The Dissertation Committee for Birgi Tamersoy
certifies that this is the approved version of the following dissertation:

Facial Feature Localization using Highly Flexible yet Sufficiently Strict Shape Models

Committee:

J.K. Aggarwal, Supervisor

Ross Baldick

Craig M. Chase

Wilson S. Geisler III

Kristen Grauman

Facial Feature Localization using Highly Flexible yet Sufficiently Strict Shape Models

by

Birgi Tamersoy, B.S.; M.S.E.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2014

Dedicated to my parents,

Çimen & Mahmut Tamersoy.

Facial Feature Localization using Highly Flexible yet Sufficiently Strict Shape Models

Publication No. ____________________

Birgi Tamersoy, Ph.D.
The University of Texas at Austin, 2014

Supervisor: J.K. Aggarwal

Accurate and efficient localization of facial features is a crucial first step in many face-related computer vision tasks. Some of these tasks include, but not limited to: identity recognition, expression recognition, and head-pose estimation.

Most effort in the field has been exerted towards developing better ways of modeling prior appearance knowledge and image observations. Modeling prior shape knowledge, on the other hand, has not been explored as much.

In this dissertation I primarily focus on the limitations of the existing methods in terms of modeling the prior shape knowledge. I first introduce a new pose-constrained shape model. I describe my shape model as being “highly flexible yet sufficiently strict”. Existing pose-constrained shape models are either too strict, and have questionable generalization power, or they are too loose, and have questionable localization accuracies. My model tries to find a good middle-ground by *learning* which shape constraints are more "informative" and should be kept, and which ones are not-so-important and may be omitted.

I build my pose-constrained facial feature localization approach on this new shape model using a probabilistic graphical model framework. Within this framework, observed and unobserved variables are defined as the *local* image observations, and the feature locations, respectively. Feature localization, or "probabilistic inference", is then achieved by nonparametric belief propagation. I show that this approach outperforms other popular pose-constrained methods through qualitative and quantitative experiments.

Next, I expand my pose-constrained localization approach to unconstrained setting using a *multi-model strategy.* While doing so, once again I identify and address the two key limitations of existing multi-model methods: 1) semantically and manually defining the models or "guiding" their generation, and 2) not having efficient and effective model selection strategies. First, I introduce an approach based on unsupervised clustering where the models are *automatically* learned from training data. Then, I complement this approach with an efficient and effective model selection strategy, which is based on a multi-class naïve Bayesian classifier. This way, my method can have many more models, each with a higher level of expressive power, and consequently, provides a more effective partitioning of the face image space. This approach is validated through extensive experiments and comparisons with state-of-the-art methods on state-of-the-art datasets.

In the last part of this dissertation I discuss a particular application of the previously introduced techniques; facial feature localization in unconstrained videos. I improve the frame-by-frame localization results, by estimating the *actual* head-movement from a sequence of noisy head-pose estimates, and then using this information for detecting and fixing the localization failures.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Face perception is a critical aspect of social interactions. One may almost instantly recognize familiar faces, generate names for them, bring-up relevant memories, and interpret expressions. This seemingly easy task is possibly one of the most highly developed visual perceptual skills humans have [31].

A commonly accepted cognitive model for face perception argues that understanding faces involves multiple, distinct processes [7]. This functional model of Bruce and Young is presented in Figure 1.1.

From a psychological point of view the highlight of this model is the fact that face perception involves multiple, distinct processes. For this dissertation though the highlight is much more subtle, and is the fact that all these processes, either directly or indirectly, depend on some initial, low-level, view-dependent representation of the perceived face.

Perhaps not so surprisingly, face-related computer vision research follows a very similar model to the one in Figure 1.1. Furthermore, in computer vision, this "initial, low-level, view-dependent representation" is fairly well defined, and is based on "facial features".

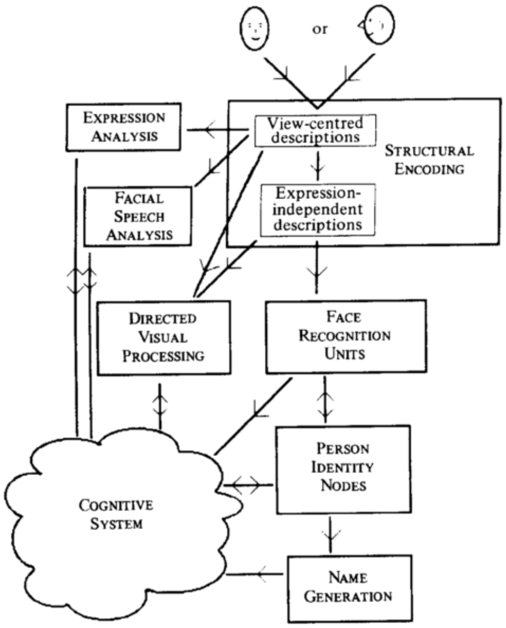


Image Credit: Bruce and Young [7].

Figure 1.1: A functional model for face processing.

Facial features are the salient points on a face. These points usually consist of a set of anatomical landmarks (such as the eye corners) and a set of pseudo-landmarks that are equally-spaced between these anatomical landmarks (Figure 1.2). *The focus of this dissertation is "facial feature localization", which is the problem of detecting these landmarks in face images.*

This is an important problem because just like the "view-dependent descriptions" of the human perception model in Figure 1.1, most of the face-related computer vision applications rely on accurate and efficient localization of facial features. Some of these applications include: facial identity recogni-

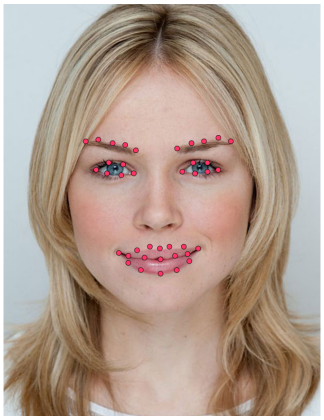


Image Credit: http://www.dailymail.co.uk.

Figure 1.2: Facial features.

tion, facial expression recognition, head-pose estimation, and markerless facial motion capture.

Although it seems that the appearance of these features and their configuration across the population are fairly uniform [28], from a computer's perspective they are *not*, and this is what makes facial feature localization a remarkably difficult task. As illustrated in Figure 1.3, facial features and their configuration change significantly in face images depending on: the facial attributes of the person, the expression of the person, the viewer (i.e. camera) angle, and the imaging conditions.

Facial feature localization is by no means a new problem. In fact related work may be traced back to early 1970s making this one of the very first problems, which the researchers in the field were interested in. That being said, after more than 40 years we are yet to find solutions that are robust, accurate,

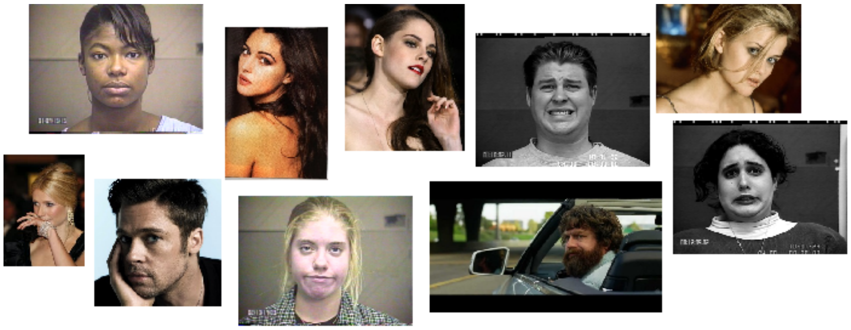


Figure 1.3: Facial features and their configuration change significantly in unconstrained face images.

and efficient enough to satisfy the demands of the real-world applications. Our collective progress so far has been truly impressive, but the real exciting part is what we may possibly achieve in the future.

1.1 Overview of Dissertation

Any facial feature localization algorithm needs to blend-in two distinct sources of information in order to achieve good results:

- Prior knowledge, and
- Image observations.

This is actually true for many computer vision problems. Systems that rely only on image observations exhibit high false positive and false negative rates. Whereas systems that fuse this information with prior domain knowledge are usually capable of eliminating most, if not all, of these false positives and false negatives.

In facial feature localization, the prior knowledge consists of two parts: *prior appearance knowledge*, and *prior shape knowledge*. The former is our prior knowledge on the appearance of these facial features, and the latter is our prior knowledge on their locations with respect to each other.

Interestingly, most of the previous work in the field focus on different ways of modeling the prior appearance knowledge and/or the image observations. In contrary, modeling the prior shape knowledge has not been explored as much.

In the following subsections I provide brief summaries of the main components of my dissertation. First, I introduce a nonparametric facial feature localization method. This method is accompanied by a *learned frontal face shape model*, and consequently is “pose-constrained”. Next, I introduce a *divide-and-conquer* strategy for expanding the pose-constrained approach to the unconstrained setting. This strategy is based on partitioning the face image space into smaller subsets, and learning one pose-constrained model for each of these partitions. Finally, I discuss facial feature localization in unconstrained videos as a specific application of the previously introduced techniques. While doing so, I use *head-pose continuity* as a heuristic to improve the localization results.

1.1.1 Constrained Facial Feature Localization

Depending on how the prior shape knowledge is modeled and enforced, all existing methods may be grouped into three categories: *parameterized shape models*, *part-based shape models*, and *implicit shape models.*

By far the most common approach among the related work is the parameterized shape models (e.g. [12, 14, 19, 45, 53]). These methods model the non-rigid shape variations *linearly*, where the variational bases are learned from training data. They are holistic in nature and this makes them fairly strict models of shape.

If we imagine a "strictness spectrum" for shape models, and position the parameterized shape models on the "strict end" of this spectrum, the "loose end" would contain the existing part-based models (e.g. [8, 26, 58, 63, 69]). These methods model the face as a *configuration of parts.* A common strategy among part-based methods is to omit some inherited facial shape constraints for the sake of computability. This usually results in models which are too loose for accurate facial feature localization.

A relatively more recent strategy is to model and enforce the shape constraints with the use of "shape regressors" (e.g. [9, 44, 67]). These methods are of less interest to this dissertation, since they do *not* employ explicit shape models. When the "big picture" of face processing is considered, the benefits of an explicit shape model is apparent.

The insight here is to have a shape model which is highly flexible so *that it generalizes well to unseen images, but at the same time strict enough so that it does not allow unnatural deformations.* With this insight, I introduce a novel part-based shape model, which tries to find a good middle-ground within the previously mentioned "strictness spectrum". My model "learns" which shape constraints are inherently "more important" and captures them, while omitting the "not-so-important" constraints. This way, it is *necessarily* more strict than the existing part-based models, while still being a lot more flexible than the existing parameterized models.

I use this *learned* shape model as the basis of a probabilistic graphical model, where the feature locations become the unobserved random variables, and the shape constraints are captured within the graph topology. Image observations are incorporated into this graphical model as observed variables. Within this framework, facial feature localization (i.e. probabilistic inference in the graph) is achieved through nonparametric belief propagation.

The details of this framework is discussed in Chapter 3. Section 3.1 provides more insight on the motivation, primarily focusing on the limitations of the existing methods. Then, the reader is provided with the required technical background knowledge in Section 3.2, which includes:

1. Procrustes Analysis (Section 3.2.1),

2. Probabilistic Graphical Models (Section 3.2.2),

3. Belief Propagation (Section 3.2.3), and

4. Nonparametric Belief Propagation (Section 3.2.4).

Next, Section 3.3 goes into the details of the introduced framework and Section 3.4 validates the method through qualitative and quantitative results and comparisons. The chapter is concluded with a discussion in Section 3.5.

1.1.2 Unconstrained Facial Feature Localization

So far, I have introduced a pose-constrained facial feature localization method, which is based on a "highly flexible, yet sufficiently strict" shape model. This method is capable of accurately localizing facial features in *generic near-frontal face images.* In other words, it can handle the feature appearance and configuration changes due to facial attributes and/or mild facial expressions, but it *cannot* handle the changes due to viewer angle (or equivalently subject pose) and/or extreme facial expressions. A single two-dimensional shape model is simply not powerful enough to precisely model that level of prior knowledge.

That being said, we are interested in facial feature localization in the wild due to ever increasing amounts of such unconstrained data. In order to achieve this, once again, we need to model our prior domain knowledge and incorporate that with the image observations. However, this time modeling the prior domain knowledge is much harder since now the domain covers *all* face images.

Existing methods have addressed this problem in one of three ways: by employing more sophisticated shape models (usually three-dimensional, e.g.

[6,65]), by employing multiple two-dimensional shape models (e.g. [15,69]), or by using elaborate "shape regressors" (e.g. [9,67]).

Out of these three strategies, using multiple two-dimensional shape models is preferable due to several reasons:

1. Even though faces are three-dimensional objects, and a three-dimensional shape model seems like an intuitive way of modeling the prior shape and appearance knowledge, in reality building such models is not an easy task. The resulting models are usually *not powerful enough* to cover all shape and appearance variations exhibited in unconstrained localization.

2. Methods which employ elaborate shape regressors model the prior shape and appearance knowledge implicitly. However, having an explicit shape model is more intuitive and more beneficial when the "big picture" is considered.

3. The constrained facial feature localization method which I have introduced is based on a single two-dimensional shape model. Hence, expanding this with a multi-model strategy is more natural.

Here, I introduce a new multi-model approach for attacking the unconstrained localization problem. This method is based on partitioning the space of all face images into a number of clusters and then training separate cluster-specific shape and appearance models. *Unlike any existing multi-model* *approach, I do not define these clusters semantically (or "guide" their generation manually), but rather automatically learn them from a very large training dataset.*

For the cluster classification task I train a multi-class naïve Bayesian classifier using *ferns* [47] as image features. These features are extremely efficient to compute, and the resulting cluster classifier is proved to be sufficiently discriminative.

This divide-and-conquer approach effectively reduces the difficult "unconstrained localization" problem into a set of much simpler "constrained localization" sub-problems, each with very precise shape and appearance models.

This approach is further discussed in Chapter 4. Section 4.1 provides more insight on the motivation, primarily focusing on the limitations of the existing methods. Then, the reader is provided with the required technical background knowledge in Section 4.2, which includes:

1. Principal Component Analysis (Section 4.2.1), and
2. k-means Clustering (Section 4.2.2).

Next, Section 4.3 goes into the details of the *divide-and-conquer* strategy and Section 4.4 validates the method through qualitative and quantitative results and comparisons. The chapter is concluded with a discussion in Section 4.5.

1.1.3 Facial Feature Localization in Videos

Facial feature localization in *unconstrained videos* is a challenging task. These videos consist of *a sequence of unconstrained images* and hence form a very good testbed for the underlying localization method. A successful method is expected to handle significantly different head-poses, as well as the transitions in between.

Most existing facial feature localization methods (including the ones I have introduced in the previous sections) are designed for still images. Usually these methods are extended to work with videos by leveraging the *motion continuity*. In most cases, this step simply involves initializing the models in one frame, with the results obtained in the previous frame. Clearly this approach is exploiting the motion continuity within *the image domain*, and may result in significant failures due to accumulating errors. Common ways of addressing this limitation include: using more elaborate tracking techniques (such as Kalman filters [35]) and/or failure detection mechanism which are based on trained classifiers (e.g. [53]).

Unlike these methods, I primarily exploit the motion continuity in *the real-world domain*. In an unconstrained video with a high-enough frame-rate, the subject is expected to exhibit a continuous motion. This *actual* head-movement may be used for detecting and fixing the localization failures.

In order to do so, one first needs to estimate the head-pose in each frame. For this, I formulate the head-pose estimation as an optimization problem, where yaw, pitch, and roll angles are estimated by fitting a generic three-dimensional face model to the two-dimensional feature localization results. This formulation makes crude assumptions (such as image formation based on orthographic projection), but nevertheless performs well with challenging examples.

Next, I take the estimated head-pose parameters within a window of frames, and fit n-order polynomials to these results for enforcing a *continuous* head-motion. The fitted polynomials provide the “expected” parameter values, which are then used for detecting and fixing the failures. Furthermore, these polynomials also provide a means for “predicting” the head-pose parameters in near-future frames.

This method is further discussed in Chapter 5. Section 5.1 provides more insight on the motivation. Then, the reader is provided with the required technical background knowledge in Section 5.2, which includes:

1. Orthographic Projection (Section 5.2.1).

Next, Section 5.3 goes into the details of the approach highlighting the head-pose continuity heuristic and Section 5.4 validates the method through qualitative results and comparisons. The chapter is concluded with a discussion in Section 5.5.

1.2 Key Contributions

The key contributions of this dissertation are three-fold:

- Modeling the prior shape knowledge in facial feature localization has *not* been well explored. Most existing work focuses on modeling the prior appearance knowledge and/or the image observations, while taking the shape models for granted. Instead in this dissertation, I focus primarily on shape modeling. I introduce a novel model, which is *highly flexible, yet sufficiently strict*, and formulate the constrained facial feature localization problem as an inference problem within an intuitive probabilistic graphical model framework. This framework, accompanied by the new shape model, addresses the limitations of the existing constrained localization methods.

- The two key limitations of the existing multi-model unconstrained localization methods are: 1) semantically and manually defining the models, and 2) not having efficient and effective model selection strategies. Consequently, these methods have relatively few number of semantically-fixed models. It is questionable whether these methods may take full advantage of ever growing amounts of training data, without serious manual intervention. I address both of these two limitations in this dissertation. First, I introduce an approach based on unsupervised clustering where the models are *learned* from the training data. Then, I complement this approach with an efficient and effective model selection strategy, which is based on a multi-class naïve Bayesian classifier. This way, if in a few years the amount of training data we have increases by n-fold, all it will take for my method to take full advantage of this much larger dataset is a simple re-clustering and re-training.

- For last, I explore the problem of facial feature localization in unconstrained videos as a validation of the previously introduced techniques. I show how "head-pose continuity" may be leveraged for improving the frame-by-frame localization results by: 1) detecting and fixing past failures, and 2) predicting future models.

1.3 Road Map

The rest of this dissertation is organized as follows: In Chapter 2 I provide an in-depth review of the related work. I review the existing work in two primary, and several secondary categories, so that the reader has a better understanding of where the techniques described in this dissertation fit in. In Chapter 3, I present my constrained facial feature localization method, with an emphasis on the underlying shape model. Later in Chapter 4, I describe a "divide-and-conquer" strategy for expanding the *constrained* localization framework of Chapter 3 to *unconstrained* setting. In Chapter 5 I present a particular application: facial feature localization in videos, and discuss how *head-pose continuity* may be leveraged as a heuristic in order to improve the frame-by-frame localization results. Finally, I conclude by summarizing my key contributions in Chapter 6.

Chapter 2

Related Work

In this chapter, I review the related work in two main sections: *constrained* and *unconstrained (i.e. in-the-wild)* facial feature localization. Within each section, I first categorize the methods based on how the prior shape knowledge is modeled, and then discuss the categorized work in more detail.

2.1 Constrained Facial Feature Localization

I define "constrained facial feature localization" to be the problem of localizing facial features in *near-frontal* face images. Both generic and person-specific localization may be considered "constrained" if the input domain consists of only frontal-face images. Hence, the methods I discuss here are primarily "pose-constrained", rather than anything else.

Based on how the prior shape knowledge is modeled, existing methods may be categorized into three groups: *part-based shape models*, *parameterized shape models*, and *implicit shape models*. Former two groups explicitly define shape models. On contrary, methods in the third group enforce the shape constraints implicitly. In this section I review the related work in the first two groups, and postpone the review of implicit shape models to the next section.

