Imperial College London
Department of Computing

Robust Statistical Deformable Models

Epameinondas Antonakos

March, 2017

Supervised by Dr. Stefanos Zafeiriou

Submitted in part fulfilment of the requirements for the degree of PhD in Computing and the Diploma of Imperial College London. This thesis is entirely my own work, and, except where otherwise indicated, describes my own research.

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Acknowledgements

I would like to thank my supervisor Dr. Stefanos Zafeiriou for being a remarkable mentor and an inspiring teacher throughout my Ph.D. He has always provided me unique guidance and showed me how to reach my full potential, while patiently transferring me his knowledge and way of thinking. I am also thankful to Prof. Maja Pantic for accepting me in the iBUG group and supporting me with whatever I needed. I would also like to thank Dr. Georgios Tzimiropoulos for our collaboration in the early stages of my Ph.D.

Moreover, I would like to express my most genuine gratitude to my friends and members of the Menpo team – Joan Alabort-i-Medina, James Booth, Patrick Snape and George Trigeorgis. No words can express the amount of things they all individually taught me and my Ph.D. experience would never be the same without our constructive collaboration. My gratitude and appreciation also goes to all my colleagues at the iBUG group, and especially to Christos Georgakis, Stefanos Eleftheriadis, Simos Nikitidis and Thanos Papaioannou, since their friendship and support made these last years an amazing and unforgettable experience.

Finally, I would like to express my utmost gratefulness to my parents, Petros and Eleni, and my brother, George, for their continuous and unconditional support. Last but foremost, my deepest gratitude goes to my life partner and soul-mate, Leda, for her boundless love. This thesis is dedicated to them.

Abstract

During the last few years, we have witnessed tremendous advances in the field of 2D Deformable Models for the problem of landmark localization. These advances, which are mainly reported on the task of face alignment, have created two major and opposing families of methodologies. On the one hand, there are the generative Deformable Models that utilize a Newton-type optimization. This family of techniques has attracted extensive research effort during the last two decades, but has lately been criticized of achieving inaccurate performance. On the other hand, there is the currently predominant family of discriminative Deformable Models that treat the problem of landmark localization as a regression problem. These techniques commonly employ cascaded linear regression and have proved to be very accurate.

In this thesis, we argue that even though generative Deformable Models are less accurate than discriminative, they are still very valuable for several tasks. In the first part of the thesis, we propose two novel generative Deformable Models. In the second part of the thesis, we show that the combination of generative and discriminative Deformable Models achieves state-of-the-art results on the tasks of *(i)* landmark localization and *(ii)* semi-supervised annotation of large visual data.

Contents

Chapter 1

Introduction

Contents

1.1 Problem Scope and Challenges

Digital cameras exist everywhere around us and are the artificial "eyes" of the current and future technological era. We find them embedded in most everyday smart electronic devices (*e.g.*, phones, tablets, laptops, TVs, cars, gaming consoles, etc.), installed in almost all major urban streets and inside commercial stores for surveillance and service purposes, while, of course, they are an essential part of modern robotics. This wealth of electronic "eyes" has increased the need and effort to make computers to *"recognize and understand what they see"* by inculcating them with the ability to learn, detect and recognize.

An important step towards this direction is to enable computers to accurately *detect deformable objects under unconstrained conditions (commonly referred to as "in-the-wild")*, *i.e.* images obtained in uncontrolled recording settings typically containing large variations in terms of illumination, identity, pose, and containing occlusions. Deformable objects are articulated objects that exhibit rigid shape variations and, in most cases, large appearance variations,

e.g. the human face, body, cars, etc. Note that the term "detection" does not refer to the task of finding the bounding box of an object[1]. It refers to the task of localizing a set of *sparse landmark (fiducial) points* that correspond to semantically meaningful parts of the object. This problem is typically addressed using Deformable Models, which have emerged as an important research field during the last few decades, existing at the intersection of Computer Vision, Statistical Pattern Recognition and Machine Learning. The application of a Deformable Model typically has two phases:

- *Training*: This step involves the training of a model that can describe a deformable object, thus captures its shape and appearance variations. It requires the annotation of visual data that contain the object with a set of landmark points that need to correspond to semantically meaningful parts of the object.
- *Fitting (or Matching)*: This procedure aims to fit the learned Deformable Model to a new image by localizing the landmark points of the object. This is usually achieved through an energy minimization procedure [117, 18, 124, 153, 8, 156, 9, 11, 10, 5] or, more recently, by applying a cascade of learned rules [171, 128, 82, 14, 183, 152, 151]. Note that the optimization finds a local minimum, thus fitting is commonly initialized with a bounding box that provides a sparse shape that is close to the optimum.

Deformable Models can be separated in two major families based on the characteristics of their training and fitting:

1. **Discriminative Models**: The methodologies of this category commonly employ some kind of regression in a cascaded manner in order to localize the landmarks' coordinates [171, 128, 82, 14, 183, 152, 151]. Thus, they learn average rules (descent directions) from the training set that are readily applied on a test image. This reveals their biggest advantage of having real-time fitting performance. Additionally, they have been proved to be very robust to bad initializations that are far from the desired optimum. However, these techniques are data hungry. Given that they learn a set of generic rules from the training set, they tend to become more accurate by increasing the number of training examples. This, in combination to the fact that their training procedure is computationally expensive due to their discriminative nature, makes the Deformable Models of this category difficult to fine-tune.

[1]The problem of bounding box object detection is modeled differently. The dominant and most popular trend is to learn features invariant to the object parts' deformations, such as those learned by Deep Convolutional Neural Networks. Hence, the parts and their deformations are not modeled [66, 76, 103, 129].

2. **Generative Models**: The methodologies of this family model the shape and appearance of a deformable object in a probabilistic manner which results in the ability to generate unseen instances of the object [117, 39, 18]. Specifically, they model the joint distribution between observed data and some latent (hidden, unobserved) structure (*e.g.*, the structure of the parts of an observed object and their temporal dynamics). Thanks to their generative nature, their training process is very fast and requires much less training examples compared to discriminative Deformable Models. Their fitting process usually involves minimizing a non-linear least squares energy function that is commonly solved with iterative algorithms such as Gauss-Newton and Gradient Descent [117, 18, 124, 153, 8, 156, 9, 11, 10, 5]. Thus, their optimization estimates image-specific descent directions which makes them very accurate when initialized close to the optimum. However, their fitting tends to be slow and requires many iterations to converge.

During the last decade, we have witnessed **tremendous developments** in the field of Deformable Models, mainly due to:

- The *abundance of visual data*, spread mostly through the Internet via web services such as Google Images, Bing and Youtube. This has led to the development of huge databases (such as PASCAL [55], LFW [98] and the series of ImageNet corpora [49]) consisting of visual data captured under unconstrained realistic settings (in-the-wild).
- The development of *powerful visual features* that can describe objects in a robust manner (*e.g.*, Scale Invariant Feature Transforms (SIFT) [109], Histogram of Oriented Gradients (HOG) [46], Local Binary Patterns (LBP) [120, 121, 122] and recently Deep Convolutional Neural Networks (DCNNs) [144, 56], etc.).
- The incorporation of *powerful, mainly discriminative, methodologies* for classification and regression, which led to the development of efficient visual object detection and recognition algorithms [58, 171, 14, 82, 128].

However, even though the above research developments are significant, there still exist some important **disadvantages and challenges** that need to be addressed:

- Due to their discriminative nature, most existing methodologies require collection of *many training data* in order to build a powerful Deformable Model with good generalization performance. This means that their training demands plenty of computing

resources and time, which makes them inappropriate for re-training and fine-tuning using a common everyday-use device with limited processing power and memory.

