aligner les images de voitures dans un système de contrôle de circulation.

List of Publications

International journals :
[1] W. Ni, N. Vu, and A. Caplier. Lucas-kanade based entropy congealing for joint face alignment. *Image and Vision Computing* (2012), `http://dx.doi.org/10.1016/j.imavis.2012.08.016`.

International conferences :
[1] W. Ni and A. Caplier. Appearance adaptive face tracking with alignment feedbacks. In *IEEE International Conference on Image Processing (ICIP)*, Orlando, USA, 2012.

[2] W. Ni and A. Caplier. Newton optimization based Congealing for facial image alignment. In *IEEE International Conference on Image Processing (ICIP)*, Brussels, Belgium, 2011.

[3] W. Ni, N. Vu, and A. Caplier. An online three-stage method for facial point localization. In *International Conference on Computer Analysis of Images and Patterns (CAIP)*, Seville, Spain, 2011.

Annexe A

Scale-Invariant Feature Transform (SIFT)

Scale-Invariant Feature Transform (SIFT), proposed by Lowe [82], is an algorithm for detecting and describing local features in images. SIFT aims at finding distinctive features which are invariant to scale, rotation, illumination, and viewpoint. The main steps of this algorithm include scale-space extrema detection, keypoint localization, orientation assignment, and descriptor building. In this report, our interest is to use SIFT descriptor to represent pixels or local regions in images, i.e., the last stage of canonical SIFT algorithm.

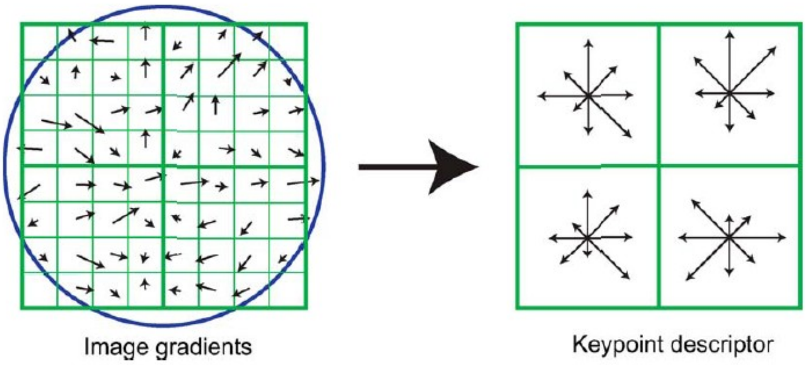


FIGURE A.1 – This figure shows a 2×2 SIFT descriptor computed from an 8×8 region [82].

The computation of SIFT descriptor is illustrated in Figure A.1. First the image gradient magnitudes and orientations are sampled around the target pixel. These gradients are shown with small arrows at each sample location on the left side of

Figure A.1. A Gaussian function (a circular window on the left side of Figure A.1) is used to weight the magnitude of each sample point. The use of this Gaussian window can avoid sudden changes in the descriptor with small changes in the position of the window, and to give smaller weights to points that are far from the center of the descriptor.

These gradients are then accumulated into orientation histograms summarizing the contents over subregions of a pre-defined size. The example on the right side of Figure A.1 is a 2×2 SIFT descriptor computed from an 8×8 region, i.e., the size of subregions is 4×4. In general, histograms have eight directions. For example, [82] uses 4×4 descriptors from 16×16 which leads to a 1×128 (4×4×8=128) vector.

Finally, the feature vector is normalized to unit length to improve the robustness to uniform illumination changes. To deal with non-linear illumination variations, large gradient magnitudes in unit feature vector are set to a pre-defined value (0.2 in [82]). After that, the vector is renormalized.

Annexe B

Local Binary Pattern (LBP)

The LBP operator was originally proposed by Ojala et al. [90] for representing the texture of an image. For each pixel in the image, the operator thresholds its eight neighbor of a fixed radius with the center value. All the neighbors will then have a value of 1 if their value is greater than or equal to the current pixel, and 0 if the value is lower (see Figure B.1). LBP code of the current pixel is then generated by concatenating the 8 values to form a binary code which is between 0 and 255. In this way, we can obtain a new gray level image containing LBP values of pixels.