2.1.1 Part-based Shape Models

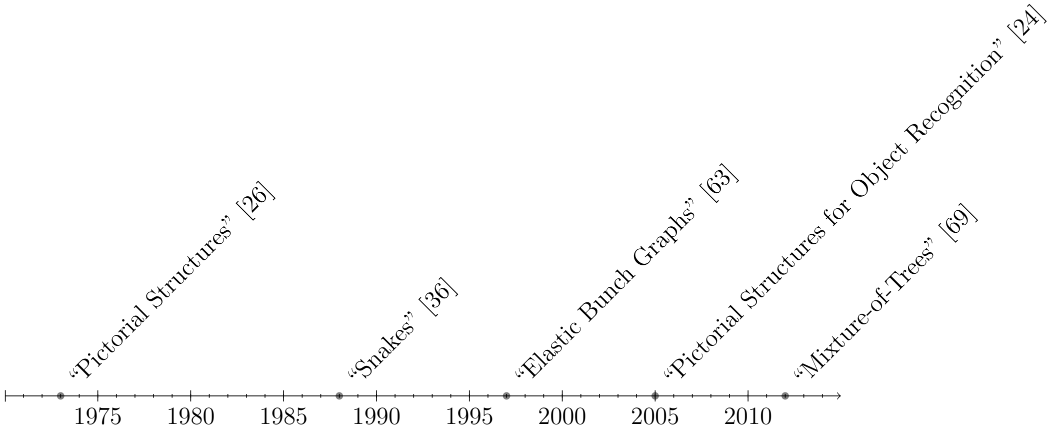


Figure 2.1: Selected influential work based on part-based shape models.

Figure 2.1 illustrates some of the selected influential work in this category.

The seminal work of Fischler and Elschlager [26], published in 1973, may be considered one of the very first related works in this field. Even though the authors' problem statement involves "object detection", their object model is "composed of a number of rigid pieces (components) held together by 'springs'" (see Figure 2.2), and hence inherently involves localizing the parts of the corresponding object. Foundations laid by this work in terms of defining "part-based shape models" have been used intensively in the object detection (or "recognition") field (e.g. [8, 22–25]), but not so much in the facial feature localization field (e.g. [57, 63, 69]). Hence, in this section I do *not* limit the discussion with the methods which particularly attach the problem of facial feature localization. Instead, I also mention some related work from the object detection field, where sometimes the objects of interest are “faces”.

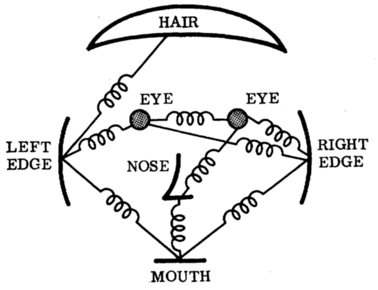


Image credit: Fischler and Elschlager [26]

Figure 2.2: Spring model of a face.

As Figure 2.2 demonstrates, part-based methods model the objects as a configuration of parts. Depending on the object and the application of interest, these parts may be defined in varying levels of abstraction. For example, in Figure 2.2, parts (e.g. “left eye”, “hair”, etc.) are defined at a relatively higher level since the application of interest is face detection. On the other hand, in facial feature localization, these parts are defined as the *facial features*, which are at a much lower abstraction level.

An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a natural way of expressing such part-based models. Here, $\mathcal{V} = \{v_1, \ldots, v_n\}$ represent the n parts of the model, whereas the edges, $(v_i, v_j) \in \mathcal{E}$, represent the encoded spatial relationships. The location of the object in an image is given by a configuration of its parts $\mathcal{L} = \{l_1, \ldots, l_n\}$, where $l_i = (x_i, y_i)$ is the location of the i^{th} part. Then the localization problem may be formulated as finding the most probable configuration of the parts given an image I:

$$\mathcal{L}^* = \arg\max_{\mathcal{L}}[P(\mathcal{L}|I)] = \arg\max_{\mathcal{L}}[P(I|\mathcal{L})\ P(\mathcal{L})] \tag{2.1}$$

It is useful to further examine two terms in this optimization problem: $P(I|\mathcal{L})$ and $P(\mathcal{L})$. $P(I|\mathcal{L})$ represents the image support of a given configuration of parts, and in a way models both the prior appearance knowledge *and* the image observations. $P(\mathcal{L})$, on the other hand, represents the likelihood of a particular configuration, and hence models the prior shape knowledge.

Part-based facial feature localization methods usually differ at how they define and incorporate: $\mathcal{V}$, $\mathcal{E}$, and $P(\mathcal{L})$. Most of the existing work either over-simplifies the graph structure $\mathcal{E}$, or over-constraints $P(\mathcal{L})$, primarily to make the inference tractable and efficient.

Due to the algorithmic and computational limitations of the time, Fischler and Elschlager [26] employ binary-state edge potentials. Wiskott et al. [63], on the other hand, define "graph elasticity" in terms of the difference between the edges of the fitted model and the actual shape model, but choose to adjust the landmark locations *individually*, over a *single-pass* procedure. Note that both of these earlier works do have cyclic graphs (i.e. graphs with loops) as their shape models.

A common over-simplification of the graph structure is to assume that $\mathcal{E}$ is an acyclic graph (i.e. a tree). Felzenszwalb and Huttenlocher [24] show that when the graph $\mathcal{G}$ is a tree, and the pairwise spatial relationships are assumed to be Mahalanobis distances, the model matching essentially becomes linear in the number of parts. Ramanan and Sminchisescu [50] further improve this idea by using Gaussian relationships and a tree-structured Conditional Random Field (CRF) [38] as the shape model. Recently, Zhu and Ramanan [69] proposed a “mixture-of-trees” model, where they model multiple facial configurations with individual trees. The tree property allows for efficient inference, and gives good results with relatively simpler object classes such as airplanes, motorcycles, and horses. However, it lacks the necessary loopy spatial constraints and produces unnatural deformations in the case of facial feature localization.

Crandall et al. [17] addresses this limitation of the tree-structured graphs by introducing a class of spatial priors, which they call “k-fans” (see Figure 2.3). k-fans depend on choosing k reference nodes from the n parts of the shape model. The reference nodes form a fully connected sub-graph, and each non-reference node is simply connected to every reference node. For simpler object classes, k-fans do capture more spatial relationships. However, for the case of facial feature localization, choosing the correct number and the configuration of these reference nodes is not intuitive. Moreover, k-fans may introduce unnecessary constraints, while failing to capture the necessary ones.

Another fairly less explored approach involves approximating the probabilistic inference, rather than the underlying shape model. Sudderth et al. [56] approximates the complex node potentials and spatial relationships using Gaussian kernel density estimates (KDEs) [54]. Just like any other work in

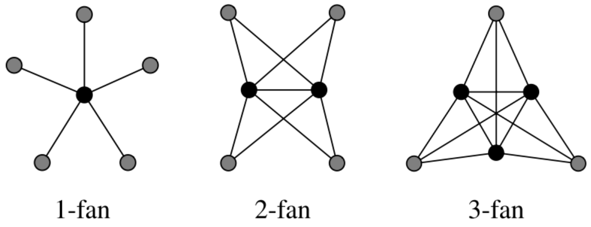


Image credit: Crandall et al. [17]

Figure 2.3: Some k-fans on 6 nodes.

this and the following section, Sudderth et al. *manually* define the underlying shape model (see Figure 2.4). Nodes in this model are high-dimensional feature vectors, representing *both* the location and the appearance of the corresponding part (e.g. left eye).

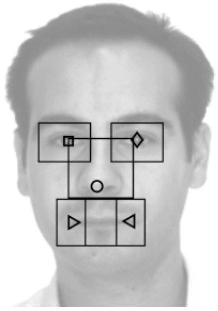


Image credit: Sudderth et al. [56]

Figure 2.4: Part-based face model used in [56].

Other simplifications applied by existing methods in order to make the solution of Equation 2.1 tractable include: assuming that the parts of the model are jointly Gaussian [25], defining graph nodes with binary states (nodes representing part relations, rather than part locations) [44, 58], and doing “hard-detection” (first detecting candidates and *then* applying the constraints [10, 18, 58]. All of these methods try to simplify the initial localization problem, in order to employ efficient inference algorithms.

The last method I discuss in this section is called “Snakes” [36]. The basic snake model is a “controlled continuity spline”. The continuity of the spline is enforced through “internal spline forces”, whereas “external image forces” push the snake towards salient image features. Since the snake model consists of a set of loosely connected vertices, it may still be considered a part-based shape model. Note that, the snake model does *not* encode global shape constraints. Nevertheless, the parameterized shape models that are reviewed in the next section are all influenced by this basic model.

2.1.2 Parameterized Shape Models

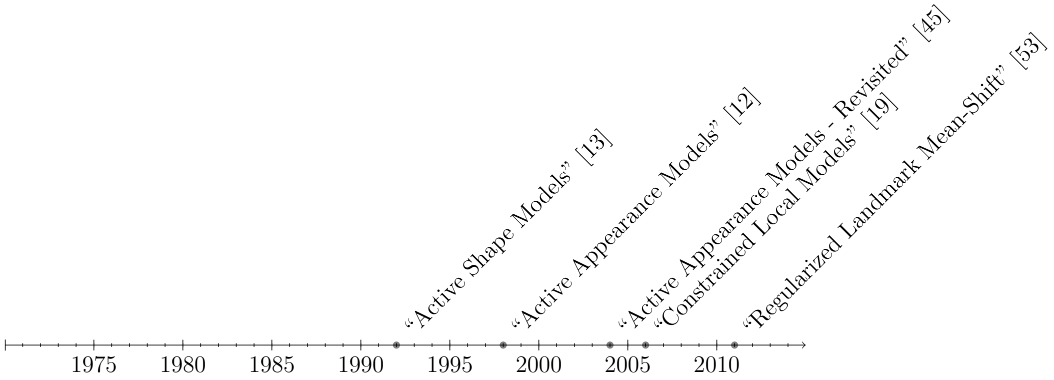


Figure 2.5: Selected influential work based on parameterized shape models.

Figure 2.1.2 illustrates some of the selected influential work in this category. Even though the parameterized shape models were introduced almost two decades after the work of Fischler and Elschlager [26], they have been the most popular method for modeling the prior shape knowledge in the facial feature localization field.

Cootes and Taylor [13] have laid the foundations of parameterized shape models in their seminal work "Active Shape Models - 'Smart Snakes' ". They have named their approach "Smart Snakes" because they address a particular limitation of the basic snake model of Kass et al. [36]. As I have discussed in the previous section, regular snakes do not encode any global shape constraints. The fitting procedure only enforces that the fitted spline has "controlled continuity", which maps to simple and local shape constraints. Most object classes, on the other hand, have well defined global shape constraints (such as faces, resistors, cars, etc.).

Cootes and Taylor [13] address this limitation by their "Point Distribution Model (PDM)". PDM models the non-rigid shape variations of an object linearly and composes it with a global similarity transform (see Figure 2.6):

$$x = sR(\bar{x} + \Phi q) + t \tag{2.2}$$

where $x = (x_0, y_0, \ldots, x_n, y_n)^T$ denote the 2D-coordinates of the n feature points, and $p = \{s, R, t, q\}$ are the PDM parameters consisting of a global scaling s, a rotation R, a translation t, and a set of non-rigid shape parameters q. Here, $\bar{x}$ is the *mean shape*, and $\Phi = [\phi_1, \ldots, \phi_k]$ is a matrix consisting of k linearly independent modes of variation (i.e. "shape vectors"). Both $\bar{x}$ and

Φ are estimated from a set of training examples using Principal Component Analysis (PCA) [49].

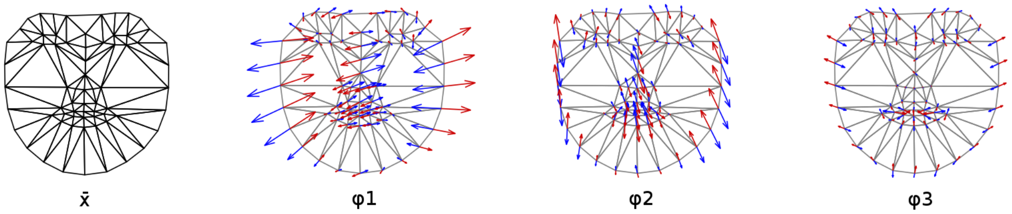


Image credit: Matthews and Baker [45]

Figure 2.6: The mean shape and the shape vectors used in PDMs.

Given the PDM of Equation 2.2, the facial feature localization problem may be formulated as a minimization over the model parameters:

$$p^* = \arg\min_{p} [R(p) + D(x; I)] \tag{2.3}$$

where $R(p)$ is *the regularization term*, and $D(x; I)$ represents *the misalignment error* associated with model x given image I. One may think of these two terms as analogous to $P(\mathcal{L})$ and $P(I|\mathcal{L})$ of Equation 2.1, respectively. Here, $R(p)$ is used to bound the model parameters so that no unnatural deformations are allowed. $D(x; I)$, on the other hand, is a measure representing the fitting quality based on the *image observations*. Almost all PDM-based methods solve Equation 2.3 iteratively, and relate the image observations to the parameter updates δp for computing:

$$p \leftarrow p + \delta p \tag{2.4}$$

Depending on how the image observations are modeled and incorporated with the PDM of Equation 2.2, methods in this category may further be divided into two main classes. The first class of methods, which are derived from "Active Shape Models" [13], make use of *local* image observations. These methods are collectively named Constrained Local Models (CLMs). CLMs utilize an independent set of local detectors (i.e. "experts") to obtain feature-specific *response maps*. Then, the optimization in Equation 2.3 is performed over these local response maps (see Figure 2.7).

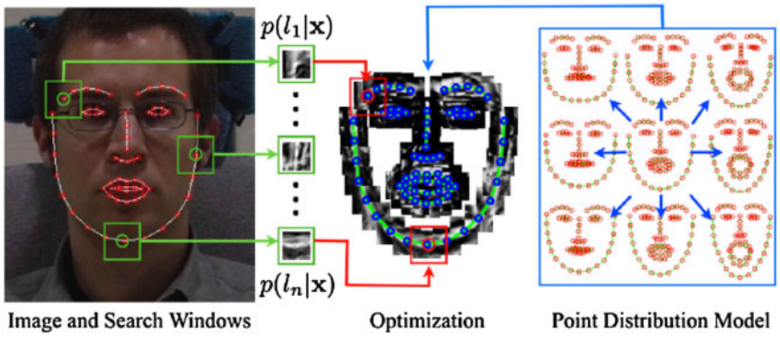


Image credit: Saragih et al. [53]

Figure 2.7: Constrained local models.

Nonparametric and noisy response maps are usually replaced with parametric approximations in order to make the problem more stable and tractable. Using isotropic Gaussians (e.g. [13]), anisotropic Gaussians (e.g. [46, 60, 68]), or Gaussian Mixture Models (e.g. [30]), are common techniques for this. Even though the resulting objectives are easier to solve, parametric approximations are just crude approximations of the true response maps and may potentially miss out important details. Saragih et al. [53] addresses this limitation by using nonparametric approximations in the form of homoscedastic isotropic

Gaussian kernel density estimates (KDEs) [54].

Cristinacce and Cootes [19] follow a slightly different approach for both obtaining the response maps, and handling the noise associated with these response maps. They first built a joint parameterized shape *and* "local" texture model from a set of training images. During the fitting, at each iteration they generate feature-specific "templates" using the estimated model parameters, and then use these templates to obtain the response maps, making their method more robust.

The second class of methods, derived from "Active Appearance Models (AAMs)" [12], use a more holistic approach for modeling and incorporating the prior appearance knowledge. Unlike CLMs, which embed this prior knowledge in local experts, in AAMs holistic "shape-normalized" images are used for determining the parameter updates δp. "Shape-normalization" is a piece-wise linear warping of a particular face image into the mean shape ($\bar{x}$ of Equation 2.2). This process removes all shape variations, and consequently results in an "appearance-only" image.

In the initial AAM formulation of Cootes et al. [12], a second PCA is performed on shape-normalized training images to get a "mean appearance" and a set of linearly independent "appearance vectors" (see Figure 2.8). At each iteration of the fitting, the current parameter estimates are used to generate a model, which is then used to compute an "error image". In this work the error images are related to the parameter updates δp using linear regression.

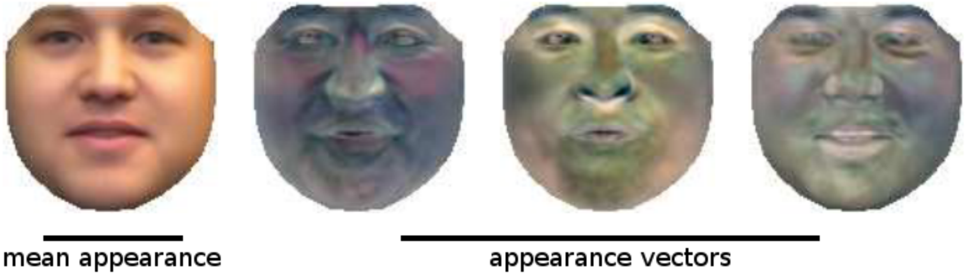


Image credit: Matthews and Baker [45]

Figure 2.8: The mean appearance and the appearance vectors used in AAMs.

Cootes et al. [12] employ linear regression primarily due to efficiency concerns. Using linear regression does make the parameter update computations very fast, but nevertheless it is a rough approximation of the real relationship. Matthews and Baker [45] address this limitation by interchanging the roles of input images with the generated models. Their efficient inverse compositional algorithm results in a much faster and a more accurate localization.

These initial AAM formulations [12, 45] perform well in person-specific feature localization (i.e. training images contain the images of the test subject). However, as discussed in Gross et al. [29] the performance degrades rapidly when the same formulation is used for generic facial feature localization. Gross et al. [29] attributes this degradation primarily to the shape component of the AAM. That being said, most AAM-based methods try to address this limitation by re-formulating the appearance component of AAMs in different ways.

Liu [41], Wu et al. [64], and Saragih and Goecke [52], all choose to model the appearance by a set of discriminative features (usually Haar-like features [59]) instead of the PCA-based approach described above. In "Boosted Appearance Model" of Liu [41], these features are related to the parameter updates through the use of a trained binary boosted classifier, which distinguishes between *correct* and *incorrect* alignments. "Boosted Ranking Model" of Wu et al. [64] takes this one step further by introducing a boosted classifier that behaves as an *alignment score function.* Similarly, Saragih and Goecke [52] learns a *nonlinear boosted regressor* to relate the appearance features to the parameter updates of the model.

2.2 Unconstrained Facial Feature Localization

Simple 2D shape models are not capable of modeling the significant head-pose and expression variations in face images captured in uncontrolled settings. Consequently, in unconstrained facial feature localization we need more sophisticated methods for modeling the prior shape and appearance knowledge.

A common strategy among the researchers is to employ multiple-models in order to address the challenges of unconstrained feature localization (see Figure 2.9). "View-based Active Appearance Models (AAM)" of Cootes et al. [15], "Direct Appearance Models (DAM)" of Li et al. [40], and "Mixture-of-Trees" of Zhu and Ramanan [69] are all examples of such multi-model methods. Former two methods employ a relatively small number of linear shape models, whereas Zhu and Ramanan [69] use a “mixture-of-trees” consisting of 18 tree-models to model the shape constraints. All existing multi-model methods define their models manually and semantically based on head-pose [15, 40] and/or facial expressions [69], or manually “guide” their generation [34].

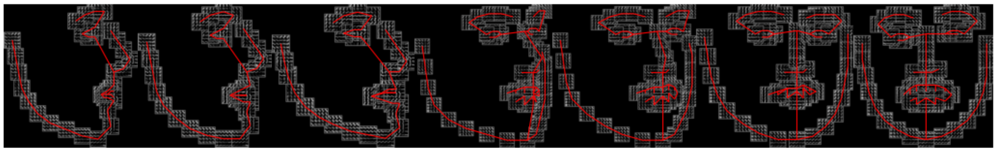


Image credit: Zhu and Ramanan [69]

Figure 2.9: “Mixture-of-Trees” model of Zhu and Ramanan [69].