- Although it is easy to gather large amounts of visual data, their *semantic annotation* in terms of parts of deformable objects, their behaviors, their interactions, and outliers *still remains an expensive, tedious, labor intensive and prone to human errors procedure*. For example, as explained in [132], in the case of facial images' annotation, a trained annotator needs about 5 minutes to manually annotate from scratch an image with 68 landmark points (depending on the difficulty of the image). This means that the annotation of 1000 images requires about 3.5 days of continuous work, 10000 images require a bit more than a month of continuous work, etc. It is worth mentioning, that due to fatigue a person cannot annotate correctly for more than 4-5 hours per day. Furthermore, except for face, there hardly exists another object that has been annotated with regards to parts.

- Due to the lack of a standardized way (benchmark) to compare methodologies and to the *limited existence of open-source code*, the evaluation of newly proposed techniques is inconsistent and, most of the times, unfair. Researchers employ different databases and experimental protocols, which lead to unfair comparisons between existing methods. Moreover, in the vast majority of cases, the released implementations have the form of pre-compiled binaries accompanied with pre-trained models, which makes it impossible to tweak and experiment with.

As explained above, the work presented in this Ph.D. thesis aims to solve the problem of landmark localization by exploring generative and discriminative 2D Deformable Models. Nevertheless, there has been significant research effort on directions that **approach the problem in different ways**. Specifically, these are the most important current trends and the reasons why they are not within the scope of this thesis:

- 3D facial shape estimation from monocular images is the main alternative to 2D Deformable Models. The predominant lines of research include 3D Morphable Model (3DMM) [26, 27, 30, 31, 32, 125] and Shape-from-Shading (SfS) [21, 54, 83, 141, 150]. 3DMM is a generative statistical model of the 3D shape and texture of a deformable object. The biggest advantage of 3DMMs is the fact that dense 3D shape modeling provides a more natural and accurate representation of the human face that overpasses the limitations and ambiguities of 2D sparse landmarks (*e.g.*, the semantic meaning of

the 2D landmarks around the jaw is ambiguous and inconsistent over the head pose variation [132]). However, capturing 3D facial data is a tedious task that also requires specialised acquisition devices that cannot operate under unconstrained conditions. As a result, there only exist small databases with limited variance that capture a few hundred faces under laboratory conditions [125, 26] and are not suitable neither for "in-the-wild" applications, nor for training discriminative methodologies. These are the main reasons why 3D Deformable Models are not within the scope of this thesis. Nevertheless, during the last year, 3D Deformable Models have re-attracted increased interest thanks to the development of the first powerful 3D models trained on thousands of subjects [32, 31], as well as the organization of the first challenges on the task [75].

- Deep Learning, and more importantly, Convolutional Neural Networks (CNNs) have become the most popular trend in Computer Vision and have significantly contributed in improving the performance of various tasks such as image classification [94, 145, 146, 73], generic object detection [66, 129], semantic segmentation [66, 108, 37, 70] and instance segmentation [127, 72]. The progress witnessed over the last decade is highly related to the spatial accuracy that CNNs were able to achieve over time, starting from boxes, moving to coarse instance regions until reaching accurate pixel-level labelling. As a result, it was not until recently that CNNs were able to perform tasks with accurate spatial localization, such as body pose estimation [148, 178] and facial landmark localization [130, 144, 181, 151, 93, 70]. However, despite the fact that facial databases include reasonably large numbers of "in-the-wild" annotated images for the generative or discriminative methodologies of this thesis, they are not large enough in order to train CNNs. As a matter of fact, LFPW [22] and HELEN [97], which are the largest facial databases annotated with 2D landmark points, consist of 1035 and 2330 images, respectively. This is orders of magnitude less than the size of ImageNet [49] ($\sim 15M$), MegaFace [84] ($1M$), WIDER [177] ($\sim 400k$) or Microsoft COCO [105] ($330k$) that are commonly used for other tasks. Finally, it is worth mentioning that the research community has been actively attempting to increase the size of annotated data during the last few months [180], which will benefit Deep Learning approaches and potentially further improve face alignment accuracy.

1.2 Objectives

The aim of this Ph.D. thesis is to investigate ways to address the aforementioned challenges by combining the main concepts and advantages of generative and discriminative Deformable

Models. Specifically, this work has the following objectives:

- **Objective 1: Develop generative Deformable Models that achieve accurate performance without requiring a large amount of training data.** Generative Deformable Models have attracted extended research interest during the last two decades. However, they have often been criticized [68, 159] for their inability to generalize well to conditions beyond the ones exhibited in the training set and have been characterized as inappropriate for fitting in-the-wild images. As a matter of fact, they have always been regarded as ideal options to be used with data captured under controlled recording scenarios and for building instance-specific models. One of the objectives of this thesis is to develop generative Deformable Models that take advantage of recent advances in component analysis and visual feature extraction in order to achieve accurate and robust performance without the need of large annotated training datasets. An additional aim is to compare the advantages and disadvantages between *holistic* and *part-based* appearance representations. A holistic appearance representation takes into account the texture that lies inside the whole surface of a deformable object. On the other hand, a part-based appearance representation extracts local texture patches that are centered around the landmark points.

- **Objective 2: Propose methodologies for training Deformable Models with limited or even no human supervision** and explore solutions towards the online incremental update of these models with new training samples (**lifelong learning**). This refers to the task of constantly updating Deformable Models with images coming from the web − in other words, the task of semi-automatic annotation of large collections of images. During the past twenty years, there has been huge dispute about whether generative or discriminative approaches are more appropriate for learning visual data [80]. Even though, there is no solid theoretical proof that discriminative models are always better than generative ones [80], and in many cases the latter produce state-of-the-art results [9, 8, 156], the majority of researchers use discriminative models for learning from annotated data. However, discriminative methods are of limited use under an unsupervised setting. *For the purpose of applications with minimal, or even no supervision, the family of generative techniques is more suitable.* Nevertheless, although the cost of manual annotation is well understood, unsupervised learning of Deformable Models has not received the proper attention and has been mainly restricted to controlled conditions and in small non-representative sets [19, 88, 166].

- **Objective 3: Achieve state-of-the-art landmark localization performance by combining the advantages of generative and discriminative Deformable Models.** Discriminative (cascaded regression) Deformable Models have been shown to be more accurate and robust than generative models under challenging initializations. On the other had, generative models are very accurate when the initialization of their iterative optimization is reasonably close to the desired optimum solution. One of the objectives of this thesis is to analyze the main characteristics of these two families and create a unified model that benefits from their advantages and achieves state-of-the-art performance by *outperforming both*.

- **Objective 4: Release an open-source implementation of all proposed approaches that contributes towards the need to standarize benchmarking.** One of the goals of this Ph.D. thesis is to accompany all the proposed methodologies of Objectives 1, 2 and 3 with a stable, tested and well-documented open-source implementation of both training and fitting. This can have a huge impact on the research community, since it allows to tweak with the proposed models and easily compare with them.

It should be highlighted that the ideas and methodologies presented in this Ph.D. thesis are directly applicable to various deformable objects. However, this work focuses entirely on the object of **human face**. The main reasons behind that is that there are many large and carefully annotated databases with facial images − much more than for any other kind of deformable object. In fact, academic research lacks annotated databases for the vast majority of deformable objects. Furthermore, the human face is a very representative example of an object that exhibits large variations in deformations and appearance due to the plethora of facial expressions, race, identity, gender, etc. In addition to that, it is an object of great interest for many research fields with multiple applications. As a result, almost all research on Deformable Models is applied and tested on the human face. Recent large-scale challenges on facial alignment [133, 134, 132] are characteristic examples of the rapid progress being made in the field.

1.3 Contributions

In this section, the main contributions of this Ph.D. thesis are described in more detail and related to the aforementioned objectives of Sec. 1.2.