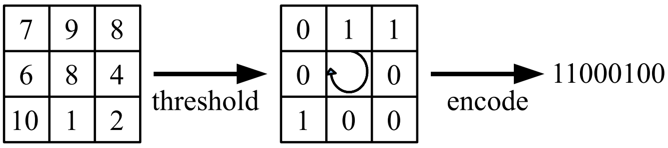
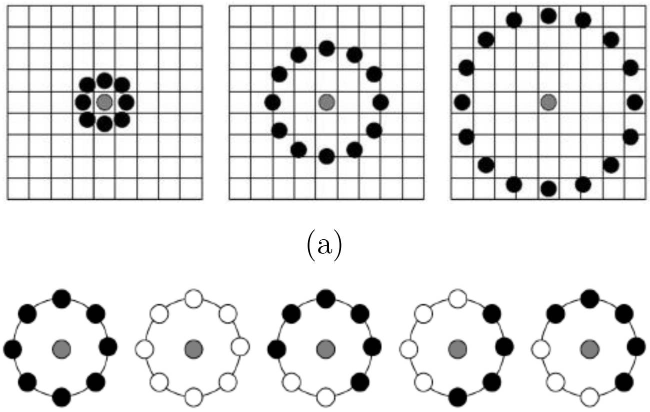


Figure B.1 – LBP operator [120].

The LBP was further extended by using neighborhoods of different sizes. In this case, a circle of radius R around the center pixel is considered. Values of P points on the edge of the circle are taken and compared with the value of central pixel. To obtain the values of points exactly locating on the circle of any radius R, an interpolation is necessary. We use the notation (P, R) to represent the neighborhood with P sampling points on a circle of radius of R. Figure B.2 (a) shows three neighborhoods for different values of R and P.

Let g_c be the gray value of central pixel, $g_p(p = 1, ..., p)$ be the gray levels of its neighbors, the LBP index of the current pixel is calculated as :



(a)

(b)

Figure B.2 – (a)Three neighborhoods for different values of R and P, (b) special textures detected by LBP^{u2}

$$LBP_{P,R}(x_c, y_c) = \sum_{p=1}^{P} s(g_p - g_c)2^{p-1} \tag{B.1}$$

where (x_c, y_c) are the coordinates of the current pixel, $LBP_{P,R}$ is the LBP code for the neighborhood (P, R), and function $s()$ is :

$$s(x) = \begin{cases} 1 & if \quad x \geq 0 \\ 0 & if \quad x < 0 \end{cases} \tag{B.2}$$

The LBP operator obtained with $P = 8$ and $R = 1$ ($LBP_{8,1}$) is very close to the original LBP operator. The main difference is that the pixels must first be interpolated to obtain values of points on the circle.

Another extension to the original operator is the uniform LBP. A LBP code is called uniform if it contains at most two bitwise transitions from 0 to 1 or vice versa when the binary string is considered circular. For example, 00000000, 00011110 and 10000011 are uniform codes. Using a uniform LBP code, noted LBP^{u2} has two advantages. The first is the save of computation time and memory. In the computation of the LBP labels, uniform patterns are used so that there is a separate label for each uniform pattern and all the non-uniform patterns are labeled with a single label. For example, when using $(8, R)$ neighborhood, there are a total of 256 patterns, 58 of which are uniform, which yields in 59 different labels. The second is that LBP^{u2}

detects only significant local textures, such as spots, end of line, edges, and corners (examples of these special textures are shown in Figure B.2 (b)). Indeed, Ojala et al. [90] showed that uniform LBPs contain over 90% of the information of an image.

An important property of LBP is that the code is invariant to global uniform changes of illumination, because the LBP of a pixel depends only on the differences between its gray level and those of its neighbors.

B.1 Face recognition with LBP

After obtaining the LBP codes for all pixels of a facial image, we calculate the histogram of the image LBP to form a feature vector representing the facial image.