Belhumeur et al. [3] use all the training shapes (around 1000 of them) as global models, and hence in a way avoids defining explicit shape models. Given a new face image, the authors identify a set of best matching global models, which are then used as prior shape knowledge.

Similarly Cao et al. [9] and Yang and Patras [67] choose not to use explicit shape models. Shape constraints are implicitly enforced in shape regressors of [9] and in the head center sieve of [67].

More recently, Supervised Descend Method of Xiong and De la Torre [66], and Discriminative Response Map Fitting approach of Asthana et al. [1], are shown to perform well in unconstrained feature localization. In [66], a sequence of generic descent directions are learned from training data, and then used in model fitting. [1], on the other hand, learns a set of weak learners, which model the non-linear relationship between the response maps and the

3D shape model parameter update.

Chapter 3

Constrained Facial Feature Localization

3.1 Motivation

Existing shape models that are used for constrained facial feature localization are either too strict or too loose. If a shape model is too strict, its generalization power is questionable. This is the case for parameterized shape models. They are holistic models of shape, where the non-rigid shape variations are modeled linearly. Changing one parameter, in theory, affects *all* the feature locations in this shape model.

On the other hand, if a shape model is too loose, then it allows for unnatural deformations. This results in poor localization results. Existing part-based shape models face this difficulty. In order to employ efficient inference algorithms, they either omit some necessary spatial constraints, or make unrealistic assumptions about the feature relationships.

In this chapter, I introduce a new shape model, which aims at finding a good compromise between these two extremes. My part-based model *learns* “important” constraints and captures them while omitting the not-so-important ones. This way, it is necessarily more strict than the other part-based models, while still being more flexible than the parameterized models.

This shape model is incorporated with *local* image observations within a probabilistic graphical model framework, where inference is achieved by non-parametric belief propagation.

3.2 Background

In the following subsections I provide the required technical background. First, I define the concept of a "true shape", and discuss Procrustes Analysis, which is a method for "aligning" a set of shapes. Then, I provide a brief explanation of Probabilistic Graphical Models, and discuss the details of an approximate inference technique called "Belief Propagation". In the last section, I focus on an extension of the standard Belief Propagation algorithm called "Nonparametric Belief Propagation".

3.2.1 Procrustes Analysis

A "true shape" is defined as *"all the geometrical information that remains when location, scale, and rotational effects are filtered out from an object"* [21]. In other words, the "true shape" of an object is invariant to the Euclidean similarity transforms [55].

When we are trying to model our prior shape knowledge about an object, it is very important to filter out these similarity transforms, so that our model captures only the "true shape", and is not affected by the shape variations due to similarity transforms.

A two-dimensional shape with N-points (i.e. "landmarks") may be represented by a matrix $p \in \mathbb{R}^{2\times N}$, where $p_n = [x_n, y_n]^T$ represents the two-dimensional coordinates of the n^{th} point. "Aligning" one shape p_2 to another shape p_1 may then be formulated as a minimization problem:

$$\{s^*, R^*, t^*\} = \underset{s,R,t}{\arg\min} \,||\, p_1 - sR[p_2 + t] \,||^2 \tag{3.1}$$

where, $s \in \mathbb{R}$, $R \in \mathbb{R}^{2\times 2}$, and $t \in \mathbb{R}^{2\times N}$, represent the alignment parameters: scale, rotation, and translation, respectively.

Procrustes Analysis provides an algorithm for the solution of this alignment problem:

1. **Compute the centroids** - The centroid of a shape p is defined as the center of mass of the physical system consisting of unit masses at each landmark:

$$\bar{p} = (\bar{p^x}, \bar{p^y}) = (\frac{1}{N}\sum_{n=1}^{N} x_n, \frac{1}{N}\sum_{n=1}^{N} y_n) \tag{3.2}$$

2. **Re-scale the shapes** - Each shape is scale-normalized using the *Frobenius norm* as the size metric:

$$p_1 = \frac{p_1}{FN(p_1)} \qquad p_2 = \frac{p_2}{FN(p_2)} \tag{3.3}$$

where

$$FN(p) = \sqrt{\sum_{n=1}^{N} (x_n - \bar{p^x})^2 + (y_n - \bar{p^y})^2} \tag{3.4}$$

3. **Align w.r.t. location** - A common strategy is moving both shapes to the center of the coordinate system:

$$p_1 = p_1 - \bar{p_1} \qquad p_2 = p_2 - \bar{p_2} \tag{3.5}$$

4. **Align w.r.t. rotation** - The best rotational alignment between p_2 and p_1 is determined by computing the Singular Value Decomposition (SVD) of $p_1 p_2^T$:

$$UDV^T = p_2 p_1^T \tag{3.6}$$

where VU^T gives us the R^* of Equation 3.1.

The "Generalized Procrustes Analysis (GPA)" is an extension of this algorithm used for aligning a set of shapes (see Figure 3.1). At each iteration of the GPA algorithm all shapes in the set are aligned to the mean shape, and then the mean shape is re-computed. This iterative refinement is done until the mean shape does not change.

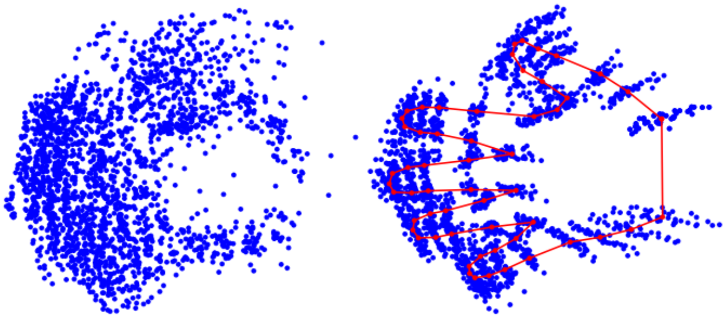


Image Credit: [55].

Figure 3.1: Generalized Procrustes Analysis (left: unaligned shapes, right: aligned shapes with mean shape in red).

3.2.2 Probabilistic Graphical Models (PGMs)

PGMs use a graph-based representation as the basis for compactly encoding complex joint distributions over multiple, high-dimensional random variables [37]. An undirected graph $\mathcal{G}$ is defined by a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$ (see Figure 3.2). Each node $s \in \mathcal{V}$ represents either an unobserved, or hidden, random variable x_s, or a noisy local observation y_s. Following the notation in [56], the neighborhood of a node $s \in \mathcal{V}$ is defined as $\Gamma(s) \triangleq \{t|(s,t) \in \mathcal{E}\}$.

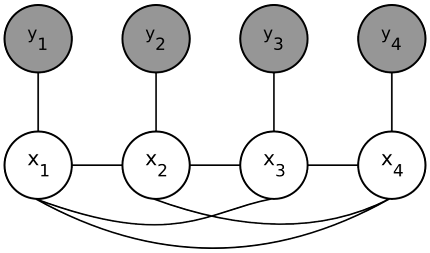


Figure 3.2: An undirected graph.

In undirected, pairwise PGMs the joint distribution over all variables $p(x, y)$ factorizes as:

$$p(x,y) = \frac{1}{Z} \prod_{(s,t)\in\mathcal{E}} \phi_{s,t}(x_s, x_t) \prod_{s\in\mathcal{V}} \phi_s(x_s, y_s) \tag{3.7}$$

where Z is a normalization constant, $\phi_{s,t}(x_s, x_t)$ is the *compatibility potential* between nodes s and t, and $\phi_s(x_s, y_s)$ is the *observation potential* of node s.

While the joint distribution $p(x, y)$ is hard to estimate, in many applications, the real interest is in the computation of conditional marginal distributions $p(x_s|y)$ for all $x_s \in \mathcal{V}$.

3.2.3 Belief Propagation (BP)

BP provides a convenient way for computing the conditional marginal distributions $p(x_s|y)$ in a graph. At iteration n of the BP algorithm, each node $t \in \mathcal{V}$ sends a message $m_{t,s}^{n}(x_s)$ to each of its neighbors $s \in \Gamma(t)$:

$$m_{t,s}^{n}(x_s) = \alpha \int_{x_t} \phi_{s,t}(x_s, x_t)\phi_t(x_t, y_t) \prod_{u \in \Gamma(t)\setminus s} m_{u,t}^{n-1}(x_t)dx_t \tag{3.8}$$

where α denotes a proportionality constant.

At any iteration n, the *belief* of node s about the hidden variable x_s may be computed as follows:

$$\hat{p^n}(x_s|y) = \alpha\phi_s(x_s, y_s) \prod_{u \in \Gamma(s)} m_{u,s}^{n}(x_s) \tag{3.9}$$

BP algorithm guarantees that the node beliefs will converge to the correct conditional marginals in singly connected graphs [48]. Even though there is little theoretical analysis on the performance of BP in graphs with loops ([61, 62]), loopy BP has shown excellent empirical performance in a number of applications [4, 27].

3.2.4 Nonparametric Belief Propagation (NBP)

Equation 3.8 may be evaluated analytically only when both the compatibility and the observation potentials have special forms. When both are Gaussians, the calculations are straightforward since the product of a number of Gaussian densities is another Gaussian. When either potential is a Gaussian mixture and the other one is a Gaussian or a Gaussian mixture, still the integration is straightforward, but now the number of mixture components increase exponentially at every iteration. And when the potentials do not have special forms, analytical evaluation of the integral in Equation 3.8 becomes intractable.

In order to address this limitation of the BP algorithm, Sudderth et al. [56] and Isard [33] independently developed almost identical algorithms, which incorporate particle filters into the BP framework.

In these algorithms, nonparametric Gaussian KDEs [54] are used to represent the messages at each iteration. Then the BP update rule defined in Equation 3.8 becomes:

$$m_{t,s}^n(x_s) = \sum_{i=1}^{M} w_s^{(i)} \mathcal{N}(x_s; \mu_s^{(i)}, \Lambda_s) \tag{3.10}$$

where $w_s^{(i)}$ is the weight associated with the i^{th} kernel with mean $\mu_s^{(i)}$ and bandwidth Λ_s.

Given input messages $m_{u,t}(x_t)$ for each $u \in \Gamma(t) \backslash s$, the output message $m_{t,s}(x_s)$ is then computed as follows:

1. Draw M independent samples $\{\hat{x}_t^{(i)}\}_{i=1}^M$ from $\phi_t(x_t, y_t) \prod_{u \in \Gamma(t) \backslash s} m_{u,t}^{n-1}(x_t) dx_t$, and
2. For each $\{\hat{x}_t^{(i)}\}_{i=1}^M$ sample $\hat{x}_s^{(i)} \sim \phi_{s,t}(x_s, x_t = \hat{x}_s^{(i)})$.

Details may be found in [56].

3.3 Approach

In this section I introduce a new framework for constrained facial feature localization. I first define my shape model, and then explain how the prior appearance knowledge and image observations are modeled and incorporated into this shape model. In the last subsection, I discuss the handling of the similarity transforms, which is very important for robustness.

3.3.1 Prior Shape Knowledge

In my formulation, $x = \{x_s | s \in \mathcal{V}\}$ represent the 2D landmark locations, and $y = \{y_s | s \in \mathcal{V}\}$ represent the corresponding local image observations.

As previously discussed, the proposed approach may be thought as modeling the prior shape knowledge in terms of *multiple*, *weak*, *pairwise* spatial relationships. In order to fully specify this shape model, one needs to define both the pairwise compatibility potentials *and* the topology of the underlying graph.

3.3.1.1 Compatibility Potentials

Anisotropic Gaussians are used to model the pairwise compatibility potentials:

$$\phi_{s,t}(x_s, x_t) = \mathcal{N}((x_s - x_t); \mu_{s,t}, \Sigma_{s,t}) \tag{3.11}$$

where $\mu_{s,t}$ is the mean, and $\Sigma_{s,t}$ is the covariance matrix of the Gaussian. Both of these parameters are learned from training data.

This potential encloses the prior shape knowledge between two landmarks, since given the location of a landmark x_t and the potential $\phi_{s,t}(x_s, x_t)$, one may estimate the likely locations of landmark x_s by:

$$\phi_{s,t}(x_s, x_t = \hat{x}_t) = \mathcal{N}(x_s; \mu_{s,t} + \hat{x}_t, \Sigma_{s,t}) \tag{3.12}$$

As Figure 3.3 illustrates, anisotropic Gaussians model the compatibility potentials well. Furthermore, one may estimate the "importance" of a particular pairwise potential within the model, simply by examining the learned covariance matrices. A potential with a smaller $\Sigma_{s,t}$ will have a higher precision, and hence would be more informative than a potential with a larger $\Sigma_{s,t}$.

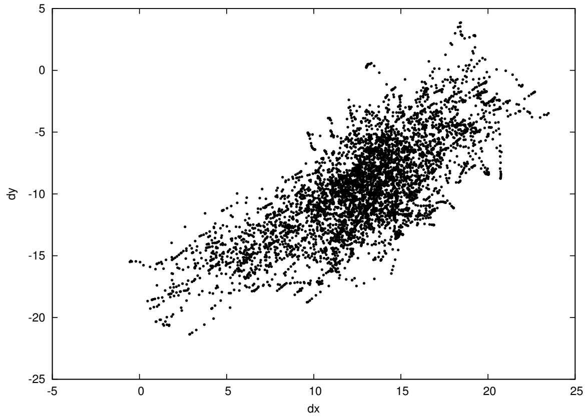


Figure 3.3: Example $(x_s - x_t)$ scatter plot. Note that an anisotropic Gaussian would model this distribution fairly well.

3.3.1.2 Graph Topology

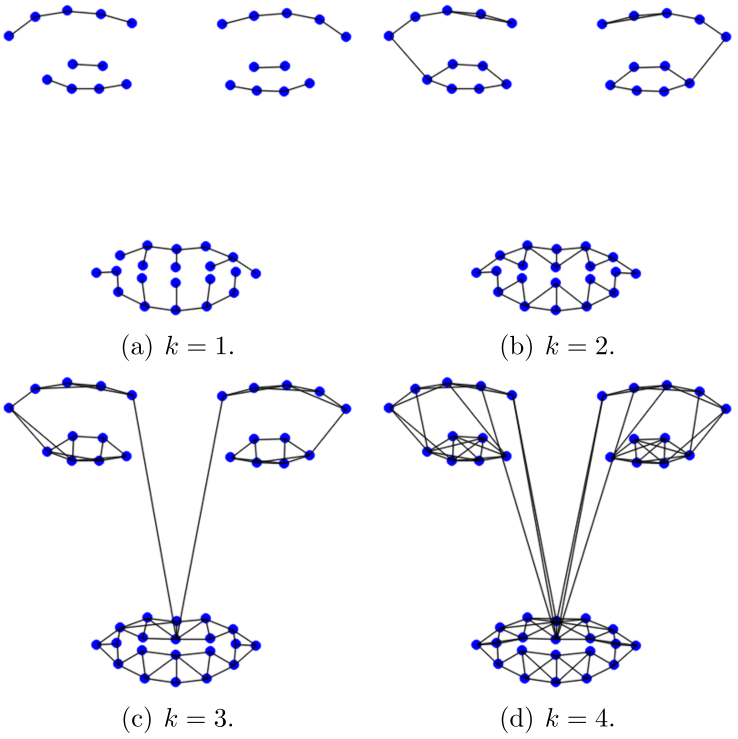
One of the primary advantages of using a PGM-based shape model is the flexibility in determining the graph topology. The possibilities range from a loose, singly connected graph, to a very strict, fully connected graph. A singly connected graph would reminiscence Snakes [36], whereas parametrized shape models [14] would be considered fully connected graphs.

In this work, the graph topology of the shape model is *learned* from training data. For each node s, the neighborhood $\Gamma(s)$ is determined using the computed compatibility potentials. Only the k most informative (smaller $\Sigma_{s,t}$) nodes are connected to node s. Note that the "informativeness" of a neighbor is determined with respect to the eigenvectors of the covariance matrix of the corresponding pairwise relationships. This approach allows for capturing a lot of shape knowledge in a fairly simple graph. Figure 3.4 illustrates the computed topologies for $k = 1, 2, 3, 4$.

Please note that the proposed approach actually generates a "class" of spatial models, rather than just a single one. Hence, the appropriate level of flexibility may be chosen with respect to the application.

3.3.2 Prior Appearance Knowledge and Image Observations

A variety of observation models have been used in the facial feature localization literature. These methods vary from using gradients as in the case for Snakes [36], to using holistic error images as in the case of AAMs (e.g. [12, 45]). CLMs, on the other hand, use "local experts" (i.e. local patch



(a) $k = 1$.

(b) $k = 2$.

(c) $k = 3$.

(d) $k = 4$.

Figure 3.4: Learned graph topologies for $k = 1, 2, 3, 4$.

detectors), and have shown to perform superior [53]. Hence, I employ a “local” modeling approach in this work as well, which is illustrated in Figure 3.5.

The local experts used in this work are linear support vector machines (SVMs) [16] and the features used are 3×3 histograms of oriented-gradients (HOGs) [20] with $6-$bin histograms in each cell.

Inspired by the results of Saragih et al. in [53], the observation potentials are defined to be the nonparametric isotropic Gaussian KDEs [54] of the

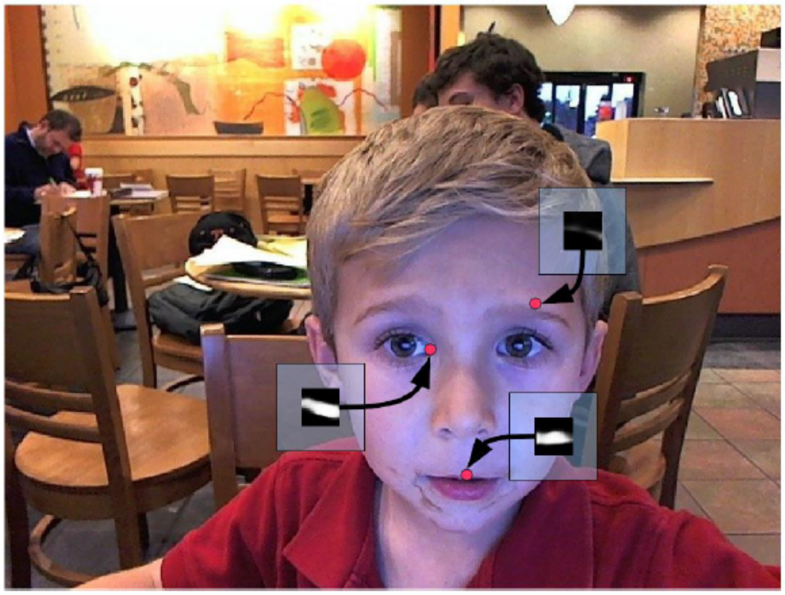


Figure 3.5: Local response maps obtained at several feature locations.

expert response maps:

$$\phi_s(x_s, y_s) = \sum_{z_i \in \Psi_s} \pi_{z_i} \mathcal{N}(x_s; z_i, \rho I) \tag{3.13}$$

where Ψ_s denotes the set of integer pixel locations within a square region centered at x_s, ρ is the bandwidth of the kernels, and π_{z_i} is the probabilistic expert response at location z_i.