- **Chapter 4. Feature-based Active Appearance Models.** Lucas-Kanade (LK) [16,

18] is a Gauss-Newton algorithm that has become the standard choice for performing parametric image alignment with respect to the parameters of an affine transform. Various alterations have been proposed depending on the characteristics of the performed optimization. Additionally, Active Appearance Models (AAMs) [39, 117] is the most popular generative Deformable Model that employs the LK algorithm during fitting. Even though lots of improvements had been proposed for LK and AAMs, their performance was still poor compared to discriminative methodologies. In this chapter, we show that the combination of the non-linear least-squares optimization of a generative holistic Deformable Model with highly-descriptive, dense appearance features (*e.g.* HOG [46], SIFT [109]) can achieve excellent performance for the task of face alignment. We show that even though the employment of dense features increases the data dimensionality, there is a small raise in the time complexity and a significant improvement in the alignment accuracy. The presented experiments also provide a comparison between various features and prove that HOG and SIFT are the most powerful. We present very accurate and robust experimental results for both face alignment and fitting with feature-based LK and holistic AAMs, that prove their invariance to illumination and expression changes and their generalization ability to unseen faces. Especially in the case of HOG and SIFT holistic AAMs, we demonstrate results on in-the-wild databases that significantly outperform various powerful and efficient discriminative Deformable Models. This chapter provides solution to Objective 1 in Sec. 1.2.

- **Chapter 5. Active Pictorial Structures.** In this chapter, we exploit the effectiveness of part-based generative Deformable Models and shed light towards using a structure-based modeling for the shape and appearance of a deformable object. Specifically, we present a novel generative Deformable Model motivated by Pictorial Structures (PS) [61, 60, 7] and AAMs [117, 8, 9] for face alignment in-the-wild. Inspired by the tree structure used in PS, the proposed Active Pictorial Structures (APS) models the appearance of the object using multiple graph-based pairwise normal distributions (Gaussian Markov Random Field) between the patches extracted from the regions around adjacent landmarks. We show that this formulation is more accurate than using a single multivariate distribution (Principal Component Analysis) as commonly done in the literature. APS employs a weighted inverse compositional Gauss-Newton optimization with fixed Jacobian and Hessian that achieves close to real-time performance and state-of-the-art results. Finally, APS has a spring-like graph-based deformation prior term that makes them robust to bad initializations. We present extensive experiments on the task of face alignment, showing that APS outperforms many generative and discriminative

Deformable Models. Note that APS is the first weighted inverse compositional technique that proves to be so accurate and efficient at the same time. Additionally, thanks to its formulation, APS is suitable for articulated deformable objects with multiple degrees of freedom, such as the human body, hand, etc. This chapter provides solution to Objective 1 of Sec. 1.2.

- **Chapter 6. Automatic Construction of Deformable Models.** As explained in Sec. 1.1, in order to train Deformable Models with good generalization ability, a large amount of carefully annotated data is required, which is a highly time consuming and costly task. In this chapter, we propose the first method for automatic construction of deformable models using images captured in-the-wild. The only requirements of the method are a crude bounding box object detector and a priori knowledge of the object's shape (*e.g.* a point distribution model). The object detector can be as simple as the Viola-Jones algorithm [162, 163, 164] (e.g. even the cheapest digital camera features a robust face detector). The 2D shape model can be created by using only a few shape examples with deformations. In our experiments on facial Deformable Models, we show that the proposed automatically built model not only performs well, but also outperforms discriminative models trained on carefully annotated data. Note that this chapter deals with Objective 2 in Sec. 1.2 and the proposed methodology is the first one that shows that an automatically constructed model can perform as well as methods trained directly on annotated data.

- **Chapter 7. Adaptive Cascaded Regression.** As explained in Sec. 1.1, the two predominant families of Deformable Models are: *(i)* discriminative models that employ cascaded regression [171, 128, 82, 14, 183, 152], and *(ii)* generative models optimized with the iterative Gauss-Newton algorithm [117, 124, 153, 8, 156, 9, 5]. Although both of these approaches have been found to work well in practice, they each suffer from convergence issues. Cascaded regression has no theoretical guarantee of convergence to a local minimum and thus may fail to recover the fine details of the object. Gauss-Newton optimization is not robust to initializations that are far from the optimal solution. In this chapter, we propose to combine the best of these two worlds under a unified model, which directly answers Objective 3 in Sec. 1.2. We show that by combining the descent directions of cascaded regressors with the gradient descent directions from Gauss-Newton optimization, we can achieve both robustness to challenging initializations and accuracy with respect to fine details. Finally, we report state-of-the-art performance on the task of facial alignment against all current state-of-the-art generative and discriminative De-

formable Models. Our experiments are shown on the latest and most challenging face alignment challenge and ACR is compared against methodologies that are trained on more data and are used by industrial companies.

- **Section 1.5. The Menpo Project.** An open-source implementation is provided for all the proposed methodologies within the Menpo Project [1, 2]. The Menpo Project is a set of open source, cross-platform Python frameworks and associated tooling that provide end-to-end solutions for 2D and 3D deformable modeling.This fulfills Objective 4 of Sec. 1.2.

1.4 Impact and Applications

Generic Deformable Models that perform efficiently and accurately for a large range of deformable objects have a tremendous impact on *Human-Computer Interaction* applications such as multi-modal interaction, entertainment, digital arts, etc., and other fields like *Robotics*, *security*, etc. Furthermore, in the specific case of the human face, the task of facial landmark localization is the cornerstone for various higher level applications such as *facial expressions recognition*, *human behavior analysis*, *face recognition/verification*, *lip reading* and *sign language recognition*.

However, as mentioned before, one of the reasons that the task of landmark localization has not advanced even more within the fields of Computer and Robot Vision and has not expanded to more deformable objects is the cost of annotations. This highlights the impact of developing unsupervised techniques for learning Deformable Models which is immense, spanning a wide, diverse range of applications, namely:

- *Consumer-level robots*, which would be able to learn ad-hoc detailed Deformable Models of various objects.
- The design of *next generation Human-Computer Interaction* and *Ubiquitous Computing* systems, assisting the rapidly growing area of first person vision systems.
- Paving the road for *next generation Data Mining* and *Information Retrieval* systems (*i.e.*, analysis, indexing and retrieval of TV/Movie content in terms of actors appearance).

Additionally, the proposed ideas of this Ph.D. thesis, along with the provided open-source implementations, have the potential to accelerate research in other disciplines, such as *Biology*

and *Psychology* and other life sciences, by making the construction of complex detailed models of animals and humans an affordable and easy - even for non computer scientists - task.

Finally, it should be noted that academic research suffers from lack of annotated data for a large variety of objects. This fact highlights the proposed ideas for learning Deformable Models with minimal annotation effort can be a decisive step towards annotating large scale databases that can greatly boost the research progress.

1.5 The Menpo Project

An implementation of all the methodologies proposed in this Ph.D. thesis is provided within the Menpo Project[2,3] [1, 2]. The Menpo Project is a set of open-source BSD licensed Python frameworks and associated tooling that provide end-to-end solutions for 2D and 3D Deformable Modeling. It aims to enable researchers, practitioners and students to easily annotate new data sources and to investigate existing datasets. Of most interest to the Computer Vision is the fact that the Menpo Project contains completely open source implementations of a number of state-of-the-art algorithms for face detection and deformable model building. Characteristic examples of widely used state-of-the-art deformable model algorithms are Active Appearance Models (AAMs) [117, 9, 8, 156, 153, 154, 3], Constrained Local Models [137, 15] and Supervised Descent Method [171, 14].

There is still a noteworthy lack of high quality open source software in the field of Deformable Modeling. Most existing packages are encrypted, compiled, non-maintained, partly documented, badly structured or difficult to modify. This makes them unsuitable for adoption in cutting edge scientific research. Consequently, research becomes even more difficult since performing a fair comparison between existing methods is, in most cases, infeasible. For this reason, the Menpo Project represents an important contribution towards open science in the area. Additionally, it is important for Deformable Modeling to move beyond the established area of facial annotations and to extend to a wide variety of deformable object classes. Menpo can accelerate this progress by providing all of our tools completely free and permissively licensed.