In order to incorporate more spatial information into the descriptor, the facial image is divided into local regions and LBP texture descriptors are extracted from each region independently. The descriptors are then concatenated to form a global description of the face, as shown in Figure B.3.

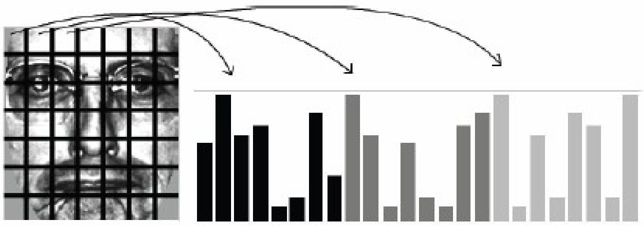


FIGURE B.3 – Using LBP histograms to represent a facial image.

Given two LBP histograms H_1, H_2 of two faces, the next step is to use a metric to calculate the similarity between these two histograms. By testing three metrics χ^2, *Histogram intersection*, and *Log-likelihood statistic*, Ahonen et al. [1] found that the first metric provides the best results :

$$\chi_2(H_1, H_2) = \sum_i \frac{(H_1^i - H_2^i)^2}{H_1^i + H_2^i} \tag{B.3}$$

where i is the number of a local region.

Acknowledgments

I would like to express my gratitude to my supervisor, Prof. Alice Caplier. Her expertise, understanding, and patience, helped me to proceed through the PhD study and complete my dissertation.

I would like to thank my co-supervisor, Dr. Annie Luciani, for the assistance she provided in many aspects of my thesis.

I would also like to acknowledge Prof. Gang Feng for his support and guidance when I was applying for this PhD position.

Special thanks to the other members of my committee, Prof. Catherine Achard, Prof. Renaud Séguier, Prof. Mohsen Ardabilian and Prof. Pierre-Yves Coulon. It is no easy task, reviewing a thesis, and I am grateful for their hard work.

My colleagues in the laboratory deserve my sincere thanks, it was an honor to work with you all. In particular, I would like to thank Dr. Ngoc-Son Vu for his kind assistance at all levels of research.

Last but not least, I would like to thank my parents and my girlfriend for the support they provided me through the last three years.

References

[1] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *European Conference on Computer Vision*, pages 469–481, 2004.

[2] A. Albarelli, E. Rodola, and A. Torsello. A game-theoretic approach to fine surface registration without initial motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[3] A. Albarelli, E. Rodolà, and A. Torsello. Loosely distinctive features for robust surface alignment. In *European Conference on Computer Vision*, 2010.

[4] A. Ashraf, S. Lucey, and T. Chen. Fast image alignment in the fourier domain. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2480–2487, 2010.

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

[6] S. Avidan. Support vector tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26 :1064–1072, 2004.

[7] B. Babenko, P. Dollár, Z. Tu, S. Belongie, et al. Simultaneous learning and alignment : Multi-instance and multi-pose learning. *Proc. Faces in Real-Life Images*, 2008.

[8] B. Babenko, M. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, 2009.

[9] S. Baker, R. Gross, I. Matthews, and T. Ishikawa. Lucas-kanade 20 years on : A unifying framework : Part 2. Technical Report CMU-RI-TR-03-01, Robotics Institute, Pittsburgh,PA, February 2003.

[10] S. Baker and I. Matthews. Lucas-kanade 20 years on : A unifying framework. *International Journal of Computer Vision*, 56(3) :221–255, 2004.

[11] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces : Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7) :711–720, 1997.

[12] T. Berg, A. Berg, J. Edwards, M. Maire, R. White, Y. Teh, E. Learned-Miller, and D. Forsyth. Names and faces in the news. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–848, 2004.

[13] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli. On the use of sift features for face authentication. In *Proc. Conference on Computer Vision and Pattern Recognition Workshop*, pages 35–42, 2006.

[14] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *CVPR*, 1998.

[15] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, and S. Nayar. Face swapping : automatically replacing faces in photographs. *ACM Transactions on Graphics (TOG)*, 27(3) :39, 2008.