This observation potential has two advantages:

1. Its Gaussian mixture form fits well into the NBP framework (much better than than the one proposed in [56]), and allows for the employment of efficient sampling methods (e.g. [32]), and

2. It estimates the true response maps much better than the existing parametric methods [53].

3.3.3 Handling Similarity Transforms

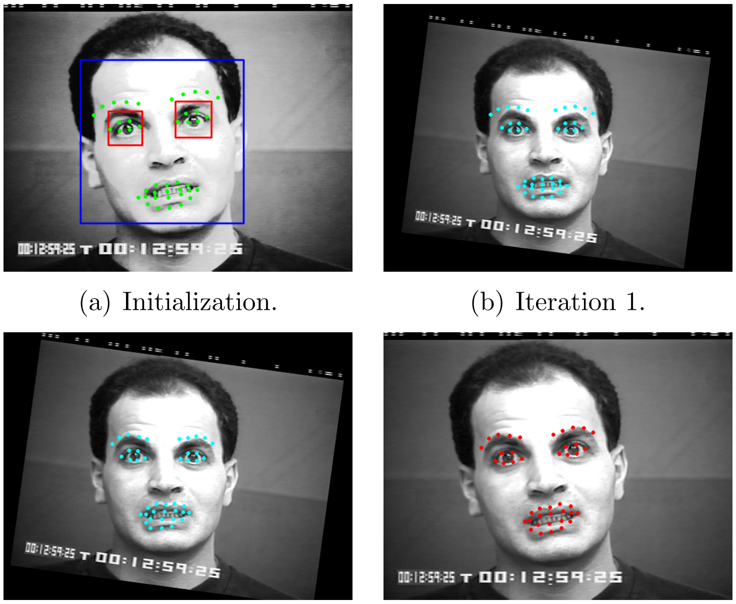
Similarity transforms may be incorporated into this model either by scaling and rotating the compatibility potentials (Equation 3.11), or by keeping them constant, but instead aligning the current landmark estimates *and* the image with the mean scale and rotation at each iteration. I followed the second approach since it also implicitly solves the scale and rotation variance of the experts. This fitting process is illustrated in Figure 3.6.

3.4 Results

Extensive qualitative and quantitative experiments are performed on The Extended Cohn-Kanade Dataset (CK+) [42] and random images obtained from the Internet. These experiments contain subjects with different ethnicities, performing acted and/or spontaneous expressions. Imaging conditions and quality change significantly between the examples.

3.4.1 Shape and Expert Training

"Ground-truth" landmarks provided by the CK+ dataset are used for both shape and local expert training. Chin and nose region landmarks are ignored since these landmarks contribute much less information in most applications.


00:12:59:25 T 00:12:59:25
00:12:59:25 T 00:12:59:25
00:12:59:25 T 00:12:59:25
00:12:59:25 T 00:12:59:25

(a) Initialization.

(b) Iteration 1.

(c) Iteration 2.

(d) Result.

Figure 3.6: The fitting process. At each iteration the image is “similarity normalized” by aligning the current shape with the mean shape (best viewed in color and high-resolution).

For the shape training, first the landmarks are shape normalized using Generalized Procrustes Analysis [21] (see Figure 3.7). Then the compatibility potential parameters, $\mu_{s,t}$ and $\Sigma_{s,t}$, are computed. Finally the graph topology is determined using the computed covariance matrices as illustrated in Figures 3.4(a)-3.4(d).

Figure 3.4 demonstrates a major advantage of this algorithm. By learning the graph topology from training data, we effectively obtain the smallest graph that would capture the most prior shape knowledge.

The resulting graph, in the case of faces, is a very intuitive one, where

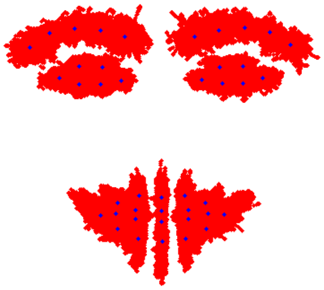


Figure 3.7: Alignment of the training images.

the parts of the face (e.g. eyes, mouth, etc.) are densely connected, while the parts themselves are loosely connected. Such a model will allow for a high variability between the locations of the parts, but at the same time enforce more strict constrains on how the parts themselves may deform. In these experiments I used $k = 3$ connectivity.

24×24 patches are used to train the local experts. For each landmark, positive examples are obtained from 1000 randomly selected images. Approximately 8000 negative examples are extracted from the remaining landmarks and other randomly selected images. LIBSVM [11] library is used for the SVM training.

3.4.2 Testing

Unless otherwise specified, all test images are *automatically* initialized. Local "search window", Ψ_s, of Equation 3.13 is set to be a 23×23 region centered around the current estimate. ρ is set to 1 and finally $M = 200$ particles are used for belief propagation.

Algorithm convergence is determined using the node beliefs. At each iteration the beliefs (i.e. landmark locations) are computed and when none of the landmarks move more than 1.5 pixels in radius, the algorithm is terminated.

3.4.3 Qualitative Results

Qualitative results on the CK+ dataset are presented in Figure 3.8. As the figure illustrates, the proposed algorithm performs equally well in a wide variety of examples, where both the facial expressions *and* the facial attributes of the subject change significantly. This is primarily due to the higher level of shape flexibility provided by the model.

3.4.4 Qualitative and Quantitative Comparisons

The proposed approach is compared with two state-of-the-art methods in Figures 3.9 and 3.10: 1) the "Tree-model" by Zhu and Ramanan [69], and 2) CLM by Saragih et al. [53].

A total of 5876 images from 327 sequences have been tested. For every sequence, the first frame is automatically initialized. Every other frame in the sequence is initialized with the results of the previous frame.

As Figures 3.9 and 3.10 illustrate, both CLM and the proposed method significantly outperforms the "Tree-model". Even though similar tree-models perform well in other applications (such as part-based object classification), for facial feature localization they are too flexible, and hence allow unnatural

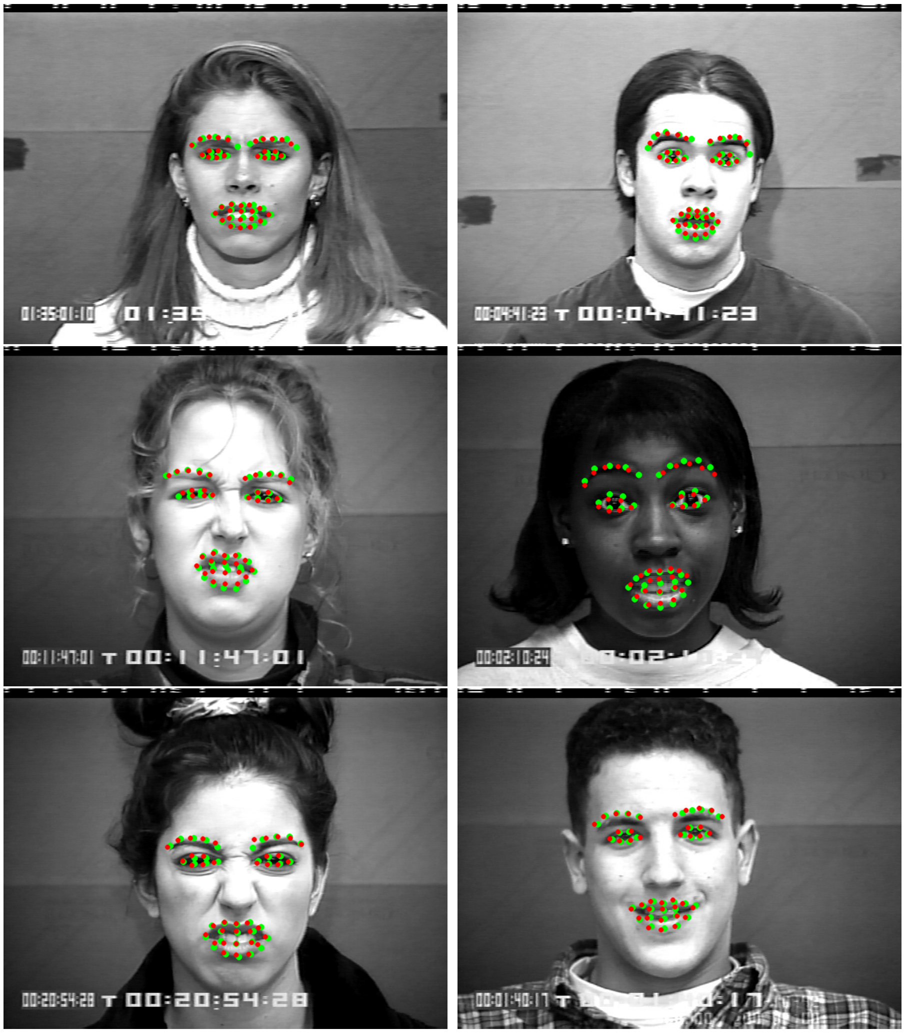

01:35:01:10
00:04:41:23 T 00:04:41:23
00:11:47:01 T 00:11:47:01
00:02:10:24
00:20:54:28 T 00:20:54:28
00:01:40:17

Figure 3.8: Qualitative results of the proposed PGM-based approach (best viewed in color and high-resolution, green: ground truth, red: results).

deformations in the shape.

Out of 5876 tested images, the proposed approach achieved a lower average error in 3468 images (59.02%), CLM achieved a lower average error

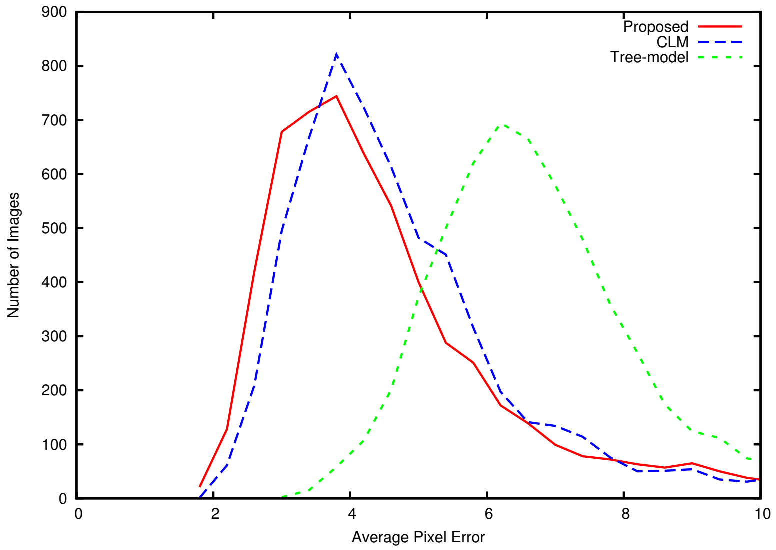
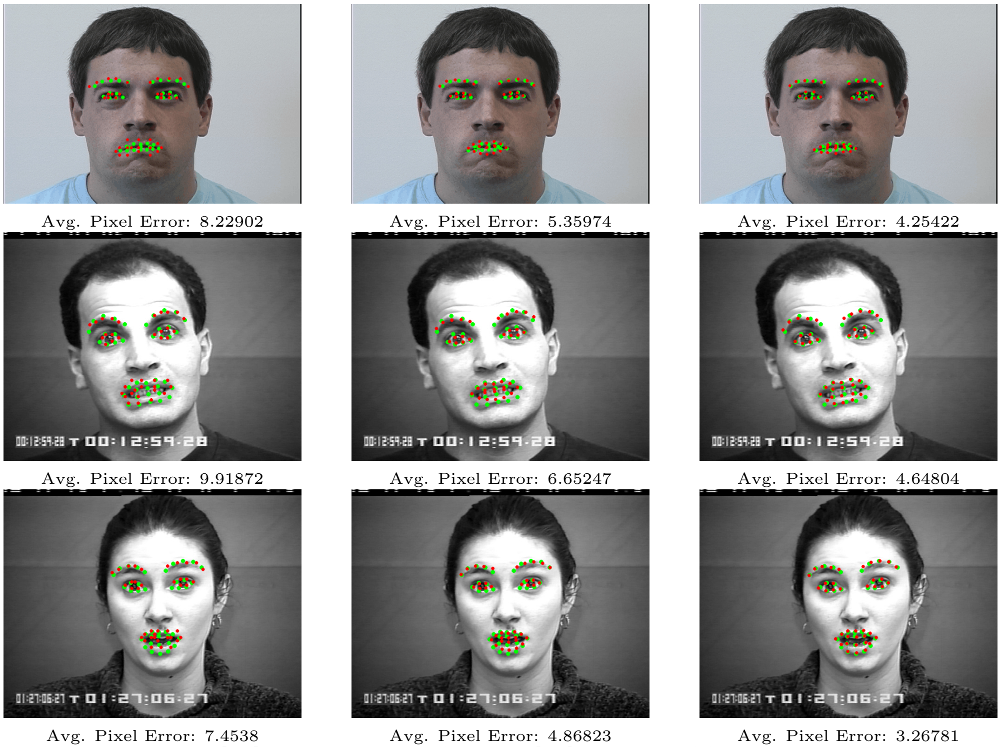


Figure 3.9: Quantitative comparison of Tree-model [69], CLM [53] and the proposed approach (best viewed in color).

in 2296 images (39.07%) and the Tree-model achieved a lower average error in 112 images (1.91%). Corresponding error distributions are presented in Figure 3.9.

3.4.5 Generalization

Even though the proposed algorithm is trained on a relatively small, fairly controlled dataset, as Figures 3.11 and 3.15 illustrate, it generalizes very well to real world images. This may be explained by two primary properties: 1) pairwise unimodal Gaussian compatibility potentials in the shape model allow for a great level of flexibility and generalization power, and 2) the HOG features capture the "generic" appearance properties of the landmarks very


Avg. Pixel Error: 8.22902
Avg. Pixel Error: 5.35974
Avg. Pixel Error: 4.25422
Avg. Pixel Error: 9.91872
Avg. Pixel Error: 6.65247
Avg. Pixel Error: 4.64804
Avg. Pixel Error: 7.4538
Avg. Pixel Error: 4.86823
Avg. Pixel Error: 3.26781

(a) Tree-model [69]. (b) CLM [53]. (c) The proposed approach.

Figure 3.10: Qualitative comparison of Tree-model [69], CLM [53] and the proposed approach (best viewed in color and high-resolution, green: ground truth, red: algorithm-specific results, less green seen means a better fit).

well.

The generalization power and the localization accuracy of the proposed algorithm is further demonstrated in Figures 3.12 and 3.13. In these figures the proposed algorithm is qualitatively compared with the "Tree-model" [69] and CLM [53] on two unseen examples. Especially Figure 3.13 provides interesting results since this is a *fictitious* input, where the face of the subject is half-Asian and half-Caucasian.

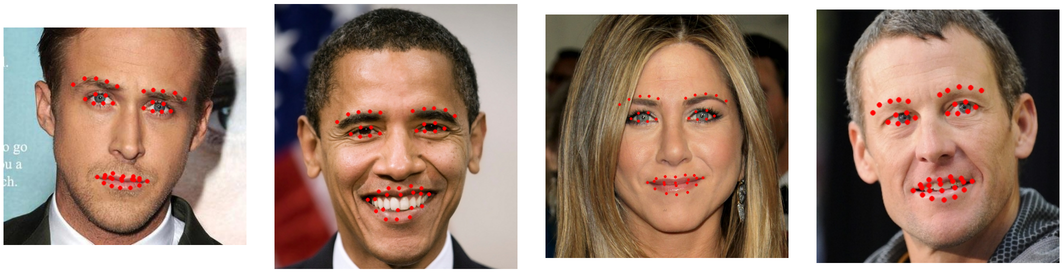
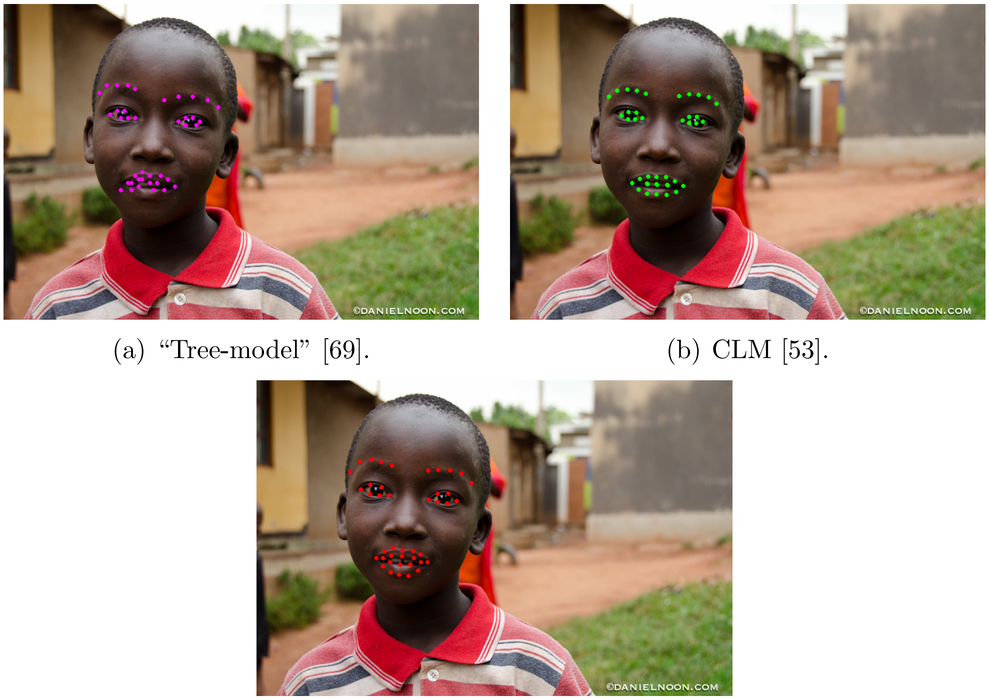


Figure 3.11: Generalization to random Internet images (best viewed in color and high-resolution).

As the figures illustrate, CLM is very strict, and hence does *not* generalize well to unseen shape variations. The “Tree-model”, on the other hand, does generalize well to unseen shape variations, *but* allows for unnatural deformations while doing so. The proposed algorithm addresses both of these limitations.

3.4.6 Importance of the Graph Topology

The fictitious input of Figure 3.13 is further used in Figure 3.14 for demonstrating the importance of the graph topology. The experiments in these figures are done using the same initialization, the same local experts, the same compatibility potentials, and the same observation potentials, *but* different graph topologies. In Figure 3.14(a) 75 randomly selected edges are used, whereas in Figure 3.14(b) 75 “most informative” (see Section 3.3.1.2) edges are selected. The importance of the graph topology on the localization results may easily be seen especially in the mouth region, where there are more complicated loopy spatial constraints.



(a) “Tree-model” [69].