The core functionality provided by the Menpo Project revolves around a powerful and flexible cross-platform framework written in Python. This framework has a number of sub-

[2]The Menpo Project is an open-source platform for all the stages of 2D and 3D Deformable Modeling. Website: http://www.menpo.org/. Github: https://github.com/menpo/

[3]The Menpo Project is created and maintained by James Booth, Patrick Snape, Joan Alabort-i-Medina and myself.

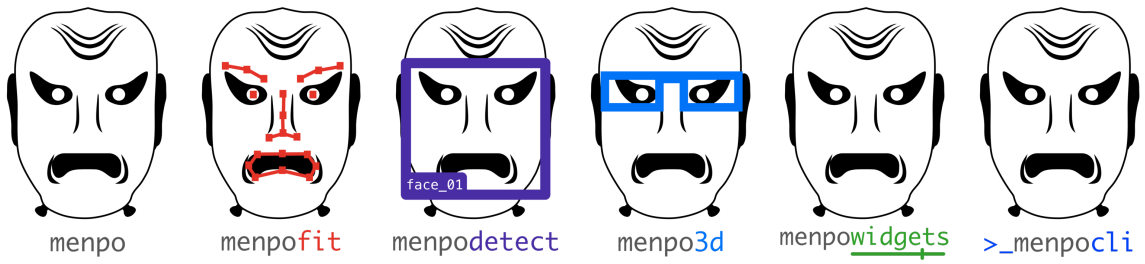


Figure 1.1: The Menpo Project [1, 2] is an open-source platform that provides solutions for all the stages of 2D and 3D Deformable Modeling (http://www.menpo.org/). It includes implementations for all the methodologies proposed in this thesis.

packages, all of which rely on a core package called `menpo`. The specialized subpackages are all based on top of `menpo` and provide state-of-the-art Computer Vision algorithms in a variety of areas (`menpofit`, `menpodetect`, `menpo3d`, `menpowidgets`).

- `menpo`: This is a general purpose package that is designed from the ground up to make importing, manipulating and visualizing image and mesh data as simple as possible. In particular, we focus on data that has been annotated with a set of sparse landmarks. This form of data is common within the fields of Machine Learning and Computer Vision and is a prerequisite for constructing Deformable Models. All `menpo` core types are landmarkable and visualizing these landmarks is a primary concern of the `menpo` library. Since landmarks are first class citizens within `menpo`, it makes tasks like masking images, cropping images within the bounds of a set of landmarks, spatially transforming landmarks, extracting patches around landmarks and aligning images simple.

- `menpofit`: This package provides all the necessary tools for training and fitting a large variety of state-of-the-art Deformable Models under a unified framework, including the ones presented in this thesis. The provided methods are:

  - Active Appearance Model (AAM) [117, 9, 8, 156, 153, 154, 3]
  - Supervised Descent Method (SDM) [171, 14]
  - Ensemble of Regression Trees (ERT) (powered by Dlib[4] [85]) [82]
  - Constrained Local Model (CLM) [137, 15]
  - Active Shape Model (ASM) [42]
  - Active Pictorial Structures (APS) [10]

[4]Dlib Machine Learning toolkit: http://dlib.net/

  - Lucas-Kanade (LK) and Active Template Model (ATM) [18, 16, 112, 9]

- `menpodetect`: This package contains methodologies for performing generic object detection in terms of a bounding box. The provided techniques include Viola-Jones object detector [162, 163, 164, 33], Support Vector Machines with HOG features [85, 86], Pico [113] and Deformable Part Model (DPM) [58, 116].

- `menpo3d`: It provides an open source implementation of 3D Morphable Models in-the-wild [26], as well as useful tools for importing, visualizing and transforming 3D data.

- `menpowidgets`: Package that includes widgets for "fancy" visualization of `menpo` objects. It provides user friendly, aesthetically pleasing, interactive widgets for visualizing images, shapes, landmarks, trained models and fitting results.

- `menpocli`: Command Line Interface (CLI) for the Menpo Project that allows to readily use pre-trained state-of-the-art `menpofit` facial models.

1.6 Publications

In this section, we provide a list of publications that were authored during the course of this Ph.D. thesis. We split these publications in two categories: *(i)* those that are related to the contents of this thesis 1.6.1 and *(ii)* other publications that are not directly relevant 1.6.2.

1.6.1 Related Publications

The work presented in this thesis is directly related to the following publications:

- **E. Antonakos**, and S. Zafeiriou. "Automatic Construction of Deformable Models In-The-Wild", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, pp. 1813-1820, June 2014.

- **E. Antonakos**, J. Alabort-i-Medina, G. Tzimiropoulos, and S. Zafeiriou. "HOG Active Appearance Models", *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Paris, France, pp. 224-228, October 2014.

- J. Alabort-i-Medina[5], **E. Antonakos**[5], J. Booth[5], P. Snape[5], and S. Zafeiriou. "Menpo: A Comprehensive Platform for Parametric Image Alignment and Visual Deformable

[5]Joint first authorship.

Models", *Proceedings of ACM International Conference on Multimedia (ACMM)*, Orlando, FL, USA, pp. 679-682, November 2014.

- **E. Antonakos**, J. Alabort-i-Medina, G. Tzimiropoulos, and S. Zafeiriou. "Feature-Based Lucas-Kanade and Active Appearance Models", *IEEE Transactions on Image Processing (T-IP)*, 24(9): pp. 2617-2632, September 2015.

- **E. Antonakos**, J. Alabort-i-Medina, and S. Zafeiriou. "Active Pictorial Structures", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, pp. 5435-5444, June 2015.

- **E. Antonakos**[5], P. Snape[5], G. Trigeorgis, and S. Zafeiriou. "Adaptive Cascaded Regression", *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA, *Oral*, September 2016.

1.6.2 Other Publications

This is a list of publications that are not directly relevant to the contents of this thesis, but, in most cases, are based on the outcome of some parts of this thesis:

- L. Zafeiriou, **E. Antonakos**, S. Zafeiriou, and M. Pantic. "Joint Unsupervised Face Alignment and Behaviour Analysis", *Proceedings of the 13th European Conference on Computer Vision (ECCV)*, Zurich, Switzerland, pp. 167-183, September 2014.

- **E. Antonakos**[5], A. Roussos[5], and S. Zafeiriou[5]. "A Survey on Mouth Modeling and Analysis for Sign Language Recognition", *Proceedings of IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Ljubljana, Slovenia, pp. 1-7, *Oral*, May 2015.

- G. Chrysos, **E. Antonakos**, S. Zafeiriou, and P. Snape. "Offline Deformable Face Tracking in Arbitrary Videos", *Proceedings of IEEE International Conference on Computer Vision Workshops (ICCVW), 300 Videos in the Wild (300-VW): Facial Landmark Tracking in-the-Wild Challenge & Workshop*, Santiago, Chile, December 2015.

- L. Zafeiriou, **E. Antonakos**, and S. Zafeiriou. "Joint Unsupervised Deformable Spatio-Temporal Alignment of Sequences", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

- G. Trigeorgis, P. Snape, M. Nicolaou, **E. Antonakos**, and S. Zafeiriou. "Mnemonic Descent Method: A recurrent process applied for end-to-end face alignment", *Proceedings of*

IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, June 2016.

- Y. Zhou, **E. Antonakos**, J. Alabort-i-Medina, A. Roussos, and S. Zafeiriou. "Estimating Correspondences of Deformable Objects "In-the-wild"", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

- C. Sagonas, **E. Antonakos**, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. "300 Faces In-The-Wild Challenge: Database and Results", *Image and Vision Computing (IMAVIS), Special Issue on Facial Landmark Localisation "In-The-Wild"*, vol. 47, pp. 3-18, 2016.

- G. Chrysos, **E. Antonakos**[6], P. Snape[6], A. Asthana, and S. Zafeiriou. "A Comprehensive Performance Evaluation of Deformable Face Tracking "In-the-Wild"", *International Journal on Computer Vision (IJCV)*, 2017.