[16] M. Black and P. Anandan. The robust estimation of multiple motions : Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, 63(1) :75–104, 1996.

[17] M. Black and A. Jepson. Eigentracking : Robust matching and tracking of articulated objects using a view-based representation. *Internationl Journal of Computer Vision*, 26 :63–84, 1998.

[18] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *Proceedings of the Seventh European Conference on Computer Vision*, 2002.

[19] H. Chen, P. Belhumeur, and D. Jacobs. In search of illumination invariants. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, SC, USA*, volume 1, pages 254–261, 2000.

[20] B. Chunyan Xie, S. Palanivel, and B. Yegnanarayana. A still-to-video face verification system using advanced correlation filters. In *Proc. first international conference Biometric authentication, Hong Kong, China*, page 102, 2004.

[21] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25 :564–577, 2003.

[22] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6) :681–685, 2001.

[23] T. Cootes and C. Taylor. On representing edge structure for model matching. In *IEEE Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA*, 2001.

[24] T. Cootes, C. Taylor, D. Cooper, J. Graham, et al. Active shape models-their training and application. *Computer vision and image understanding*, 61(1) :38–59, 1995.

[25] T. Cootes, C. Taylor, and A. Lanitis. Active shape models : Evaluation of a multi-resolution method for improving image search. In *British Machine Vision Conference*, York, UK, 1994.

[26] M. Cox, S. Sridharan, S. Lucey, and J. Cohn. Least squares congealing for unsupervised alignment of images. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, USA*, pages 1–8, 2008.

[27] M. Cox, S. Sridharan, S. Lucey, and J. Cohn. Least-squares congealing for large numbers of images. In *International Conference on Computer Vision, Kyoto, Japan*, 2009.

[28] D. Cristinacce and T. Cootes. Facial Feature Detection and Tracking with Automatic Template Selection. In *7th IEEE International Conference on Automatic Face and Gesture Recognition*, 2006.

[29] D. Cristinacce, T. Cootes, and I. Scott. A multi-stage approach to facial feature detection. In *15th British Machine Vision Conference*, pages 277–286, 2004.

[30] W. Crum, T. Hartkens, and D. Hill. Non-rigid image registration : theory and practice. *British journal of radiology*, 77(Special Issue 2) :S140, 2004.

[31] F. De la Torre and M. Nguyen. Parameterized kernel principal component analysis : Theory and applications to supervised and unsupervised image alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[32] L. Ding and A. Martinez. Precise detailed detection of faces and facial features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2008.

[33] R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof. Fast active appearance model search using canonical correlation analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10) :1690–1694, 2006.

[34] T. D'Orazio, M. Leo, G. Cicirelli, and A. Distante. An algorithm for real time eye detection in face images. In *Proc. 17th International Conference on Pattern Recognition*, volume 3, pages 278–281, 2004.

[35] N. Dowson and R. Bowden. Mutual information for lucas-kanade tracking (milk) : An inverse compositional formulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1) :180–185, 2008.

[36] G. Evangelidis and E. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10) :1858–1865, 2008.

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

[38] B. Frey and N. Jojic. Estimating mixture models of images and inferring spatial transformations using the em algorithm. In *IEEE Conference on Computer Vision and Pattern Recognition, Ft. Collins, CO, USA*, 1999.

[39] B. Frey and N. Jojic. Transformed component analysis : Joint estimation of spatial transformations and image components. In *Proc. International Conference on Computer Vision, Kerkyra, Greece*, volume 2, pages 1190–1196, 1999.

[40] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many : Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6) :643–660, 2001.

[41] G. Golub and C. Van Loan. *Matrix computations*, volume 3. Johns Hopkins University Press, 1996.

[42] M. Grgic, K. Delac, and S. Grgic. Scface- surveillance cameras face database. *Multimedia Tools and Applications*, 51(3) :863–879, 2011.

[43] R. Gross, L. Sweeney, F. de la Torre, and S. Baker. Semi-supervised learning of multi-factor models for face de-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[44] S. Gundimada and V. Asari. Face alignment and adaptive weight assignment for robust face recognition. *Advances in Visual Computing*, pages 191–198, 2005.