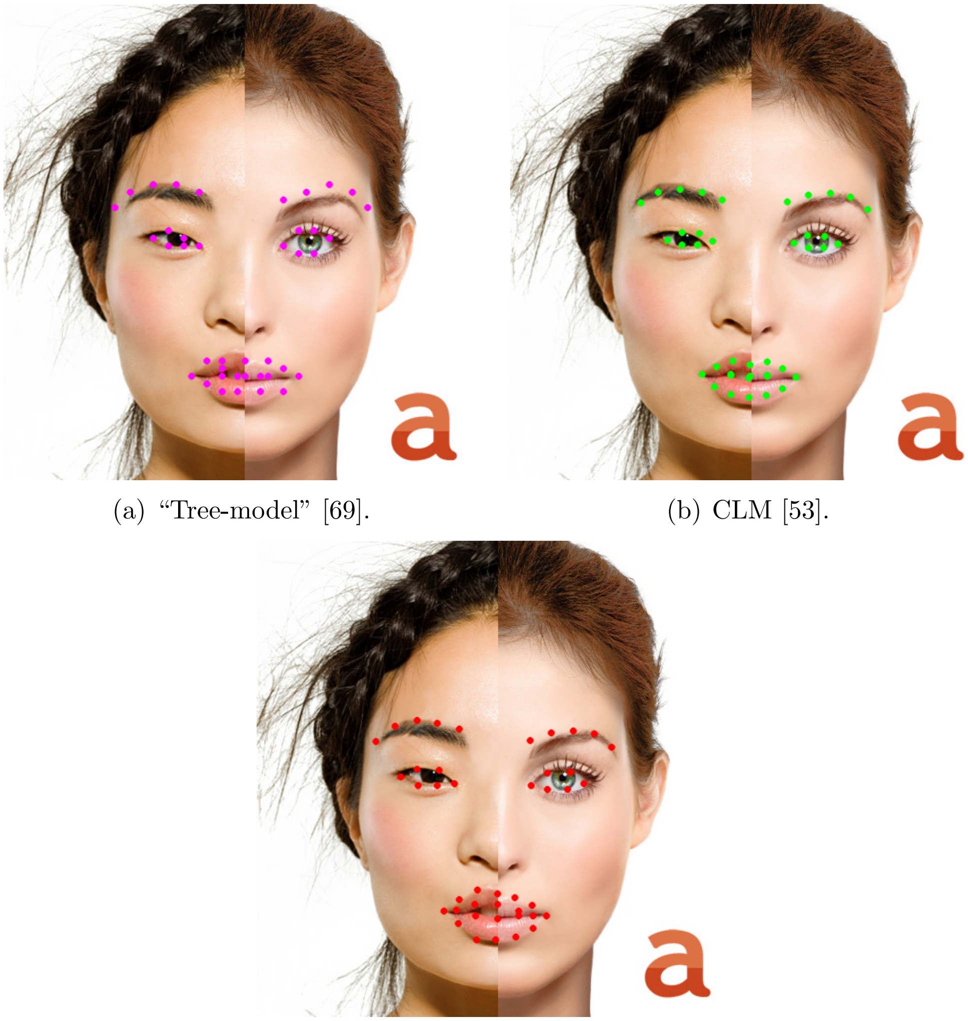
(b) CLM [53].

(c) The proposed approach.

Figure 3.12: Qualitative comparison in terms of generalization power and localization accuracy.

3.4.7 Implicit Occlusion Handling

Unlike parameterized shape models, my approach models the prior shape knowledge as pairwise local spatial relationships. Figure 3.15 illustrates an important advantage of this local model over the holistic approaches. Even with highly occluded faces: 1) the visible landmarks are not affected from the occlusion, and 2) reasonable predictions can be made about the occluded landmarks. Please note that the results in the figure are obtained without *any* explicit occlusion handling mechanism.



(a) “Tree-model” [69].

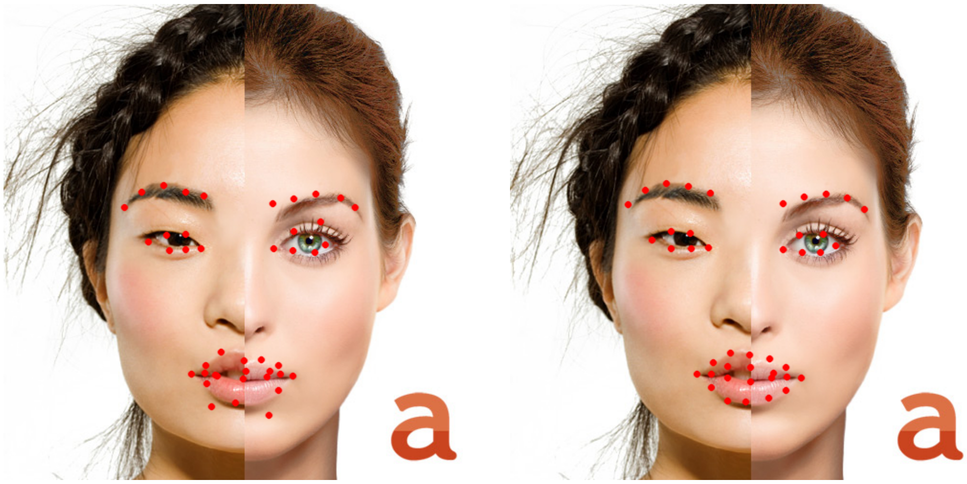
(b) CLM [53].

(c) The proposed approach.

Figure 3.13: Qualitative comparison in terms of generalization power and localization accuracy.

3.5 Discussion

I have presented a new framework for pose-constrained facial feature localization. This approach is based on a new shape model, which addresses



(a) 75 random edges.

(b) 75 “most informative” edges.

Figure 3.14: Importance of the graph topology on localization accuracy.

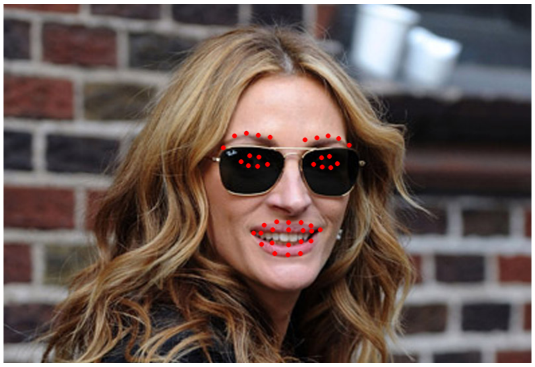


Figure 3.15: Implicit occlusion handling example (best viewed in color and high-resolution).

the limitations of the existing methods. My model “learns” which shape constraints are inherently “more important” and captures them, while omitting the “not-so-important” constraints. This way, it is *necessarily and sufficiently* more strict than the existing part-based models, while still being a lot more flexible than the existing parameterized models.

Prior shape knowledge, prior appearance knowledge, and the image observations, are incorporated within a graphical model. Within this framework, facial feature localization (i.e. probabilistic inference in the graph) is achieved through nonparametric belief propagation.

I have validated my method through qualitative and quantitative experiments and comparisons with the state-of-the-art.

Chapter 4

Unconstrained Facial Feature Localization

4.1 Motivation

In Chapter 3, I introduce a pose-constrained feature localization method that is based on a *highly flexible, yet sufficiently strict* shape model. In this chapter, I extend this pose-constrained method to unconstrained setting by introducing a multi-model approach. While doing so, I address two key limitations of the existing multi-model methods:

1. Semantically and manually defining the models, and

2. Not having efficient and effective model selection strategies.

Unlike any existing multi-model method, my approach uses unsupervised clustering on a large training set for *automatically* learning a large number of pose-constrained models. Furthermore I complement this multi-model approach with an effective model selection strategy to be used in testing.

This way, my method can have many more models, each with a higher level of expressive power. Consequently, it is a more effective partitioning of the face image space.

4.2 Background

In the following subsections I provide the required technical background. First, I discuss Principal Component Analysis, which is a widely used dimensionality reduction technique, based on finding a subspace that captures the most variation in the training data. Then, I discuss a popular data clustering technique called "k-means".

4.2.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA), is a dimensionality reduction technique that is widely used in a variety of applications. It is originally introduced by Pearson [49] in 1901 as "finding a line (or plane) which will be the 'best fit' to a system of points". In other words, PCA seeks for the linear projection which minimizes the projection error, defined as the mean squared distance between the data points and their projections [5].

Following the notation of Bishop [5], we assume a complete orthonormal set of D-dimensional basis vectors $\{u_i\}$ where $i = 1, \ldots, D$. Any data point in this space may be represented as a linear combination of the basis vectors:

$$x_n = \sum_{i=1}^{D} \alpha_{n,i} u_i \tag{4.1}$$

where $\alpha_{n,j} = x_n^T u_j$ is the coefficient of the data point along the direction of the basis u_j. Using this relationship, we can also represent the same data point by:

$$x_n = \sum_{i=1}^{D} (x_n^T u_i) u_i \tag{4.2}$$

This data point may be approximated in a lower dimensional subspace using:

$$\tilde{x}_n = \sum_{i=1}^{M} z_{n,i} u_i + \sum_{i=M+1}^{D} b_i u_i \tag{4.3}$$

where $M < D$ is the dimensionality of the subspace and $\{b_i\}$ are constants that are the same for all data points.

PCA tries to find the "best" such subspace, which would result in minimum average approximation error:

$$J = \frac{1}{N} \sum_{n=1}^{N} ||x_n - \tilde{x}_n||^2 \tag{4.4}$$

By substituting for $\tilde{x}_n$, and setting the derivative of J with respect to $z_{n,j}$ to zero, we obtain:

$$z_{n,j} = x_n^T u_j \tag{4.5}$$

for $j = 1, \ldots, M$. Similarly, by substituting for $\tilde{x}_n$, and setting the derivative of J with respect to b_j to zero, gives us:

$$b_j = \bar{x}^T u_j \tag{4.6}$$

for $j = M + 1, \ldots, D$, where $\bar{x}$ is the mean of the dataset.

If we substitute for $z_{n,j}$ and b_j, and use the relation in Equation 4.2, we obtain:

$$x_n - \tilde{x}_n = \sum_{i=M+1}^{D} \{(x_n - \bar{x})^T u_i\} u_i \tag{4.7}$$

and the cost function becomes:

$$J = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=M_1}^{D} (x_n^T u_i - \bar{x}^T u_i)^2 = \sum_{i=M+1}^{D} u_i^T S u_i \tag{4.8}$$

where S is the data covariance matrix defined by:

$$S = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^T \tag{4.9}$$

The general solution of this minimization problem is obtained by choosing the $\{u_i\}$ to be the eigenvectors of the covariance matrix S where $i = 1, \ldots, D$. Equation 4.8 is minimized when the selected eigenvectors are the ones corresponding to the $D - M$ smallest eigenvalues. *Hence, the eigenvectors corresponding to the M largest eigenvalues form the M-dimensional subspace, which we are interested in.* The principal components of a sample two-dimensional dataset is illustrated in Figure 4.1.

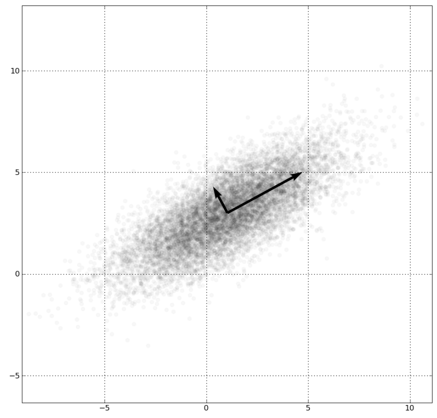


Image Credit: Ben FrantzDale.

Figure 4.1: Principal components of a sample two-dimensional dataset.

4.2.2 k-means Clustering

The k-means algorithm is a clustering algorithm for assigning a set of D-dimensional data points $\{x_n\}_{n=1}^{N}$ into $K \leq N$ clusters. It is an example of a "competitive learning algorithm", where K clusters compete with each other for the ownership of the data points [43].

The objective of the k-means algorithm is to partition N data points into K clusters $S = \{S_1, \ldots, S_K\}$, such that the overall sum-squared assignment error is minimized:

$$\arg\min_{S} \sum_{k=1}^{K} \sum_{x_j \in S_k} ||x_j - \mu_k||^2 \tag{4.10}$$

where $\mu_k \in \mathbb{R}^D$ is the mean of the data points in S_k. Note that here "the assignment error" is defined as the Euclidean distance between a data point and its cluster mean.

The most common k-means algorithm uses an iterative refinement technique:

Initialization - Set $\{\mu_k\}_{k=1}^{K}$ to random values.

Assignment Step - At each iteration t, assign each data point to the closest cluster:

$$S_k^{(t)} = \{x_j \mid ||x_j - \mu_k^{(t)}||^2 \leq ||x_j - \mu_p^{(t)}||^2 \;\; \forall\, p, 1 \leq p \leq K, p \neq k\} \tag{4.11}$$

Update Step - Calculate the new means to be the centroids of the clusters:

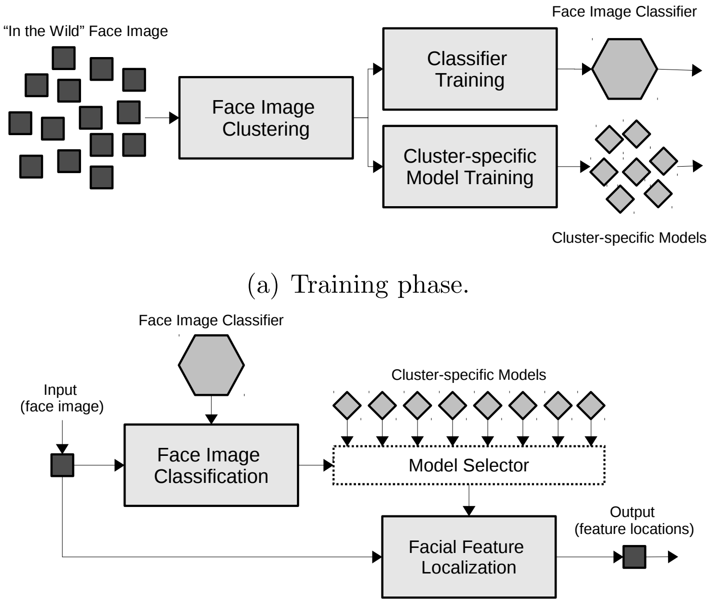
$$\mu_k^{(t+1)} = \frac{1}{|S_k^{(t)}|} \sum_{x_j \in S_j^{(t)}} x_j \tag{4.12}$$

Repeat - Until the assignments do not change.

The "hierarchical k-means" algorithm is a recursive implementation of the above algorithm where the dataset is recursively partitioned into two groups until either a stopping criteria is reached, or every cluster contains only a single element.

4.3 Approach

In this section, I first give an overview of my method and then discuss each of its components in more detail.



(a) Training phase.

(b) Testing phase.

Figure 4.2: System overview.

Figure 4.2 illustrates the system overview of my algorithm. In the training phase, I first partition the face image space and then train a cluster classifier, and a set of cluster-specific shape and appearance models.

In the testing phase, I first determine the correct cluster of the given image and then use the corresponding shape and appearance models for feature localization. These precise models provide a better prior knowledge on relative landmark locations and their appearances.

4.3.1 Face Image Clustering

As Figure 4.2 illustrates, my approach is built on partitioning the space of all face images into a number of clusters, and then training cluster-specific models for each of these partitions.

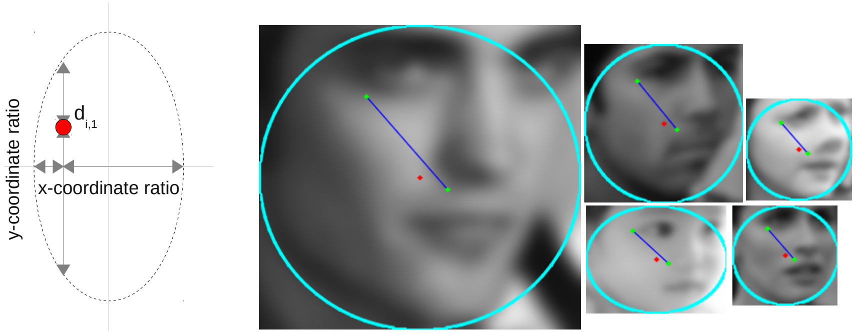
For face image clustering I first form a training set consisting of thousands of annotated “in the wild” face images. Next, I centralize and scale normalize all the shapes in the training set and perform a principal component analysis (PCA). Even though the common practice is to do a Generalized Procrustes Analysis [21] before PCA, note that I omit the “rotation” component. This way, the principal components capture both the “head-pose variations” and the variations due to non-rigid deformations (e.g. expressions).

I then perform a hierarchical k-means clustering using only the top 8 principal components (capturing 88% of the variation). Since, changes in the head-pose result in very significant variations in landmark locations, these top principal components are expected to capture mostly the head-pose changes (3-degrees of freedom). However, in practice, they do also capture the expression variations.

4.3.2 Face Image Classification

The face image space partitioning explained in Section 4.3.1 is done solely based on ground truth landmark locations. However, with an unseen test image, one does not have this information, and needs to develop a mechanism for determining the cluster of the image using only the image features. This mechanism needs to be very efficient and fairly accurate.

Note that for face image classification it is assumed that the bounding box of the face within the image is available, i.e. the face is detected by a face detector. This is a common assumption between the feature localization methods.



(a) (b)

Figure 4.3: An example binary feature. Note that both pixel locations are defined as x- and y-coordinate ratios. This way pixel locations are consistent across face images of different aspect ratios and scales.

For representing the face images I use very simple binary features:

$$f_i = \begin{cases} 1, & \text{if } I(d_{i,1}) < I(d_{i,2}) \\ 0, & \text{if } I(d_{i,1}) \geq I(d_{i,2}) \end{cases} \tag{4.13}$$

where $I(d_{i,1})$ represents the pixel intensity of image I at pixel location $d_{i,1}$. In order to make these simple features robust against image noise, I apply a Gaussian filter to the images before computing them.

$d_{i,1}$ and $d_{i,2}$ of each binary feature is selected *randomly* as x- and y-coordinate *ratios* within the *ellipsis inscribed by the bounding box.* As illustrated in Figure 4.3, this way selected pixel locations are consistent across face images of different aspect ratios and scales.

Given these features, the classification may be formulated in a Bayesian way as in [47]. Let $f_i, i = 1, \ldots, N$ be the set of binary features, and $c_j, j = 1, \ldots, M$ be the set of clusters. Then the cluster assignment is done as:

$$c^* = \arg\max_{cj} P(C = c_j | f_1, \ldots, f_N) \tag{4.14}$$

Using Bayes' theorem and assuming uniform cluster priors, Equation 4.14 reduces to:

$$c^* = \arg\max_{cj} P(f_1, \ldots, f_N | C = c_j) \tag{4.15}$$

With hundreds of binary features and dozens of clusters complete representation of Equation 4.15 becomes infeasible. [47] introduced *ferns* to address this problem. Each fern is a set of randomly assigned binary features: $F_k = \{f_{\sigma(k,1)}, \ldots, f_{\sigma(k,S)}\}$, where S is the number of features per fern, and $\sigma(k, s)$ is a random permutation function with range $1, \ldots, N$.

Assuming ferns are independent, the cluster assignment equation becomes:

$$c^* = \arg\max_{cj} \prod_{k=1}^{K} P(F_k|C = c_j) \tag{4.16}$$

where $P(F_k|C = c_j)$ is computed in the training as an occurrence frequency.

Ferns usually are used for feature matching, where misclassification tolerance is fairly high. However, it is demonstrated in Section 4.4 that they perform exceptionally well in this difficult classification problem despite their extreme simplicity.

4.3.3 Cluster-specific Model Fitting

Cluster-specific shape constraints are modeled using the part-based approach of Chapter 3. Note that any of the simple models mentioned in Section 2 could have been employed. However, I have already discussed in Chapter 3 that my "learned" model is "highly flexible, yet sufficiently strict". These properties are particularly useful when one have a significant number of clusters, and some clusters have relatively few number of training examples.