- R. Guler, G. Trigeorgis, **E. Antonakos**, P. Snape, and S. Zafeiriou. "DenseReg: Fully Convolutional Dense Shape Regression In-the-Wild", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

- J. Booth, **E. Antonakos**, S. Ploumpis, G. Trigeorgis, Y. Panagakis, and S. Zafeiriou. "3D Face Morphable Models "In-the-Wild"", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

1.7 Thesis Outline

Chapter 2 makes a review of the related literature on the main two topics of this Ph.D. thesis: landmark localization with Deformable Models and their unsupervised training. Chapter 3 provides the basic definitions and notations that apply to all the topics of this thesis. Then, the remainder of the thesis is split in two parts, each one consisting of two chapters. Part I focuses on generative Deformable Models. Specifically, Chapter 4 presents feature-based holistic Active Appearance Models and Chapter 5 proposes Active Pictorial Structures, a novel part-based generative Deformable Model. Part II combines the main concepts of generative and discriminative Deformable Models with two applications: Chapter 6 shows how to

[6]Joint second authorship.

automatically train deformable Models without the need of manually annotated data, whereas Chapter 7 presents Adaptive Cascaded Regression which achieves state-of-the-art performance on face alignment in-the-wild. Finally, Chapter 8 concludes the thesis.

Chapter 2

Literature Review

Contents

2.1 Deformable Models

Deformable Models aim to solve the problem of generic object alignment in terms of localization of landmark (fiducial) points that correspond to semantically meaningful parts of the object. As explained in Sec. 1.2, although deformable models can be built for a variety of object classes, the majority of ongoing research has focused on the task of facial alignment. This is largely due to the plethora of existing databases with annotated facial images (*e.g.*, Labeled Face Parts in the Wild (LFPW) [22, 134], Annotated Faces in the Wild (AFW) [185, 134], HELEN [97, 134], IBUG [133, 134], 300W [133, 134, 132], Annotated Facial Landmarks in the Wild (AFLW) [89], MultiPIE [69, 134]), most of which have in-the-wild data. Recent large-scale challenges on facial alignment [133, 134, 132] are characteristic examples of the rapid progress being made in the field.

Currently, the most commonly-used and well-studied face alignment methods can be separated in two major families: *(i) generative* models that are iteratively optimized using Gauss-Newton or Gradient Descent algorithms, and *(ii) discriminative* models that employ regression in a cascaded manner. Deformable Models can also be split in two categories based on whether they use *(i) holistic* or *(ii) part-based* appearance representation. In the next sections, we review the related work of each category separately.

2.1.1 Generative Deformable Models

The most dominant algorithm of this category is, by far, the Active Appearance Model (AAM), which is descendant of Active Contour Model [81] and Active Shape Model [42]. An AAM consists of parametric linear models of the shape and appearance of an object. The shape model, usually referred to as Point Distribution Model (PDM), is built by applying Principal Component Analysis (PCA) [168, 79] on a set of aligned shapes. Similarly, the appearance model is built by applying PCA on a set of shape-free appearance instances, acquired by warping the training images into a reference shape. The use of a parametric statistical model gives rise to their labeling as generative models. The AAM objective function involves the minimization of the appearance reconstruction error with respect to the shape parameters. AAMs were initially proposed in [42, 38, 39], where the optimization was performed by a single regression step between the current image reconstruction residual and an increment to the shape parameters. However, the authors in [117, 18] showed how to linearize the AAM objective function and optimize it using the Gauss-Newton algorithm, which was inspired by their Lucas-Kanade (LK) algorithm [18, 16] for parametric image alignment with respect to the parameters of an affine transform.

Following this, Gauss-Newton optimization has been the modern de facto method for optimizing AAMs. The most common choice for both LK and AAMs matching is the Inverse Compositional (IC) image alignment algorithm [18, 117]. IC is a non-linear, Gauss-Newton optimization technique that aims to minimize the ℓ_2 norm between the warped image texture and a target texture. The target texture is the static template image in the case of affine image alignment with LK and a model texture instance in the case of non-rigid face alignment with AAMs. Since IC is a Gauss-Newton optimization technique, the registration result is sensitive to initialization and to appearance variation (illumination, pose, identity, expression, occlusion, etc.) exposed in the input and the target images [16]. Especially, in the case of AAMs with intensity-based appearance representation and optimized with the Project-Out IC algorithm [117], the model is incapable of adequately generalizing in order to be robust to outliers. This is the main reason why AAMs have been criticized of being able to perform well only in person specific applications and not generic ones. Many approaches have been proposed to deal with these issues and improve efficiency [18, 124, 6, 71, 17, 106, 119, 155, 3, 156, 154, 4, 5], robustness [118, 157, 112, 51, 68, 25, 3, 4, 53, 5] and generalization [68, 158, 153]. Many of the proposed methods introduce algorithmic improvements. The authors in [124] propose an adaptation on the fitting matrix and the employment of prior information to constrain the IC fitting process. In [16, 25] the ℓ_2 norm is replaced by a robust error function and the

optimization aims to solve a re-weighted least squares problem with an iterative update of the weights. Moreover, the method in [157] aligns two images by maximizing their gradient correlation coefficient.

Most of the existing AAM works utilize an intensity-based appearance, which is not suitable to create a generic appearance model and achieve accurate image alignment. However, the work presented in this thesis proves that this limitation can be easily overpassed and highly accurate results can be achieved. Specifically, in Chapter 4 we propose the employment of highly-descriptive, *dense* appearance features for both LK and AAMs. Especially in the case of HOG [46] and SIFT [109] AAMs, we demonstrate results on in-the-wild databases that significantly outperform state-of-the-art methods in facial alignment, which are discriminatively trained on much more data.

Feature-based image representation has gained extended attention for various Computer Vision tasks such as image segmentation and object alignment/recognition. There is ongoing research on the employment of features for both LK [106, 157, 112] and AAMs [153, 41, 63, 112, 64, 138, 87, 142, 143, 170, 47, 12, 53]. The authors in [106] use correspondences between dense SIFT [109] descriptors for scene alignment and face recognition. Various appearance representations are proposed in [138, 87] to improve the performance of AAMs. One of the first attempts for feature-based AAMs is [41]. The authors use novel features based on the orientations of gradients to represent edge structure within a regression framework. Similar features are employed in [153] to create a robust similarity optimization criterion. In [142], the intensities appearance model is replaced by a mixture of grayscale intensities, hue channel and edge magnitude.

Recently, more sophisticated multi-dimensional features are adopted for AAM fitting. The work in [112] proposes to apply the IC optimization algorithm in the Fourier domain using the Gabor responses for LK and AAMs. This is different than the framework proposed in this thesis, since in our approach the optimization is carried out in the spatial domain. In [143], a new appearance representation is introduced for AAMs by combining Gabor wavelet and Local Binary Pattern (LBP) descriptor. The work in [63] is the closest to the proposed framework in this thesis (Chapter 4). The authors employ Gabor magnitude features summed over either orientations or scales or both to build an appearance model. However, even though the optimization is based on the IC technique and carried out in the spatial domain, features are extracted at each iteration from the warped image. Finally, similarly to [63], the authors in [64] model the characteristic functions of Gabor magnitude and phase by using log-normal and Gaussian density functions respectively and utilize the mean of the characteristics over

orientations and scales. Very recently, the authors of [53] proposed to replace the linear shape and appearance models used in traditional AAM for deep shape and appearance models based on restricted Boltzmann machines (RBM).