[45] P. Hall, D. Marshall, and R. Martin. Incremental eigenanalysis for classification. In *British Machine Vision Conference*, volume 1, pages 286–295. Citeseer, 1998.

[46] P. Hall, D. Marshall, and R. Martin. Adding and subtracting eigenspaces with eigenvalue decomposition and singular value decomposition. *Image and Vision Computing*, 20(13-14) :1009–1016, 2002.

[47] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50, 1988.

[48] M. Hasan, C. Pal, et al. Improving alignment of faces for recognition. In *Proc. IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, pages 249–254, 2011.

[49] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3) :328–340, 2005.

[50] D. Hill, P. Batchelor, M. Holden, and D. Hawkes. Medical image registration. *Physics in medicine and biology*, 46 :R1, 2001.

[51] D. Hond and L. Spacek. Distinctive descriptions for face processing. In *Proc. 8th British Machine Vision Conference*, volume 1, pages 320–329, 1997.

[52] C. Hu, R. Feris, and M. Turk. Active wavelet networks for face alignment. In *British Machine Vision Conference*, 2003.

[53] G. Huang, V. Jain, and E. Learned-Miller. Unsupervised joint alignment of complex images. In *11th International Conference on Computer Vision, Rio de Janeiro, Brazil*, 2007.

[54] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild : A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[55] J. Huang and H. Wechsler. Eye detection using optimal wavelet packets and radial basis functions (rbfs). *International Journal of Pattern Recognition and Artificial Intelligence*, 13 :1009–1026, 1999.

[56] Y. Huang, Q. Liu, and D. Metaxas. A component based deformable model for generalized face alignment. In *Proc. 11th International Conference on Computer Vision*, pages 1–8, 2007.

[57] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *4th ECCV*, 1996.

[58] R. Jenkins and A. Burton. 100% accuracy in automatic face recognition. *Science*, 319(5862) :435–435, 2008.

[59] R. Jenssen. Kernel entropy component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5) :847–860, 2010.

[60] A. Jepson, D. Fleet, and T. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25 :1296–1311, 2003.

[61] F. Jiao, S. Li, H. Shum, and D. Schuurmans. Face alignment using statistical models and wavelet features. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Madison, WI, USA*, volume 1, pages I–321, 2003.

[62] M. Jones and P. Viola. Fast multi-view face detection. *Mitsubishi Electric Research Lab TR-20003-96*, 2003.

[63] F. Kahraman, M. Gokmen, S. Darkner, and R. Larsen. An active illumination and appearance (aia) model for face alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[64] T. Kawaguchi, D. Hidaka, and M. Rizon. Detection of eyes from human faces by hough transform and separability filter. In *Proc. International Conference on Image Processing*, volume 1, pages 49–52, 2000.

[65] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *CVPR*, 2008.

[66] R. Kothari and J. Mitchell. Detection of eye locations in unconstrained visual images. In *Proc. International Conference on Image Processing*, volume 3, pages 519–522, 1996.

[67] C. Kuglin and D. Hines. The phase correlation image alignment method. In *Proc. IEEE Conf. Cybernetics and Soc.*, pages 163–165, 1975.

[68] J. Kwon and K. Lee. Tracking by sampling trackers. In *ICCV*, 2011.

[69] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3) :300–311, 1993.

[70] A. Lanitis, C. Taylor, and T. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7) :743–756, 1997.

[71] E. Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28 :236–250, 2006.

[72] A. Lemieux and M. Parizeau. Experiments on eigenfaces robustness. In *Proc. 16th International Conference on Pattern Recognition, Quebec, Canada*, volume 1, pages 421–424, 2002.

[73] A. Levey and M. Lindenbaum. Sequential karhunen-loeve basis extraction and its application to images. *IEEE Transactions on Image Processing*, 9(8) :1371–1374, 2000.

[74] K. Levi and Y. Weiss. Learning object detection from a small number of examples : the importance of good features. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Washington, USA*, volume 2, pages 53–60, 2004.