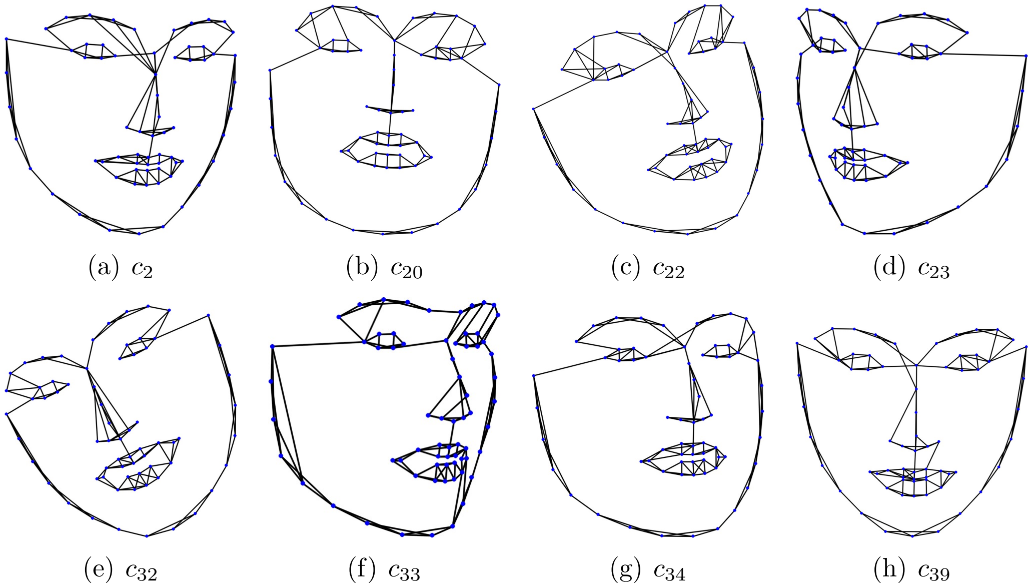
For each cluster, I define a probabilistic graphical model $\mathcal{G}_j = (\mathcal{V}_j, \mathcal{E}_j)$. Nodes in $\mathcal{V}_j$ represent landmark locations and image observations as hidden and observed random variables, respectively. The graph topology is defined by $\mathcal{E}_j$.

Pairwise spatial relationships between the landmarks are modeled using anisotropic Gaussians:

$$\psi^j_{s,t}(x^j_s, x^j_t) = \mathcal{N}((x^j_s - x^j_t); \mu^j_{s,t}, \Sigma^j_{s,t}) \tag{4.17}$$

where $x_s^j \in \mathcal{V}_j$ represents the s^{th} landmark in j^{th} cluster. Relationship parameters $\mu_{s,t}^j$ and $\Sigma_{s,t}^j$ are computed after all cluster members are aligned using Generalized Procrustes Analysis [21].

The graph topology is then learned from training data. Each node x_s^j is connected to its k most "informative" (i.e. smaller $\Sigma_{s,t}^j$) neighbors. This allows me to train a number of shape models, all with possibly different topologies and pairwise relationships (see Figure 4.4).


(a) c_2 (b) c_{20} (c) c_{22} (d) c_{23}

(e) c_{32} (f) c_{33} (g) c_{34} (h) c_{39}

Figure 4.4: Cluster-specific learned shape models. Note that some edges are enforced in all topologies to ensure that the graphs are connected. These edges are determined experimentally (there are 5 of them).

For modeling the image observations I employed the local approach explained in [53]. Cluster-specific "local experts" are trained using shape normalized images. Obtained response maps are then approximated using isotropic Gaussian Kernel Density Estimates [54] and incorporated into the shape model as “observation potentials”:

$$\psi_s^j(x_s^j, y_s^j) = \sum_{z_i \in \Psi_s^j} \pi_s^j(z_i)\, \mathcal{N}(x_s^j; z_i, \rho I) \tag{4.18}$$

where Ψ_s^j denotes the set of integer pixel locations within a square region centered at x_s^j, ρ is the bandwidth of the kernels, and $\pi_s^j(z_i)$ is the probabilistic response of the s^{th} expert of j^{th} cluster at location z_i.

With these cluster-specific models, landmark localization becomes inference on the corresponding graphs. This is achieved by nonparametric belief propagation [56].

4.4 Results

I validate and evaluate the primary components of my proposed approach separately in the following sections:

4.4.1 Face Image Clustering

I formed my face image space by combining multiple publicly available “in the wild” face image datasets: Labeled Face Parts in the Wild (LFPW) [3], Annotated Faces in the Wild (AFW) [69], Helen Facial Feature Dataset [39], and IBUG training set [51]. This combined set has a total of 3837 annotated [51] face images.

I partitioned this space into $M = 40$ clusters, using hierarchical k-means clustering. The average cardinality of the resulting clusters was 95.92, with only 3 clusters having less than 25 elements. Some cluster examples are provided in Figure 4.4.

As the dendrogram in Figure 4.5 illustrates, some clusters are much more similar to each other than the others. This important property is further discussed in the following sections.

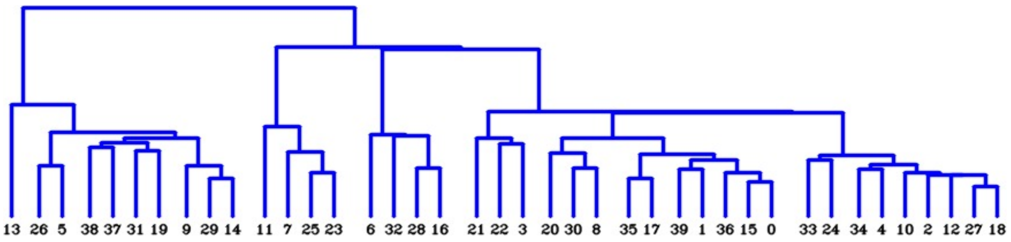


Figure 4.5: Cluster dendrogram.

4.4.2 Testing Set Analysis

For image classification and model fitting (i.e. feature localization) experiments I formed 4 separate test sets, one for each dataset used in the face space partitioning. For IBUG, this testing set included all 135 IBUG images. For Helen and LFPW, the testing sets included the provided “testsets” (330 and 224 images, respectively). And for AFW, I randomly selected 107 images as the test set. In each experiment, the training set consisted of all the *other* images in the combined dataset.

Analyzing the image distributions of these test sets, and comparing them to the image distribution of the combined dataset revealed interesting results as presented in Figure 4.6. Helen and LFPW datasets form a large portion of the combined dataset, and have similar image distributions, with their "testsets" chosen seemingly without much bias. AFW and IBUG, on the other hand, have significantly different image distributions, making corresponding experiments fairly difficult.

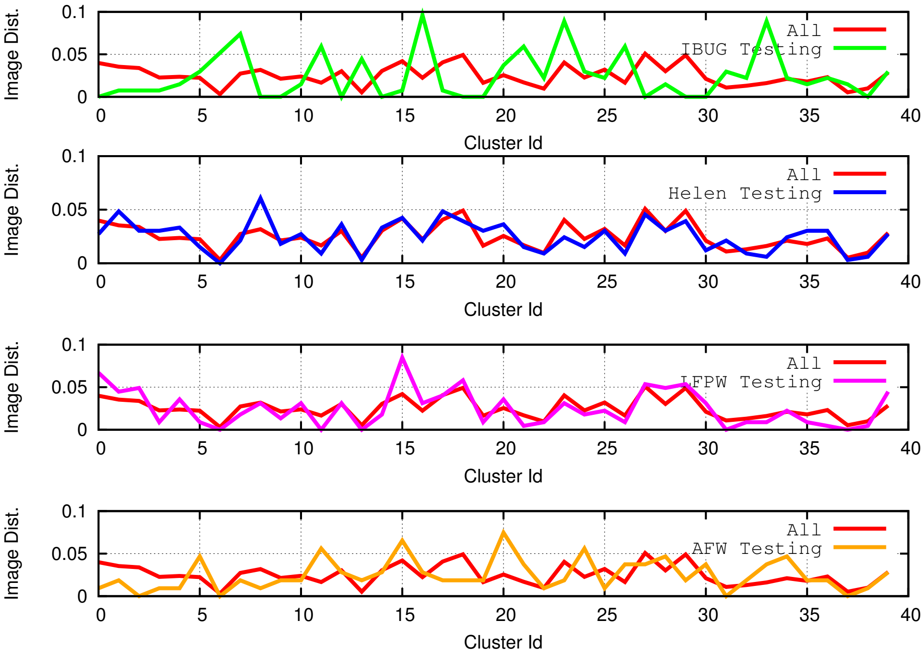


Figure 4.6: Element distributions of the test sets compared to the combined dataset.

4.4.3 Face Image Classification

For each experiment, a Naive Bayesian cluster classifier is trained using the parameters in Table 4.1. These parameters are determined empirically.

Note that for both training and testing of the classifiers, I have used

Parameter	Description	Value
M	# of Clusters	40
N	# of B. Features	5000
K	# of Ferns	1000
S	# of B. Features per Fern	5

Table 4.1: Face image classification parameters.

the face detection bounding boxes provided by [51].

Figure 4.7 presents the Rank-n accuracies achieved in each experiment. As the figure illustrates, Helen and LFPW classifiers perform much better than the AFW and IBUG classifiers. This observation is in-line with my discussion regarding the image distributions of each test set.

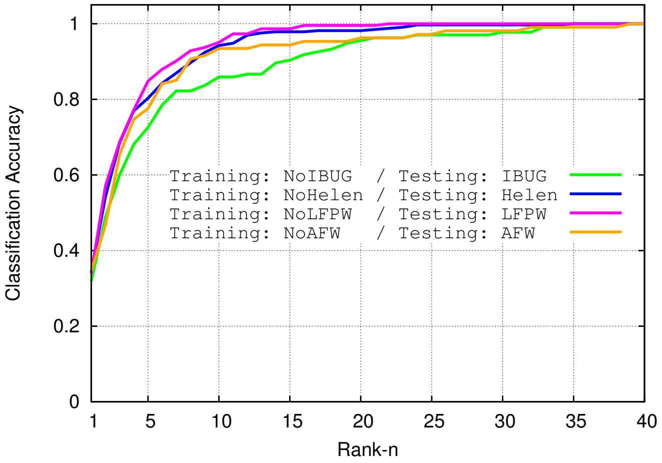


Figure 4.7: Rank-n classification accuracies.

For Helen and LFPW, Rank-1 accuracies are around 34%, rapidly increasing to about 68% in Rank-3, and reaching to more than 80% by Rank-5.

For IBUG, Rank-1 accuracy is 31.85%, hitting to 60% in Rank-3, and reaching to about 73% by Rank-5. These classification accuracies are in fact very good, considering that: 1) this is an intra-class classification problem, which is much harder than common inter-class classification problems, and 2) this is a 40-class classification problem and random assignment would give a mere 2.5% accuracy.

I further analyze the misclassifications in the IBUG experiments in Figure 4.8. Note that, the axes of this confusion matrix are re-ordered according to the "cluster similarities" presented in Figure 4.5. Hence, similar clusters are closer to each other in the figure. As the figure illustrates, a significant portion of the misclassifications in the IBUG experiments are located closer to the primary diagonal. These misclassifications may be recoverable during model fitting, since a misclassification in this step merely means that the prior shape and appearance knowledge used during model fitting will not be optimal. This is further discussed in the next section.

4.4.4 Facial Feature Localization in the Wild

Facial feature localization experiments are performed on four test sets explained in Section 4.4.2. I compare my method with two of the most recently published work in the field: Discriminative Response Map Fitting (DRMF) by Asthana et al. [1], and Supervised Descend Method (SDM) by Xiong and De la Torre [66]. For each method, I have used the implementations provided by the authors.

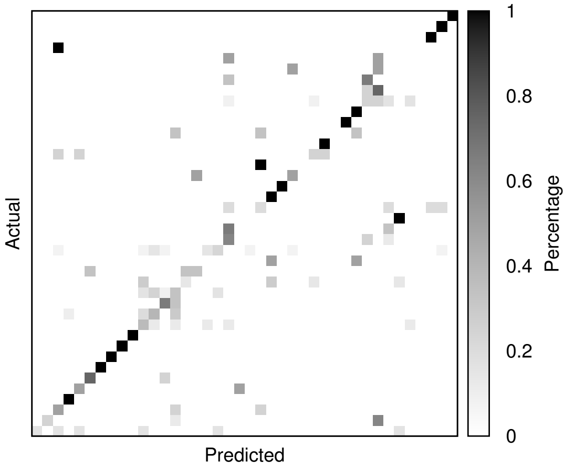


Figure 4.8: IBUG experiment confusion matrix.

All methods are initialized with the face detector bounding boxes provided by [51]. These bounding boxes are based on the "Mixture-of-Trees" face detector of [69]. For SDM, I have adjusted the bounding boxes as suggested by the authors. I found: $offset = -0.1 * bb.\{width|height\}$, to be a good offset between the "Mixture-of-Tree" face detector and the OpenCV face detector, where *bb.width* and *bb.height* are the width and the height of the bounding box, respectively. Note that this offset is applied to the top-left corner of the bounding box. For DRMF I did not perform any adjustments since the original implementation provided by the authors is based on the "Mixture-of-Tree" face detector as well.

For DNC, I have initialized the model fitting with the top-5 models given by the corresponding cluster classifier. Note that this is a common strategy among multi-model methods (eg. [15, 69]), and pose-incorporated methods (eg. [2, 40]). If at some point the fitting score (computed using the expert responses) is above a threshold the model fitting is terminated without trying the rest of the models.

DRMF, SDM, and DNC, all detect a different number of landmarks: 66, 49, and 68, respectively. Figure 4.9 illustrates the 68 facial features detected by DNC. For a fair comparison I have only considered the 49 landmarks used by SDM in the localization experiments. All the localization errors are normalized with respect to the corresponding interocular distance of the face.

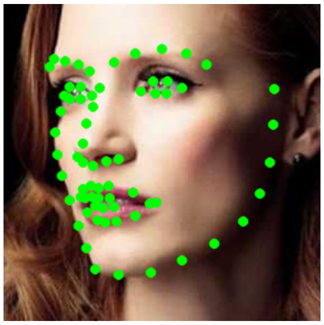


Figure 4.9: 68 facial features that are used in the localization experiments.

In Figures 4.10-4.13, I present the cumulative error distributions of DRMF, SDM, and DNC, on four testing sets. As discussed in Section 4.4.2, Helen and LFPW test set image distributions are more similar to the image distribution of the combined dataset. This, in a way, suggests that they are relatively easier test sets when compared to AFW and IBUG. As illustrated in Figures 4.10-4.13, all three methods do better in Helen and LFPW test sets.

As discussed in Section 4.4.3, the *optimal*, or *the real*, cluster of a given

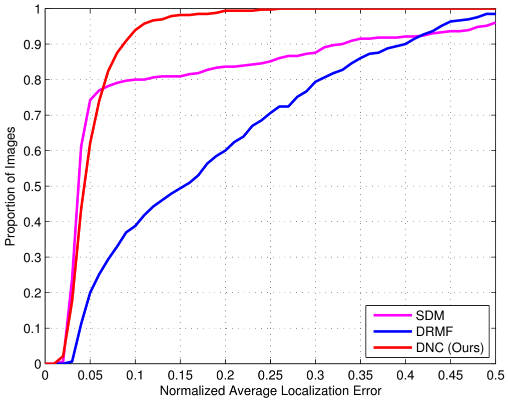


Figure 4.10: Cumulative error distributions for HELEN.

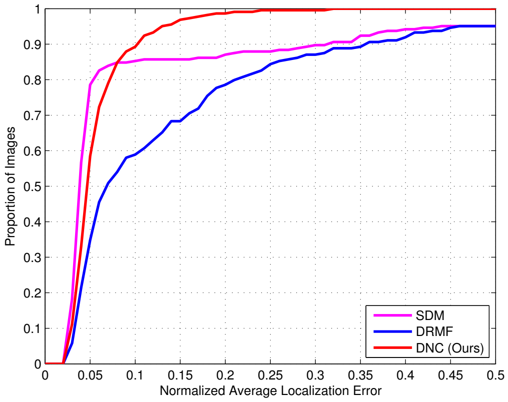


Figure 4.11: Cumulative error distributions for LFPW.

test image is expected to be in the top-5 clusters of the classifier about 80% of the time. Moreover, a misclassification in this case, does not necessarily mean a total failure in localization. It merely means that the model fitting will be performed with a *not-so-optimal* prior knowledge.

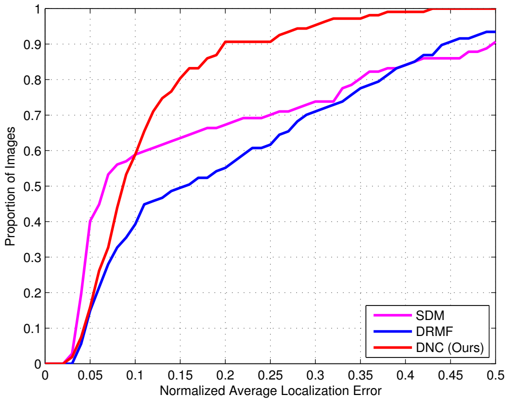


Figure 4.12: Cumulative error distributions for AFW.

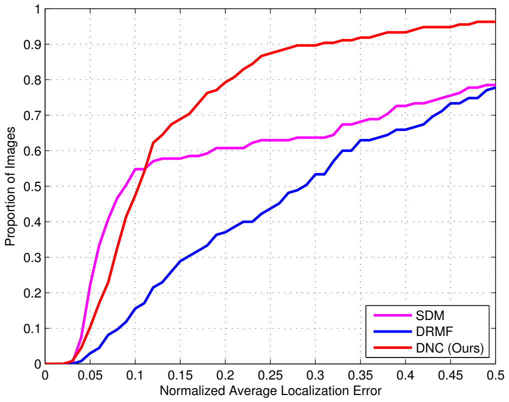


Figure 4.13: Cumulative error distributions for IBUG.

Even though SDM does extremely well for a portion of the test images, overall DNC clearly outperforms the two other methods. This can be explained by two properties: 1) DNC models the prior shape and appearance knowledge more precisely, and 2) DNC incorporates a very efficient and effective model selection strategy.

The qualitative results on the test sets are presented in Figures 4.14 - 4.17. The two-step nature of my approach has clear advantages in “in the wild” feature localization. By first classifying the face image I effectively improve both the initialization *and* the fitting accuracy.

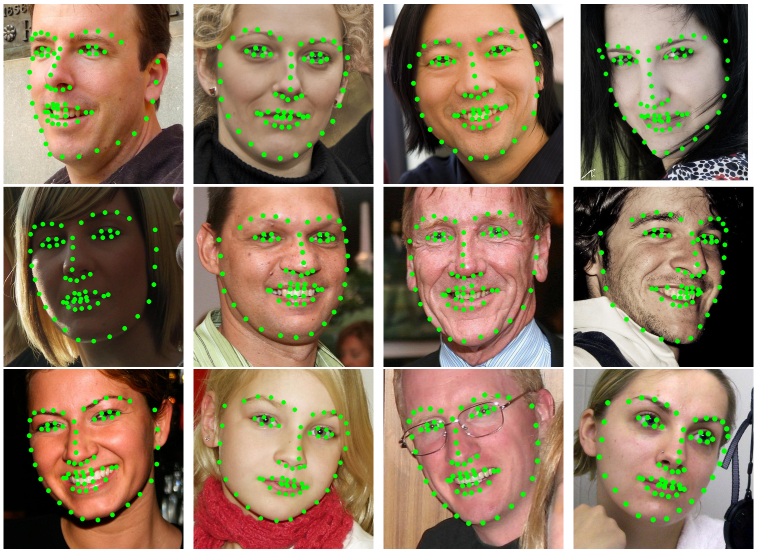


Figure 4.14: Qualitative results on Helen.