2.1.2 Discriminative Deformable Models

The methodologies of this category aim to learn a regression function that regresses from the face's appearance (*e.g.*, commonly handcrafted features [109, 46]) to the target output variables (either the landmark coordinates or the parameters of a statistical shape model (PDM)). Although the history behind using linear regression in order to tackle the problem of face alignment spans back many years [39], the research community had turned towards alternative approaches due to the lack of sufficient data for training accurate regression functions. Nevertheless, over the last few years regression-based techniques have prevailed in the field thanks to the wealth of readily available annotated data and powerful handcrafted features [109, 46]. It has been recently shown [171, 173] that a single regression step is not sufficient for accurate generic alignment. On the contrary, a cascade of regression functions is more beneficial and is in fact employed by all recent discriminative methodologies [50, 35, 175, 176, 172, 36, 82, 128, 14, 152, 183, 101, 151] which have proved to be highly efficient and to generalize well.

The most important work in the area of discriminative Deformable Models, which can be applied to a big variety of problems that involve non-linear least squares problems, is that of Supervised Descent Method (SDM) [171, 173, 172]. SDM was the first work that presented cascaded regression as a general learning framework for optimizing non-linear objective functions by learning a set of rules from training data. In particular, the regressors at each cascade of SDM are linear and learn average descent directions in the space of the objective function. Note that in the original SDM formulation [171], even though the learnt descent directions are chained in a cascade, they are only related between them by the variance remaining from the previous cascade. Therefore, the initial cascade levels are prone to large descent steps which may not generalize well. This was addressed in [173] by clustering the descent directions into cohesive groups during training. At test time, a cluster is selected that represents the correct descent direction. For example, for face alignment this requires an initial estimate of the shape and the descent directions are clustered according to the head pose.

Many different discriminative Deformable Models have emerged, since the first proposal of SDM. They can be approximately separated into two categories based on the type of the employed regression function. The first category includes methodologies that employ a linear

regression [171, 173, 172, 14, 152, 183]. These methods usually employ hand-crafted features, such as HOG [46] and SIFT [109]. The second category, which has proved to be more efficient than the first one, includes methods that achieve regression via boosting of weak learners such as random ferns [36, 35] or random forests [82, 128]. These techniques tend to utilize data-driven features that are optimized directly by the regressor [35, 50, 82]. Furthermore, the authors in [14] have proposed an incremental algorithm which allows to parallelize the training of the cascade levels. A method to combine multiple landmark hypotheses using Structured Support Vector Machines was proposed in [174]. In [101], the authors substitute linear regressors by ensembles of linear and Gaussian processes regression trees. Finally, the authors in [130] and [144] learn a mapping from the initial bounding box acquired by the face detector to the landmarks' locations using Kernel Ridge Regression and Deep Convolutional Neural Network (DCNN), respectively.

2.1.3 Holistic vs. Part-Based Deformable Models

Until recently, all research efforts had mainly focused on developing Deformable Models with holistic appearance representation [38, 39, 117, 9, 3, 8, 68, 153, 154]. This means that the whole texture information inside the object's shape is taken into account and usually warped into a canonical space using a non-linear warping function (*e.g.*, Piecewise Affine Warp [18, 16], Thin-Plate Splines [29]).

Nevertheless, mainly due to the high complexity when using a holistic appearance representation, most recent existing methods started employing a part-based one. This means that a local patch is extracted from the neighborhood around each landmark. All the discriminative Deformable Models mentioned in Sec. 2.1.2 belong to this category, whereas the first part-based AAM was proposed in [156]. Additionally, among the most important part-based methodologies is the generative model of Pictorial Structures (PS) [61, 60, 7], its discriminative descendant Deformable Part Model (DPM) [58, 185] and their extensions like Deformable Structures [187]. PS learns a patch expert for each part and models the shape of the object using spring-like connections between parts based on a tree structure. Thus, a different distribution is assumed for each pair of parts connected with an edge, as opposed to the PCA shape model of an AAM that assumes a single multivariate normal distribution for all parts. The optimization aims to find a tree-based shape configuration for which the patch experts have a minimum cost and is performed using a dynamic programming algorithm based on the distance transform [59, 57].

Among the first part-based Deformable Models is Active Shape Model (ASM), initially

proposed in [42] and later re-utilized in [137]. The methodology in [42] fits ASM with an iterative search procedure that approximated local texture responses with isotropic Gaussian estimators. The authors in [45] proposed Constrained Local Model (CLM), one of the most important existing Deformable Models. CLM is natural extension of ASM, which employs a combined statistical model to generate local response maps. A probabilistic interpretation of CLM is derived in [137] which utilizes non-parametric response maps. This is further extended with shape priors in [135] and [23]. Moreover, the authors of [97] use several independent PCA priors to model the shape. The authors in [13] fit the CLM using a robust cascaded regression approach. The authors in [114, 115] use the efficient Regularized Particle Filters (RPF) during fitting. Finally, the work in [20] proposed to learn the local patch experts using Continuous Conditional Neural Fields.

2.2 Automatic Training of Deformable Models

Herein, we present the prior work on the automatic construction of Deformable Models, which is the focus of Chapter 6. Due to the fact that manual annotation is a rather costly, labor-intensive and prune to human mistakes procedure, unsupervised and semi-supervised learning of models for the tasks of alignment, landmark localization, tracking and recognition has attracted considerable attention [88, 78, 77, 107, 161, 149, 44, 19, 40, 126, 165, 62, 96, 74, 99, 184, 166]. In Chapter 6, we propose a method to automatically construct Deformable Models for object alignment and the most related works are [88, 161, 19, 40, 126]. The related family of techniques, known as image congealing [107, 99, 74, 96], uses implicit models to align a set of images as a whole, which means that both performing alignment to a new image and constructing a model is not straightforward. Our methodology differs from these works because we employ an explicit texture model which is learned through the process.

The two most closely related works to the proposed method are the automatic construction of AAMs in [19] and the so-called RASL (Robust Alignment by Sparse and Low-rank Decomposition) methodology in [126] for person-specific face alignment. There are two main differences between our framework and [19]. (1) We use a predefined statistical shape model instead of trying to find both the shape and appearance models. We believe that with the current available optimization techniques, it is extremely difficult to simultaneously optimize for both the texture and shape parameters. (2) We employ the robust component analysis of [158] for the appearance which deals with outliers. Thus, even though our method is similar in concept to [19], these two differences make the problem feasible to solve. In particular, the methodology in [19] fails to create a generic model even in controlled recording conditions,

due to extremely high dimensionality of the parameters to be found and to the sensitivity of the subspace method to outliers. This was probably one of the reasons why the authors demonstrate very limited and only person-specific experiments. Furthermore, our methodology bypasses some of the limitations of [126], which requires the presence of only one low-rank subspace, hence it has been shown to work only for the case of congealing images of a single person. Finally, we argue that in order for an automatically constructed AAM methodology to be robust to both within-class and out-of-class outliers[1], which cannot be avoided in totally unsupervised settings, statistical component analysis techniques should be employed [19].

[1]Within-class outliers refer to outliers present in the image of an object such as occlusion. Out-of-class outliers refer to images of irrelevant objects or to background.

Chapter 3

Basic Definitions and Notation

Contents

In this thesis, we denote vectors by small bold letters, matrices by capital bold letters, functions by capital calligraphic letters and scalars by small or capital regular-font letters.