[75] Y. Li and W. Ito. Shape parameter optimization for adaboosted active shape model. In *Proc. Tenth IEEE International Conference on Computer Vision*, volume 1, pages 251–258, 2005.

[76] L. Liang, F. Wen, Y. Xu, X. Tang, and H. Shum. Accurate face alignment using shape constrained markov network. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1313–1319, 2006.

[77] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2) :79–116, 1998.

[78] X. Liu. Discriminative face alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11) :1941–1954, 2009.

[79] X. Liu, Y. Tong, and F. Wheeler. Simultaneous alignment and clustering for an image ensemble. In *Proc. International Conference on Computer Vision, Kyoto, Japan*, pages 1327–1334, 2009.

[80] X. Liu, Y. Tong, F. Wheeler, and P. Tu. Facial Contour Labeling via Congealing. In *European Conference on Computer Vision, Crete, Greece*, 2010.

[81] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, 1999.

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

[83] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International joint conference on artificial intelligence*, volume 3, pages 674–679. Citeseer, 1981.

[84] J. Luo, Y. Ma, E. Takikawa, S. Lao, M. Kawade, and B. Lu. Person-specific sift features for face recognition. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages II–593, 2007.

[85] H. B. Markus Storer, Martin Urschler. Intensity-based congealing for unsupervised joint image alignment. In *International Conference on Pattern Recognition, Istanbul, Turkey*, 2010.

[86] A. Martínez. Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *IEEE Transactions on Pattern analysis and machine intelligence*, pages 748–763, 2002.

[87] A. Martinez and R. Benavente. The AR face database. Technical report, CVC, 1998.

[88] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1) :63–86, 2004.

[89] E. Miller, N. Matsakis, and P. Viola. Learning from one example through shared densities on transforms. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, SC, USA*, volume 1, pages 464–471, 2000.

[90] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7) :971–987, 2002.

[91] M. Osborne and A. Rubinstein. *A course in game theory.* The MIT press, 1994.

[92] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. Rasl : Robust alignment by sparse and low-rank decomposition for linearly correlated images. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 763–770, 2010.

[93] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. Rasl : Robust alignment by sparse and low-rank decomposition for linearly correlated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 2012.

[94] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *Proc. IEEE Conference on Computer vision and pattern recognition*, volume 1, pages 947–954, 2005.

[95] P. Phillips, H. Moon, S. Rizvi, and P. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10) :1090–1104, 2000.

[96] E. Rentzeperis, A. Stergiou, A. Pnevmatikakis, and L. Polymenakos. Impact of face registration errors on recognition. In *Artificial intelligence applications and innovations : 3rd IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI) 2006, June 7-9, 2006, Athens, Greece*, 2006.

[97] M. Roh, T. Oguri, and T. Kanade. Face alignment robust to occlusion. In *IEEE Conference on Automatic Face and Gesture Recognition and Workshops, Santa Barbara, CA, USA*, 2011.

[98] D. Ross, J. Lim, R. Lin, and M. Yang. Incremental learning for robust visual tracking. *Internationl Journal of Computer Vision*, 77 :125–141, 2007.

[99] S. Roweis. Em algorithms for pca and spca. *Advances in neural information processing systems*, pages 626–632, 1998.

[100] A. Salah, H. Çinar, L. Akarun, and B. Sankur. Robust facial landmarking for registration. *Annals of Telecommunications*, 62(1-2) :1608–1633, 2007.

[101] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *Proc. the Second IEEE Workshop on Applications of Computer Vision*, pages 138–142. IEEE, 1994.

[102] J. Saragih, S. Lucey, and J. Cohn. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2) :200–215, 2011.

[103] S. Shan, Y. Chang, W. Gao, B. Cao, and P. Yang. Curse of mis-alignment in face recognition : Problem and a novel mis-alignment learning solution. In *6th IEEE Conference on Automatic Face and Gesture Recognition, Seoul, Korea*, 2004.

[104] H. Shum and R. Szeliski. Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2) :101–130, 2000.

[105] C. Stewart. Robust parameter estimation in computer vision. *Siam Review*, pages 513–537, 1999.