I further analyze the effects of cluster misclassification on feature localization. As mentioned in Section 4.4.3, face image misclassifications are inevitable during feature localization. However, a misclassification in the first step does not necessarily mean that the feature localization in the second step will fail. Flexibility and generalization power are two of the primary advantages of the cluster-specific shape models I employ in this work. Figure 4.18 illustrates an example. In this figure the same input image is initialized and

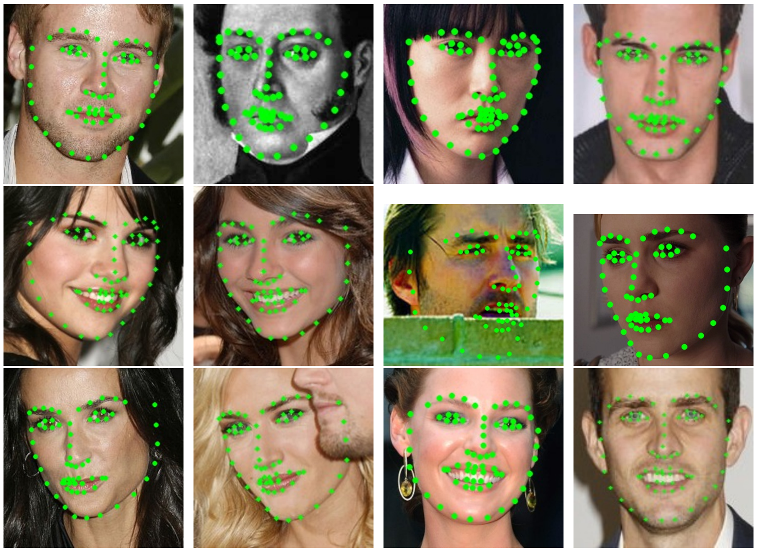


Figure 4.15: Qualitative results on LFPW.

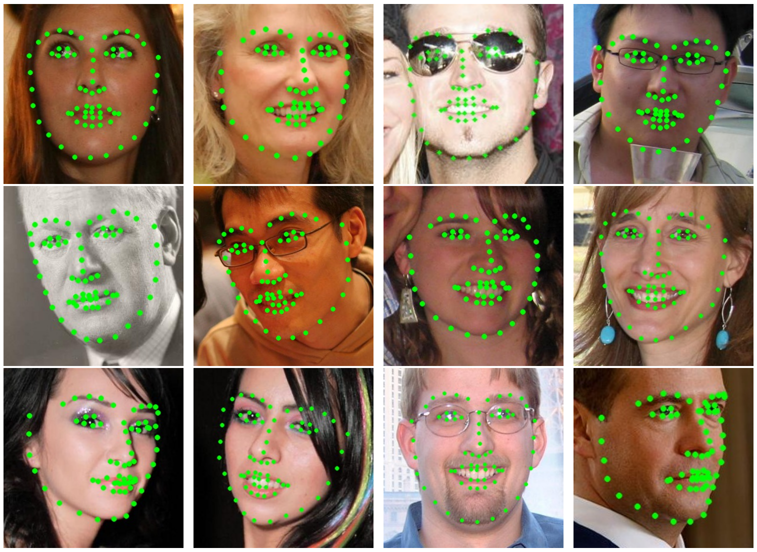


Figure 4.16: Qualitative results on AFW.

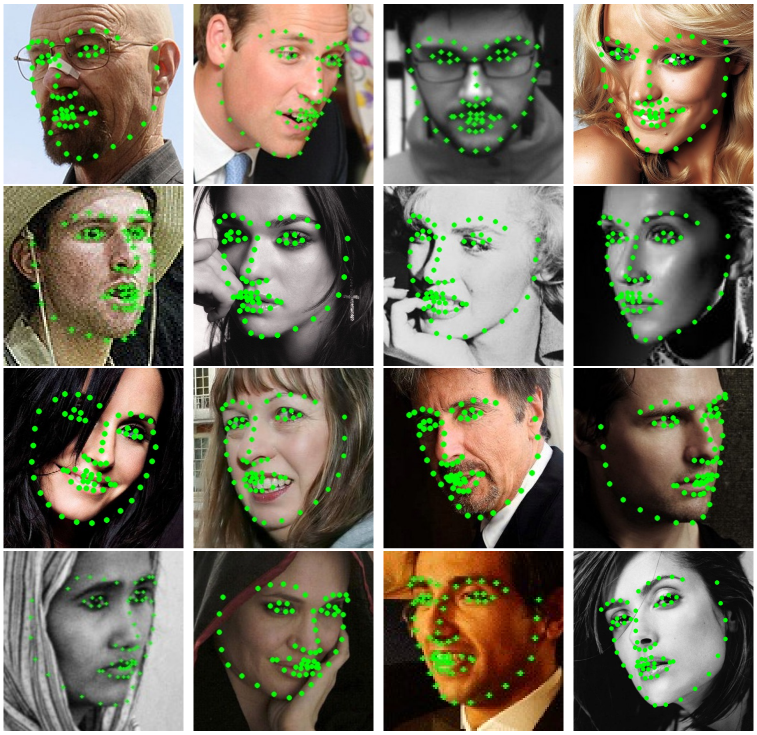


Figure 4.17: Qualitative results on IBUG.

fitted using two different, albeit “similar” models.

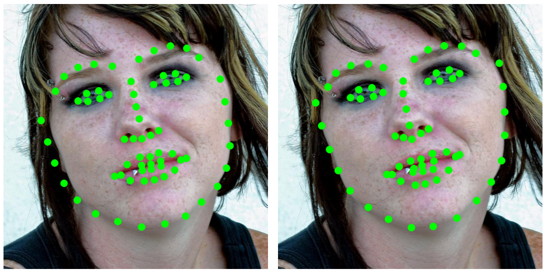


Figure 4.18: Effects of misclassification on feature localization.

4.5 Discussion

I have presented a new approach for unconstrained facial feature localization. My approach addresses the two key limitations of the existing multi-model methods, which are: 1) semantically and manually defining the models, and 2) not having efficient and effective model selection strategies.

Unlike existing methods, I do not define the models semantically and manually. Instead I generate them by first performing an unsupervised clustering on a very large in-the-wild dataset, and then training cluster-specific shape and appearance models. This way, my method can have many more models, each with a higher level of expressive power.

I have also introduced a new model selection (i.e. face image classification) technique, which is based on extremely efficient binary features and a simply naïve Bayesian classifier. In despite of its simplicity, this model selection technique is shown to perform well in challenging experiments.

I have validated the effectiveness of each component of my approach with extensive experiments. Furthermore, I have showed that the proposed two-step approach outperforms the state-of-the-art in terms of localization accuracy in unconstrained feature localization.

One exciting future work involves incorporating a coarse-to-fine model selection and fitting strategy using the built hierarchical cluster structure.

Chapter 5

Facial Feature Localization in Videos

5.1 Motivation

This chapter focuses on *facial feature localization in unconstrained videos*, which is a particular application of the techniques introduced in the previous chapters. In a way, it *further* tests and validates the two primary claims which are made (and already proved) in the previous chapters:

1. Compared to the existing multi-model methods, I claim that my approach covers the face image space more effectively. While all existing multi-model methods define their models semantically and manually, I use unsupervised clustering for generating them. Processing an unconstrained video usually requires a number of these models.

2. What makes this application even more challenging is the fact that each video covers a *continuous* subspace within the face image space. Multi-model approaches, on the other hand, are "quantizations" of this space. Consequently, regions that are close to the partition centers are modeled well, whereas the intermediate regions are *not* covered so well. I claim that my pose-constrained models are flexible-enough to cover these regions.

In order to improve the localization results within a window of frames, I use a simple heuristic: head movements are continuous functions of time. Consequently, each frame in a face video corresponds to a discrete sampling of the underlying head movement. This observation is valuable for detecting and fixing the feature localization failures as explained in the following sections.

Note that the concepts introduced in this chapter are supported by some simple examples and experiments. The goal here is to provide a "proof-of-concept" rather than extensive evaluation. The above claims, and the related techniques, have already been extensively tested and validated in the previous two chapters.

5.2 Background

In the following subsection I provide the required technical background for this chapter.

5.2.1 Orthographic Projection

A perspective camera is usually modeled as a projective mapping from three-dimensional scene to two-dimensional image plane. This mapping may be represented by a 3×4, rank-3 matrix P. The *central projection equation* is then given by:

$$\boldsymbol{x} = P\boldsymbol{X} \tag{5.1}$$

where $\boldsymbol{X} = [X \ Y \ Z \ 1]^T$ is the scene point in homogeneous coordinates and $\boldsymbol{x} = [x \, y \, 1]^T$ is the corresponding image point in homogeneous coordinates.

The matrix P may be decomposed as $P = K[R|\boldsymbol{t}]$. In this decomposition, R is a rotation matrix and $\boldsymbol{t}$ is a translation vector. These two define the location and the orientation of the camera with respect to an absolute coordinate frame. K, on the other hand, is called the *calibration matrix* and encodes the intrinsic parameters of the camera:

$$k = \begin{pmatrix} \gamma f & s & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.2}$$

Here, f is the focal length and γ is the aspect ratio. The principal point is $(u_0, v_0)^T$ and s is the skew parameter.

Orthographic projection is possibly one of the simplest such projective mappings. It is a form of *parallel projection*, where all the projection lines are orthogonal to the projection plane as illustrated in Figure 5.1. Even though, orthographic projection makes fairly crude assumptions, it is still useful in situations where the scene distance (w.r.t. the camera) is much larger than the scene depth. This is in fact true for most unconstrained face images and videos.

In homogeneous coordinates, the orthographic projection may be represented as:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.3}$$

Note that it simply eliminates the z-dimension of the scene-point.

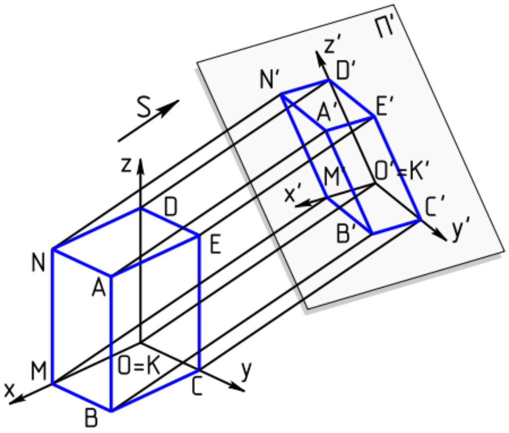


Image Credit: Yuri Raysper.

Figure 5.1: Orthographic projection.

5.3 Approach

Most still-image facial feature localization methods are extended to work with videos by enforcing motion continuity in the image domain. Usually, this simply involves initializing the models in one frame, with the results obtained in the previous frame. *My method, on the other hand, is based on enforcing the motion continuity in the real-world domain.*

The flowchart of my approach is presented in Figure 5.2. First, I use my unconstrained localization method, described in Chapter 4, for detecting the landmarks in each frame. Next, I estimate the head-pose in each frame using the two-dimensional localization results. I then estimate the actual three-dimensional head-movement and finally use this information for detecting and fixing the localization failures. I explain each of these latter three steps in

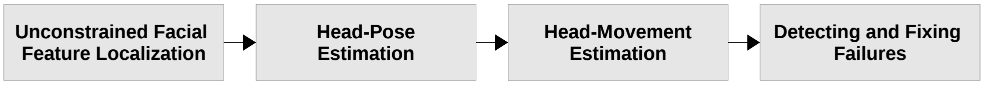

Unconstrained Facial Feature Localization
Head-Pose Estimation
Head-Movement Estimation
Detecting and Fixing Failures

Figure 5.2: Facial feature localization in unconstrained videos.

Variable	Dimensions	Definition
N	$\mathbb{R}$	Number of features.
x	$\mathbb{R}^{2\times N}$	Two-dimensional shape.
X	$\mathbb{R}^{3\times N}$	Three-dimensional generic model.
R	$\mathbb{R}^{3\times 3}$	Rotation (three-dimensional).
t	$\mathbb{R}^{3\times 1}$	Translation (three-dimensional).
s	$\mathbb{R}$	Scale parameter.
P	$\mathbb{R}^{2\times 3}$	Orthographic projection (Euclidean coordinates).

Table 5.1: Defined variables and their dimensions.

more detail in the following subsections.

5.3.1 Head-Pose Estimation

Head-pose estimation is the problem of computing the three-dimensional head orientation, given a two-dimensional image. There are two assumptions: 1) geometric configuration of the features on the face is known, and 2) these features can be located and matched in an image. Here, the first assumption corresponds to having *a generic three-dimensional face model*, and the second assumption corresponds to having *the two-dimensional feature localization results.*

Given the variables in 5.1, the head-pose estimation problem may be formulated as a minimization problem. A valid objective function is the sum-squared Euclidean distances between the coordinates of the given two-dimensional shape, and the rotated, projected generic three-dimensional model:

$$\{s^*, R^*, t^*\} = \underset{s,R,t}{\arg\min} ||x - P[s(RX - t)]||^2 \tag{5.4}$$

Assuming both the two-dimensional shape and the three-dimensional generic model are centralized, and defining $R' = sPR$, the above formulation simplifies to:

$$R'^* = \underset{R'}{\arg\min} ||x - R'X||^2 \tag{5.5}$$

where $R' \in \mathbb{R}^{2\times3}$. This is a simple least-squared problem, with six variables, and N equations.

Note that the rotation matrix is orthonormal and right-handed. Hence:

1. We need an additional constraint to enforce that the two rows of the R'^* are orthogonal:

   $$< R'_1, R'_2 >= 0 \tag{5.6}$$

   where R'_i represents the i^{th} row of R'.

2. The scale factor, $s = norm(R'^*_1) = norm(R'^*_2)$.

3. The third row of R may be obtained by cross multiplying R'^*_1 and R'^*_2.

Once R is computed; yaw, pitch, and roll parameters may be determined.

5.3.2 Head-Movement Estimation

Head-movements are continuous functions of time. Estimated head-poses are nothing but samples drawn from these continuous functions at discrete time steps. Hence, the goal here is to determine the most likely functions given a set of samples (i.e. frame-by-frame head-pose estimations).

In order to ensure a "continuous" head-movement, I fit n-order polynomials to the obtained frame-by-frame head pose estimations. This process is illustrated in Figures 5.3 - 5.5.

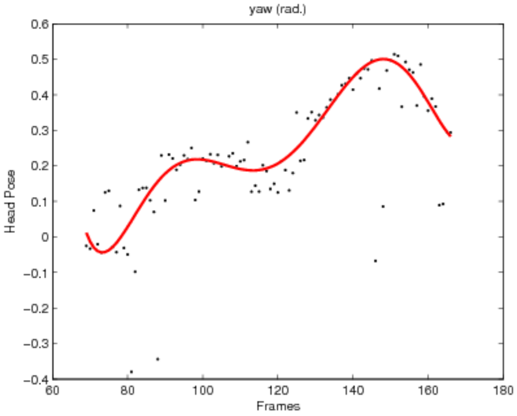


Figure 5.3: Estimated head-movement (yaw).

5.3.3 Detecting and Fixing Failures

Once the actual head-movements are estimated, detecting and fixing the localization failures become fairly straightforward. For each frame in the sequence, the first step is determining the "expected" head-pose parameters,

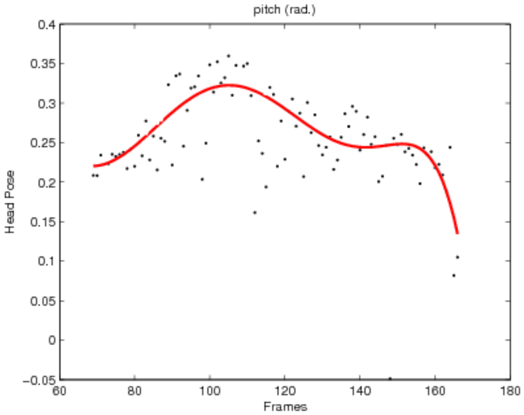


Figure 5.4: Estimated head-movement (pitch).

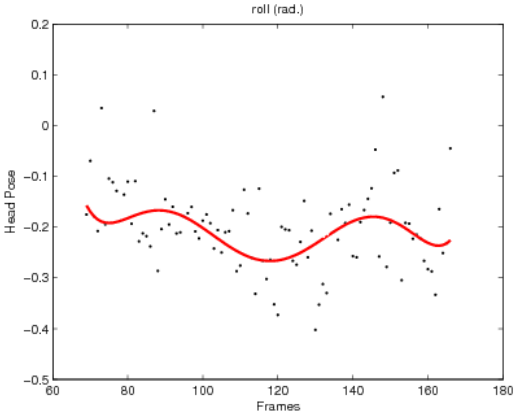


Figure 5.5: Estimated head-movement (roll).

and the corresponding "expected" shape model. This is then compared with the "computed" shape model of this frame. If these two do *not* match, then the results are considered a *failure*, and the frame is re-processed using the "expected" model. This process is illustrated in Figure 5.6.

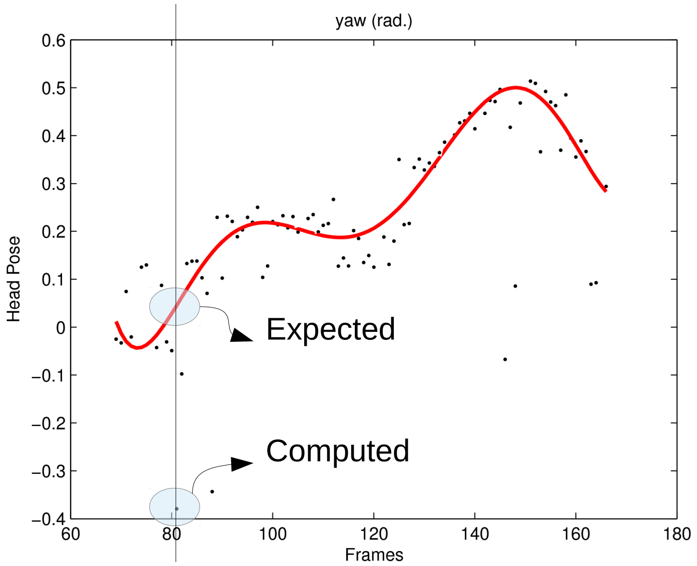


Figure 5.6: Detecting and fixing failures.

5.4 Results

5.4.1 Head-Pose Estimation

Head-pose estimation performance is tested and validated both quantitatively and qualitatively through simple experiments. The quantitative experiments are performed on synthesized data, whereas the qualitative tests are performed on real-world data.

The testing data for the quantitative experiments are generated as follows:

1. Randomly rotate the generic three-dimensional model,

2. Project it to the two-dimensional image space using orthographic projection,

3. Scale the two-dimensional shape so that it has reasonable dimensions in "pixels" (i.e. $s = 25$), and

4. Add a significant Gaussian noise ($\mu = 0$, $\sigma = 5$) to each feature location.

Note that this approach has some advantages and some disadvantages. First of all, assuming orthographic projection provides an advantage, since my head-pose estimation formulation assumes an orthographic projection as well. On the other hand, adding a significant noise to each landmark *independently*, introduces a challenge which would not be present in real localization results. That being said, since with the synthesized data one knows the precise "ground truth", it forms a nice testbed for validation.

I have tested my method with a total of $16 \times 16 \times 16 = 4096$ configurations of the form: $< \theta_{yaw}, \theta_{pitch}, \theta_{roll} >$, where $-\frac{\pi}{4} \leq \theta_{yaw}, \theta_{pitch}, \theta_{roll} \leq \frac{\pi}{4}$, with 0.1 radian increments. Each configuration is run ten times, each run having a new random Gaussian noise. The mean and the standard deviation of the pose estimation (in degrees) is presented in Table 5.2.

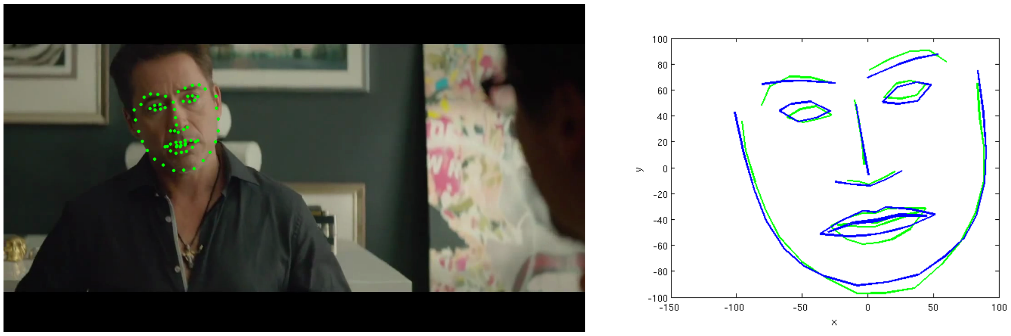
Note that the error in the pitch-angle estimates are much larger compared to the error in the yaw- and roll-angles. This is expected since, changes in yaw- and roll-angles result in much significant feature location changes, and hence are much easier to estimate. Pitch-angle changes, on the other hand,

Parameter	μ	σ
yaw	0.66296	0.19378
pitch	4.5097	3.1692
roll	0.76365	0.18554

Table 5.2: Pose estimation error (in degrees).

only result in minor changes in feature locations (when projected on the two-dimensional image plane) and hence are much more ambiguous.