3.1 Shape Representation and Model

In the problem of generic deformable object alignment (or landmark localization), the shape of an object consists of a set of n sparse landmark (fiducial) points that are located on semantically meaningful parts of the object. Assume that we have an $h \times w$ image $\mathbf{I}$ with c number of channels. Let us denote the coordinates of a landmark point within the Cartesian space of the image $\mathbf{I}$ as

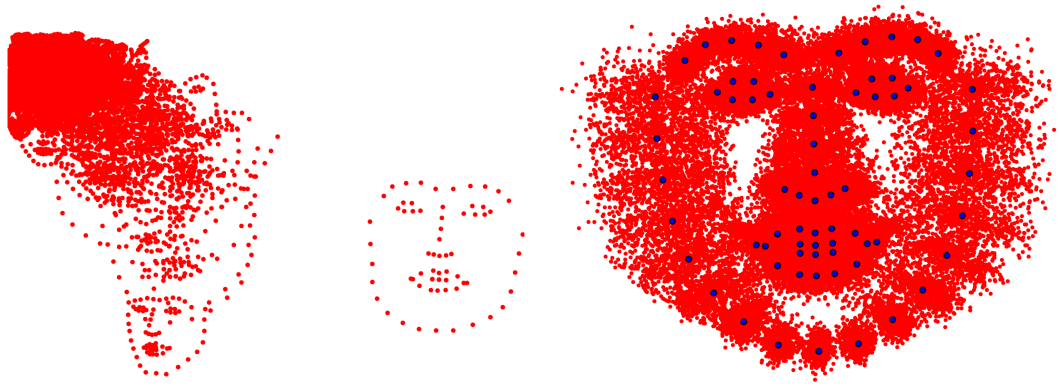
$$\boldsymbol{\ell}_i = [x_i, y_i]^\mathsf{T}, \ \forall i = 1, \ldots, n \tag{3.1}$$

where $x_i \in [1, w]$ and $y_i \in [1, h]$. The sparse *shape instance* of the object is given by the $2n \times 1$ vector

$$\mathbf{s} = \left[\boldsymbol{\ell}_1^\mathsf{T}, \boldsymbol{\ell}_2^\mathsf{T}, \ldots, \boldsymbol{\ell}_n^\mathsf{T}\right]^\mathsf{T} = [x_1, y_1, x_2, y_2, \ldots, x_n, y_n]^\mathsf{T} \tag{3.2}$$

Note that the number of landmarks used to annotate the human face in most existing databases is $n = 68$.

Given a set of N such training shape samples $\{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$, a parametric statistical model of the object's shape variance can be constructed with the following steps:



(a) Original shapes (b) Aligned shapes and mean shape

Figure 3.1: Examples of Generalized Procrustes Alignment on the shapes of LFPW trainset. The figure on the left shows the original shapes which expose large differences in terms of rotation, scale and translation due to the differences on the images resolutions and sizes. The figure on the right demonstrates the result of the alignment along with the mean shape.

1. Align the set of training shapes with respect to the global similarity transform (*i.e.*, scale, in-plane rotation and translation) using Generalized Procrustes Analysis [67]. Figure 3.1 shows an example of the result of such an alignment.

2. Apply PCA [79, 168] on the aligned shapes. This involves first centering the aligned shapes by subtracting the mean shape $\bar{\mathbf{s}}$ and then computing the basis of eigenvectors $\mathbf{U}_s \in \mathbb{R}^{2n \times N-1}$.

3. The returned shape subspace is further augmented with four eigenvectors that control the global similarity transform of the object's shape, thus the PCA subspace now consists of $N+3$ components. Please refer to [18] for further details about orthonormalizing the similarity eigenvectors with the PCA basis.

By keeping the first n_s eigenvectors, the resulting linear shape model has the form

$$\{\bar{\mathbf{s}}, \mathbf{U}_s\} \tag{3.3}$$

where $\mathbf{U}_s \in \mathbb{R}^{2n \times n_s}$ is the orthonormal basis and $\bar{\mathbf{s}} \in \mathbb{R}^{2n}$ is the mean shape vector. This linear shape model, which is also referred to as Point Distribution Model (PDM) [42, 39], can be used to generate new shape instances using the function $\mathcal{S} : \mathbb{R}^{n_s} \longrightarrow \mathbb{R}^{2n}$ as

$$\mathbf{s_p} = \mathcal{S}(\mathbf{p}) \equiv \bar{\mathbf{s}} + \mathbf{U}_s\mathbf{p} \tag{3.4}$$

where

$$\mathbf{p} = [p_1, p_2, \ldots, p_{n_s}]^\mathsf{T} \tag{3.5}$$

is the $n_s \times 1$ vector of *shape parameters* that control the linear combination of the eigenvectors. Figure 3.2 shows some exemplar shape instances generated using the first five principal components. The figure varies the parameter that corresponds to each component using the values $\left\{-3\sqrt{\lambda_i}, -\frac{3}{2}\sqrt{\lambda_i}, \frac{3}{2}\sqrt{\lambda_i}, 3\sqrt{\lambda_i}\right\}$, $\forall i = 1, \ldots, 5$ where λ_i denotes the corresponding eigenvalue.

3.2 Appearance Representation and Model

As explained in Sec. 2.1, Deformable Models can be split in two categories based on whether they utilize *(i) holistic* or *(ii) part-based* appearance representation. Figure 3.3 shows such an example. Additionally, all Deformable Models employ a feature-based image representation.

3.2.1 Feature Extraction

Features are computed by applying a *feature extraction function* that attempts to describe distinctive and important image characteristics (*e.g.*, SIFT [109], HOG [46]). Given an input image $\mathbf{I}$ with size $H \times W$, the feature extraction function $\mathcal{F}(\mathbf{I})$ is defined as

$$\mathcal{F} : \mathbb{R}^{H \times W} \longrightarrow \mathbb{R}^{H' \times W' \times D} \tag{3.6}$$

where $H' \times W'$ is the size of the output feature-based image and D is the number of channels. Note that feature functions can be separated in two categories: *(i) densely-sampled* and *(ii) sparsely-sampled*. Densely-sampled features extract a feature vector per image pixel, thus $H' = H$ and $W' = W$. On the other hand, sparsely-sampled features extract feature vectors from downsampled image locations, thus $H' < H$ and $W' < W$.

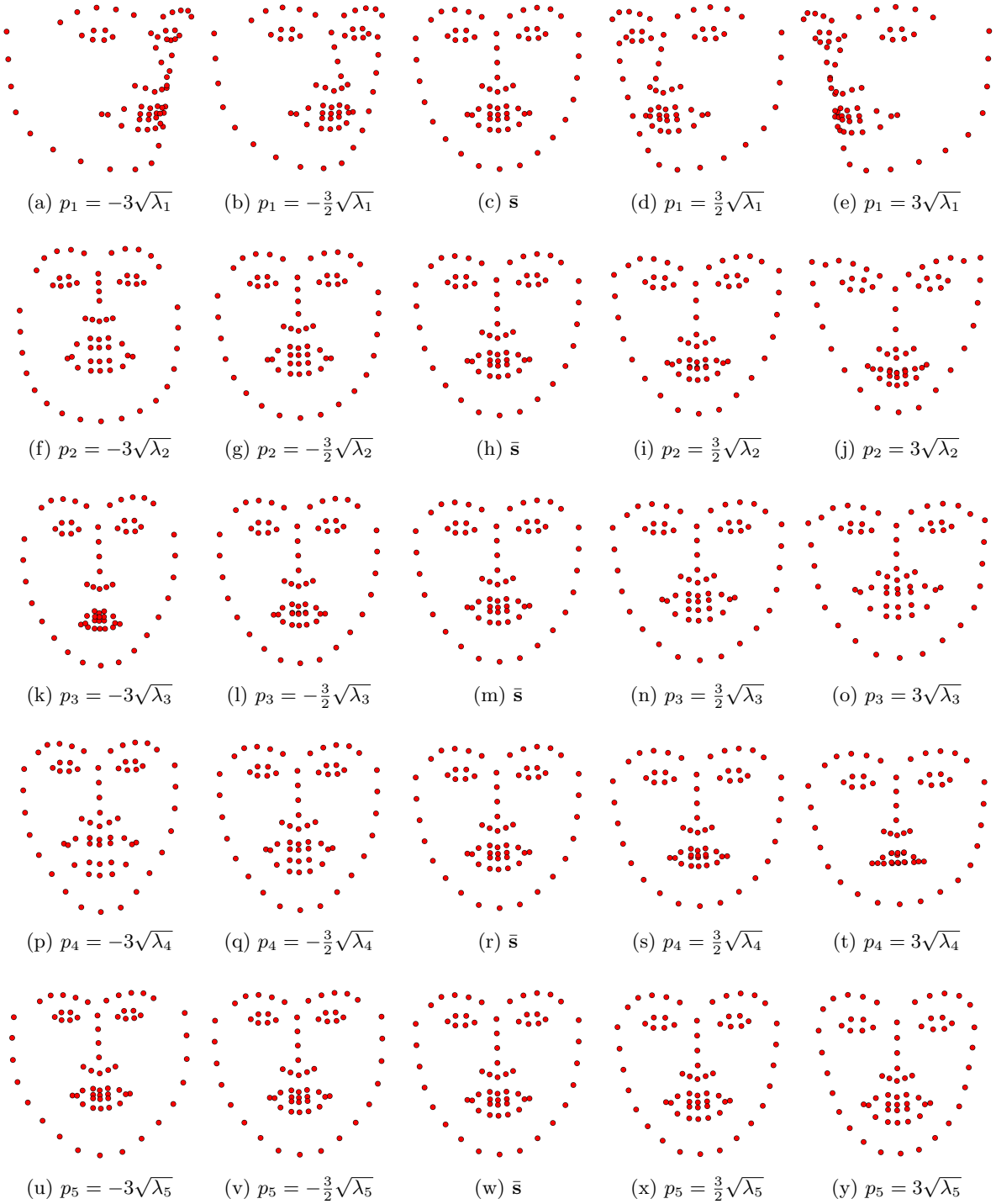
By denoting the input image in vectorial form $\mathbf{t}$ with size $HW \times 1$, the feature extraction function is redefined as