[106] J. Suo, F. Min, S. Zhu, S. Shan, and X. Chen. A multi-resolution dynamic model for face aging simulation. In *IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, USA, 2007.

[107] R. Szeliski. Image alignment and siticing : A tutorial. Technical Report MSR-TR-2004-92, Microsoft research, 2006.

[108] Y. Tong, X. Liu, F. Wheeler, and P. Tu. Automatic facial landmark labeling with minimal supervision. In *IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA*, 2009.

[109] A. Torsello, S. Bulo, and M. Pelillo. Grouping with asymmetric affinities : A game-theoretic perspective. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 292–299, 2006.

[110] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment—a modern synthesis. *Vision algorithms : theory and practice*, pages 153–177, 2000.

[111] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1) :71–86, 1991.

[112] G. Tzimiropoulos, V. Argyriou, S. Zafeiriou, and T. Stathaki. Robust fft-based scale-invariant image registration with image gradients. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10) :1899–1906, 2010.

[113] G. Tzimiropoulos and S. Zafeiriou. On the subspace of image gradient orientations. *Arxiv preprint arXiv :1005.2715*, 2010.

[114] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. Robust and efficient parametric face alignment. In *International Conference on Computer Vision, Barcelona, Spain*, 2011.

[115] M. Valstar, B. Martinez, X. Binefa, and M. Pantic. Facial point detection using boosted regression and graph models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[116] A. Vedaldi, G. Guidi, and S. Soatto. Joint data alignment up to (lossy) transformations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[117] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[118] P. Viola and W. Wells III. Alignment by maximization of mutual information. *International journal of computer vision*, 24(2) :137–154, 1997.

[119] N. Vu and A. Caplier. Face Recognition with Patterns of Oriented Edge Magnitudes. In *European Conference on Computer Vision, Crete, Greece*, 2010.

[120] N. Vu and A. Caplier. Enhanced patterns of oriented edge magnitudes for face recognition and image matching. *IEEE Transactions on Image Processing*, 21(3) :1352–1365, 2012.

[121] D. Vukadinovic and M. Pantic. Fully automatic facial feature point detection using Gabor feature based boosted classifiers. In *IEEE International Conference on Systems, Man and Cybernetics*, 2005.

[122] P. Wang, M. Green, Q. Ji, and J. Wayman. Automatic eye detection and its validation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition - Workshops, San Diego, CA, USA*, 2005.

[123] L. Wiskott, J. Fellous, N. Kuiger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7) :775–779, 1997.

[124] H. Wu, X. Liu, and G. Doretto. Face alignment via boosted ranking model. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[125] S. Yan, H. Wang, J. Liu, X. Tang, and T. Huang. Misalignment-robust face recognition. *IEEE Transactions on Image Processing,*, 19(4) :1087–1096, 2010.

[126] M. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images : A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1) :34–58, 2002.

[127] A. Yilmaz, O. Javed, and M. Shah. Object tracking : A survey. *Acm Computing Surveys (CSUR)*, 38(4) :13, 2006.

[128] C. Zhang and Z. Zhang. A survey of recent advances in face detection. *Microsoft Research*, 2010.

[129] L. Zhang, H. Ai, and S. Lao. Robust face alignment based on hierarchical classifier network. *Computer Vision in Human-Computer Interaction*, pages 1–11, 2006.

[130] C. Zhao, W. Cham, and X. Wang. Joint face alignment with a generic deformable face model. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, USA*, pages 561–568, 2011.

[131] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld. Face recognition : A literature survey. *Acm Computing Surveys (CSUR)*, 35(4) :399–458, 2003.

[132] Y. Zhou, L. Gu, and H. Zhang. Bayesian tangent shape model : Estimating shape and pose parameters via bayesian inference. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–109, 2003.

[133] Z. Zhou and X. Geng. Projection functions for eye detection. *Pattern Recognition*, 37(5) :1049–1056, 2004.

[134] B. Zitova and J. Flusser. Image registration methods : a survey. *Image and vision computing*, 21(11) :977–1000, 2003.