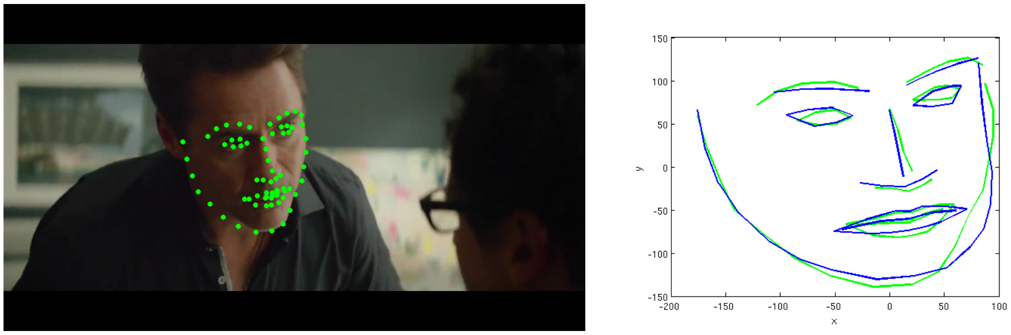
Figures 5.7-5.9 present some qualitative results. These experiments are performed on a movie trailer ("Chef"), which is download from YouTube. In these figures, plots on the right contain "centralized" and "scale-normalized" landmarks. Hence, they may look slightly distorted compared to the actual results presented on the left.



(a) Facial feature localization.

(b) Head-pose estimation.

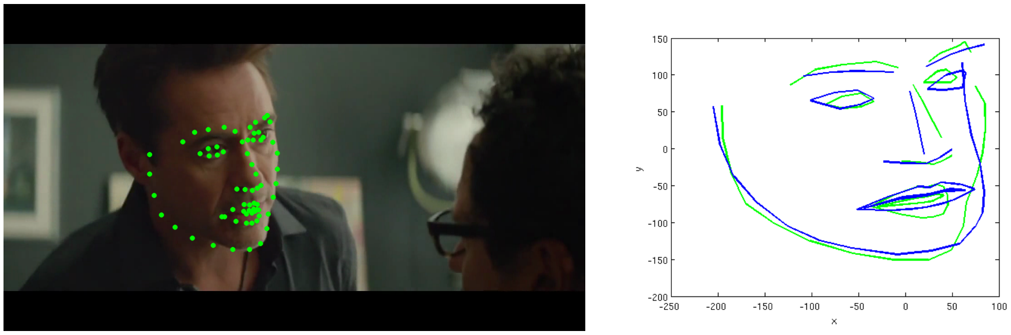
Figure 5.7: Head-pose estimation - Frame 74 (green: feature localization results, blue: aligned generic three-dimensional model).



(a) Facial feature localization.

(b) Head-pose estimation.

Figure 5.8: Head-pose estimation - Frame 115 (green: feature localization results, blue: aligned generic three-dimensional model).



(a) Facial feature localization.

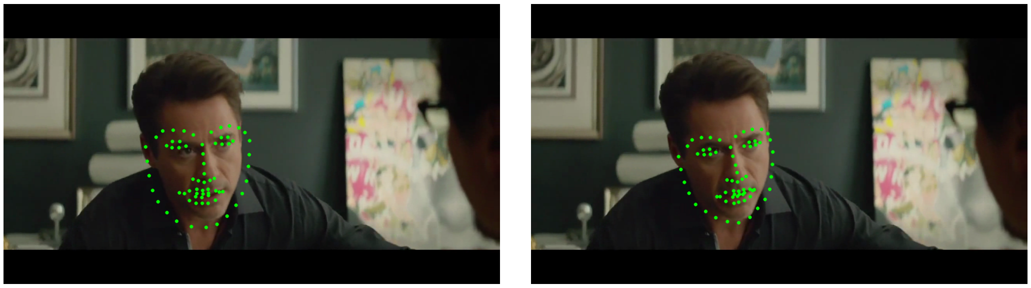
(b) Head-pose estimation.

Figure 5.9: Head-pose estimation - Frame 136 (green: feature localization results, blue: aligned generic three-dimensional model).

5.4.2 Detecting and Fixing the Failures

In these experiments I used 7-order polynomials for estimating the actual head-movements.

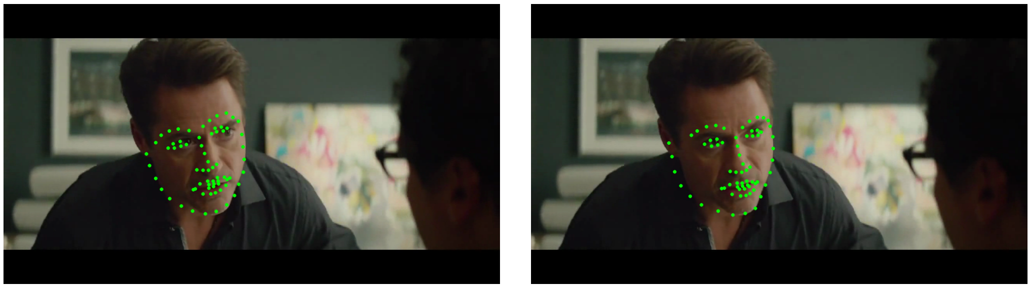
Figures 5.10 - 5.13 present some qualitative results of the method explained in Section 5.3.3. In these figures, the initial frame-by-frame results are presented on the left. Results on the right demonstrate the improvements after the corresponding failures are detected and fixed.



(a) Before detecting & fixing failures.

(b) After detecting & fixing failures.

Figure 5.10: Detecting and fixing failures - Frame 92.

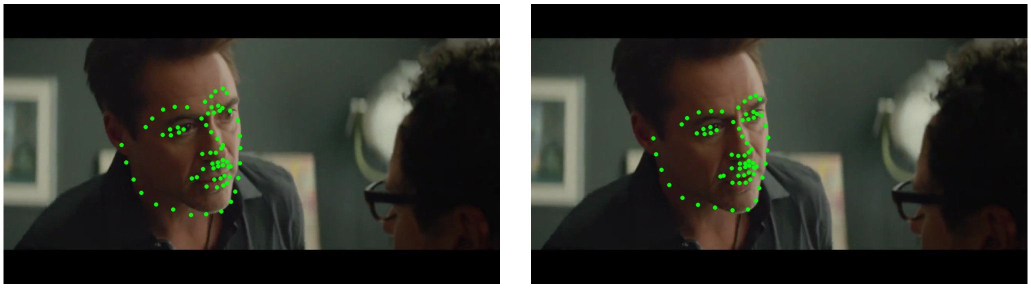


(a) Before detecting & fixing failures.

(b) After detecting & fixing failures.

Figure 5.11: Detecting and fixing failures - Frame 106.

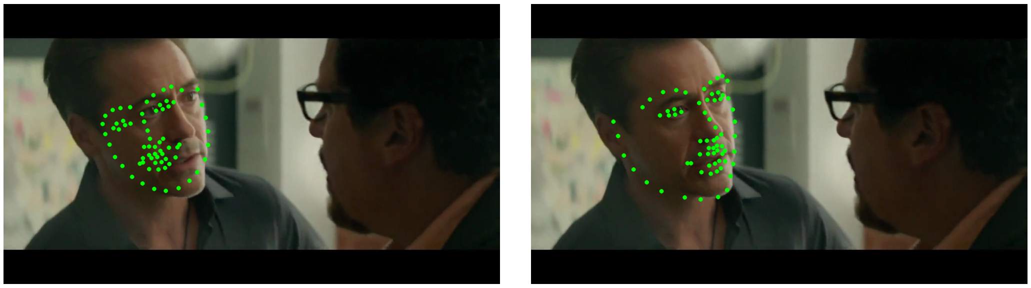
As illustrated in Figures 5.10 - 5.13, estimating the head-movement, and then using this for detecting and fixing the failures, is especially useful for identifying the *significant* errors in the frame-by-frame results. These errors are usually associated with a wrong model selection during the fitting.



(a) Before detecting & fixing failures.

(b) After detecting & fixing failures.

Figure 5.12: Detecting and fixing failures - Frame 135.



(a) Before detecting & fixing failures.

(b) After detecting & fixing failures.

Figure 5.13: Detecting and fixing failures - Frame 165.

5.5 Discussion

In this chapter, I have focused on a particular application: facial feature localization in unconstrained videos. I have presented a method for detecting and fixing the localization failures. Unlike most existing methods, I use the motion continuity in the real-world domain, rather than the motion continuity in the image domain for extending my still-image unconstrained localization method to videos. In order to so, I have also developed a clean formulation for the problem of head-pose estimation.

An exciting future work involves “predicting” the future models, using the same exact techniques I have discussed in this chapter. This way, these significant failures may be avoided all together.

Chapter 6

Conclusion

I have presented several techniques for constrained and unconstrained facial feature localization in still-images and videos. Unlike most of the existing work, which take shape modeling for granted and focus on appearance modeling, I chose to focus on the shape modeling aspect of the problem. I have shown that there are better ways for modeling the prior shape knowledge, which can make the corresponding localization algorithms more accurate and more robust.

I first introduced a "highly flexible, yet sufficiently strict" shape model, which addresses the limitations of the existing shape models. I then used this shape model within a probabilistic graphical model framework, and formulated the localization problem as a probabilistic inference on the corresponding graphical model.

Then I used unsupervised clustering to partition the face image space into a set of clusters. I trained one pose-constrained model for every partition, and used an effective model selection technique for unconstrained facial feature localization. I have shown that this is a better partitioning of the space, where each model encloses precise prior knowledge about the shape and appearance of the features of images that reside in the corresponding partition.

In the last part of the dissertation, I discussed facial feature localization in videos, which is a particular application of the previously introduced techniques. For improving the frame-by-frame localization results I first computed the head-pose in each frame, and then used this information for estimating the *actual* head-movement within a sequence of frames. The estimated head-movement is later used for detecting and fixing the localization failures.

I believe there is still a lot more to explore in terms of shape modeling. I think better shape models will provide more drastic improvements in terms of localization accuracy, robustness, and efficiency. I am particularly interested in *hierarchical* two-dimensional shape models, with more "abstract" nodes (such as "face center" or "eye center") in the coarser levels, and the actual landmark nodes in the finer levels. Such *explicit* models are yet to be fully explored.

Effective three-dimensional shape models and hybrid models that combine two-dimensional models within a three-dimensional structure are also of interest for future research. This is true not just for facial feature localization, but also for other "object"-related computer vision applications. One simple example is identity recognition. I believe the solution of this real-world problem requires models which are a lot more sophisticated than the ones we have today. Furthermore, I think "the core" of these future models will again be based on a geometric *shape model*.

Bibliography

[1] Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Robust discriminative response map fitting with constrained local models. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3444–3451, 2013.

[2] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. Constrained local neural fields for robust facial landmark detection in the wild. In *International Conference on Computer Vision and workshops*, 2013.

[3] P. Belhumeur, D. Jacobs, D. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PP(99):1–1, 2013.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: Turbo-codes. In *Communications*, volume 2, pages 1064 –1070 vol.2, 1993.

[5] Christopher M Bishop. *Pattern Recognition and Machine Learning*, volume 1. Springer New York, 2006.

[6] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer*

Graphics and Interactive Techniques, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[7] Vicki Bruce and Andy Young. Understanding face recognition. *British Journal of Psychology*, 1986.

[8] MichaelC. Burl, Markus Weber, and Pietro Perona. A probabilistic approach to object recognition using local photometry and global geometry. In Hans Burkhardt and Bernd Neumann, editors, *Computer Vision ECCV98*, volume 1407 of *Lecture Notes in Computer Science*, pages 628–641. Springer Berlin Heidelberg, 1998.

[9] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2887–2894, 2012.

[10] O. Celiktutan, H.C. Akakin, and B. Sankur. Multi-attribute robust facial feature localization. In *Automatic Face Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, pages 1–6, 2008.

[11] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[12] T. F. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *ECCV*, volume 1407, pages 484–498. 1998.

[13] T. F. Cootes and C. J. Taylor. Cj.taylor, "active shape models - "smart snakes. In *in Proceedings of the British Machine Vision Conference*, 1992.

[14] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models their training and application. *Comput. Vis. Image Underst.*, 61:38–59, January 1995.

[15] Timothy F Cootes, Gavin V Wheeler, Kevin N Walker, and Christopher J Taylor. View-based active appearance models. *Image and vision computing*, 20(9):657–664, 2002.

[16] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[17] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 10–17 vol. 1, 2005.

[18] David Cristinacce and Timothy F Cootes. Facial feature detection using adaboost with shape constraints. In *BMVC*, pages 1–10, 2003.

[19] David Cristinacce and Timothy F Cootes. Feature detection and tracking with constrained local models. In *BMVC*, volume 17, pages 929–938, 2006.

[20] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. CVPR, 2005.

[21] Ian L. Dryden and Kanti V. Mardia. *Statistical Shape Analysis.* John Wiley & Sons, 1998.

[22] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[23] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[24] PedroF. Felzenszwalb and DanielP. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

[25] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–264–II–271 vol.2, 2003.

[26] Martin A. Fischler and R.A. Elschlager. The representation and matching

of pictorial structures. *Computers, IEEE Transactions on*, C-22(1):67–92, 1973.

[27] William T. Freeman, Egon C. Pasztor, and Owen T. Carmichael. Learning low-level vis. *Int. J. of Comp. Vis.*, 40:25–47, 2000.

[28] D.J. Grelotti, I. Gauthier, and R.T. Schultz. Social interest and the development of cortical face specialization: what autism teaches us about face processing. *Developmental Psychobiology*, 2002.

[29] Ralph Gross, Iain Matthews, and Simon Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(11):1080–1093, November 2005.

[30] Leon Gu and Takeo Kanade. A generative shape regularization model for robust face alignment. In *ECCV*, volume 5302, pages 413–426. 2008.

[31] James V. Haxby, Elizabeth A. Hoffman, and M.Ida Gobbini. Human neural systems for face recognition and social communication. *Biological Psychiatry*, 51(1):59 – 67, 2002. Social Anxiety: From Laboratory Studies to Clinical Practice.

[32] Er T. Ihler, Erik B. Sudderth, William T. Freeman, and Alan S. Willsky. Efficient multiscale sampling from products of gaussian mixtures. In *In NIPS 17*. MIT Press, 2003.

[33] M. Isard. Pampas: real-valued graphical models for computer vision. In *CVPR*, 2003.

[34] S. Jaiswal, T.R. Almaev, and M.F. Valstar. Guided unsupervised learning of mode specific models for facial point detection in the wild. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 370–377, Dec 2013.

[35] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[36] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *Int. J. of Comp. Vis.*, 1:321–331, 1988.

[37] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[38] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[39] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir Bourdev, and ThomasS. Huang. Interactive facial feature localization. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision ECCV 2012*, volume 7574 of *Lecture Notes in Computer Science*, pages 679–692. Springer Berlin Heidelberg, 2012.

[40] Stan Z Li, HongJiang Zhang, Qiansheng Cheng, et al. Multi-view face alignment using direct appearance models. In *Automatic Face and Ges-*

ture Recognition, 2002. Proceedings. Fifth IEEE International Conference on, pages 324–329. IEEE, 2002.

[41] Xiaoming Liu. Generic face alignment using boosted appearance model. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.

[42] P. Lucey, J.F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *CVPRW*, pages 94 –101, june 2010.

[43] David J. C. MacKay. *Information Theory, Inference and Learning Algorithms*, volume 1. Cambridge University Press, 2003.

[44] Binefa X. Martinez B., Valstar M. F. and Pantic M. Local evidence aggregation for regression based facial point detection. *IEEE Trans. Pattern Analysis and Machine Learning*, 2013.

[45] Iain Matthews and Simon Baker. Active appearance models revisited. *Int. J. of Comp. Vis.*, 60:135–164, 2004.

[46] Kevin Nickels and Seth Hutchinson. Estimating uncertainty in ssd-based feature tracking. *Image and Vision Computing*, 20:47–58, 2002.

[47] M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.

[48] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[49] Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[50] D. Ramanan and C. Sminchisescu. Training deformable models for localization. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 206–213, 2006.

[51] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 896–903, 2013.

[52] J. Saragih and R. Goecke. A nonlinear discriminative approach to aam fitting. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007.

[53] Jason Saragih, Simon Lucey, and Jeffrey Cohn. Deformable model fitting by regularized landmark mean-shift. *Int. J. of Comp. Vis.*, 91:200–215, 2011.

[54] B. Silverman. *Density estimation for statistics and data analysis.* 1986.

[55] Mikkel B. Stegmann and David Delgado Gomez. A brief introduction to statistical shape analysis. Technical report, University of Denmark, DTU, 2002.

[56] Erik B. Sudderth, Alexander T. Ihler, William T. Freeman, and Alan S. Willsky. Nonparametric belief propagation. In *CVPR*, volume 1, pages 605–612, 2003.

[57] Birgi Tamersoy, Changbo Hu, and J. K. Aggarwal. Nonparametric facial feature localization. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 838–845. IEEE, 2013.

[58] M. Valstar, B. Martinez, X. Binefa, and M. Pantic. Facial point detection using boosted regression and graph models. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2729–2736, 2010.

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

[60] Yang Wang, S. Lucey, and J.F. Cohn. Enforcing convexity for improved alignment with constrained local models. In *CVPR*, pages 1 –8, june 2008.

[61] Yair Weiss. Correctness of belief propagation in graphical models with loops. *Neural Comp.*, 12(1):1–41, 2000.

[62] Yair Weiss and William T. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation*, 13(10):2173–2200, 2001.

[63] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. Von der Malsburg. Face recognition by elastic bunch graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):775–779, 1997.

[64] Hao Wu, Xiaoming Liu, and G. Doretto. Face alignment via boosted ranking model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[65] Jing Xiao, Simon Baker, Iain Matthews, and Takeo Kanade. Real-time combined 2d+ 3d active appearance models. In *CVPR (2)*, pages 535–542, 2004.

[66] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 532–539. IEEE, 2013.

[67] Heng Yang and Ioannis Patras. Sieving regression forest votes for facial feature detection in the wild. ICCV, 2013.

[68] Xiang Sean Zhou, A. Gupta, and D. Comaniciu. An information fusion framework for robust shape tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(1):115–129, Jan 2005.

[69] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012.

Vita

Birgi Tamersoy was born in Izmir, Turkey, the only son of retired English teacher Çimen Tamersoy and retired computer engineer Mahmut Tamersoy. After graduating from American Collegiate Institute in June 2003, he entered the Computer Engineering program of Bilkent University, Ankara, where he was awarded with three merit scholarships in three consecutive years. He received his Bachelor of Science degree in May 2007. He started pursuing his graduate degree in August 2007, in the department of Electrical and Computer Engineering of the University of Texas at Austin. Since August 2008, he is a member of the Computer and Vision Research Center and working under the supervision of Dr. J. K. Aggarwal. He received his Master of Science in Engineering degree in August 2009.

Permanent e-mail address: birgitamersoy@gmail.com

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.