$$\mathcal{F} : \mathbb{R}^{HW} \longrightarrow \mathbb{R}^{m} \tag{3.7}$$

which returns a feature-vector of length $m = H'W'D$.

3.2.2 Holistic Appearance Representation

A holistic appearance representation aims to warp all the texture information within a shape instance to a reference shape (canonical space). In general, a warp function maps the points within a source shape to their corresponding coordinates in a target shape. In the Deformable



(a) $p_1 = -3\sqrt{\lambda_1}$ (b) $p_1 = -\frac{3}{2}\sqrt{\lambda_1}$ (c) $\bar{\mathbf{s}}$ (d) $p_1 = \frac{3}{2}\sqrt{\lambda_1}$ (e) $p_1 = 3\sqrt{\lambda_1}$

(f) $p_2 = -3\sqrt{\lambda_2}$ (g) $p_2 = -\frac{3}{2}\sqrt{\lambda_2}$ (h) $\bar{\mathbf{s}}$ (i) $p_2 = \frac{3}{2}\sqrt{\lambda_2}$ (j) $p_2 = 3\sqrt{\lambda_2}$

(k) $p_3 = -3\sqrt{\lambda_3}$ (l) $p_3 = -\frac{3}{2}\sqrt{\lambda_3}$ (m) $\bar{\mathbf{s}}$ (n) $p_3 = \frac{3}{2}\sqrt{\lambda_3}$ (o) $p_3 = 3\sqrt{\lambda_3}$

(p) $p_4 = -3\sqrt{\lambda_4}$ (q) $p_4 = -\frac{3}{2}\sqrt{\lambda_4}$ (r) $\bar{\mathbf{s}}$ (s) $p_4 = \frac{3}{2}\sqrt{\lambda_4}$ (t) $p_4 = 3\sqrt{\lambda_4}$

(u) $p_5 = -3\sqrt{\lambda_5}$ (v) $p_5 = -\frac{3}{2}\sqrt{\lambda_5}$ (w) $\bar{\mathbf{s}}$ (x) $p_5 = \frac{3}{2}\sqrt{\lambda_5}$ (y) $p_5 = 3\sqrt{\lambda_5}$

Figure 3.2: Exemplar instances of a statistical shape model (PDM) trained on the shapes of LFPW trainset. Each row shows the deformations covered by the first five principal components, where λ_i is the eigenvalue that corresponds to the i-th eigenvector.

(a) Original image annotated with a set of n sparse landmarks.

(b) Holistic appearance representation using Piecewise Affine Warp.

(c) Part-based appearance representation by extracting patches centered around the landmarks.

Figure 3.3: Example of holistic and part-based appearance representation based on a sparse shape.

Models literature, the warp function is commonly referred to as *motion model* and denoted as $\mathcal{W}(\mathbf{p})$. Its role is to extrapolate the position of all the pixels inside the convex hull of the reference shape to a particular shape instance $\mathbf{s}$ (generated using the shape parameters $\mathbf{p}$ as shown in Eq. 3.4) based on their relative position with respect to the sparse landmarks (for which direct correspondences are always known).

As also discussed and proved in Chapter 4, it is more beneficial to warp the extracted features rather than extracting features on the warped image. Thus, given an input image $\mathbf{I}$ with size $H \times W$ and its vectorized form $\mathbf{t}$, we can define a holistic feature-based appearance vector as

$$\mathbf{f} = \mathbf{t}_{\mathcal{F}}\left(\mathcal{W}(\mathbf{p})\right) \text{ with } \mathbf{t}_{\mathcal{F}} = \mathcal{F}(\mathbf{t}) \tag{3.8}$$

where the feature extraction is based on Eq. 3.7.

In this thesis, we employ the Piecewise Affine Warp (PWA) [43, 117], which performs the mapping based on the barycentric coordinates of the corresponding triangles between the two shapes that are extracted using Delaunay Triangulation [100]. An example of such an appearance representation is shown in Fig. 3.3b. Other warping methods could also be employed, such as Thin Plate Splines (TPS) [43, 124].

3.2.3 Part-Based Appearance Representation

The scientific community has lately turned towards part-based appearance representation, *i.e.*, extracting appearance patches centered around the landmark coordinates. Although this depends on the object class and application, in general, the part-based representation has

proved to be more efficient than the holistic as the warp function is replaced by a simple sampling function and it is also more natural for articulated rigid objects (*e.g.*, body pose, hand, etc.). Let us denote the vectorized form of an $h \times w$ image patch that corresponds to the image location $\boldsymbol{\ell}_i = [x_i, y_i]^\mathsf{T}$ as the $hw \times 1$ vector

$$\mathbf{t}_{\boldsymbol{\ell}_i} = [\mathbf{I}(\mathbf{z}_1), \mathbf{I}(\mathbf{z}_2), \ldots, \mathbf{I}(\mathbf{z}_{hw})]^\mathsf{T}, \ \{\mathbf{z}_j\}_{j=1}^{hw} \in \boldsymbol{\Omega}_{\boldsymbol{\ell}_i} \tag{3.9}$$

where $\boldsymbol{\Omega}_{\boldsymbol{\ell}_i}$ is a set of discrete neighboring pixel locations $\mathbf{z}_j = [x_j, y_j]^\mathsf{T}$ within a rectangular region centered at location $\boldsymbol{\ell}_i$ and hw is the image patch vector's length. By using the feature extraction function of Eq. 3.7, the procedure of extracting a feature-based vector from a patch centered at a given image location can be denoted as

$$\mathcal{F}(\mathbf{t}_{\boldsymbol{\ell}_i}) \equiv \mathcal{F}\left([\mathbf{I}(\mathbf{z}_1), \mathbf{I}(\mathbf{z}_2), \ldots, \mathbf{I}(\mathbf{z}_{hw})]^\mathsf{T}\right), \ \{\mathbf{z}_j\}_{j=1}^{hw} \in \boldsymbol{\Omega}_{\boldsymbol{\ell}_i} \tag{3.10}$$

Consequently, given a shape instance of the form of Eq. 3.2, the corresponding *part-based appearance vector* $\mathbf{f}$ is an $mn \times 1$ vector that consists of the concatenation of the vectorized feature-based image patches that correspond to the n landmarks of the shape instance, *i.e.*

$$\mathbf{f}(\mathbf{s}) = \left[\mathcal{F}(\mathbf{t}_{\boldsymbol{\ell}_1})^\mathsf{T}, \mathcal{F}(\mathbf{t}_{\boldsymbol{\ell}_2})^\mathsf{T}, \ldots, \mathcal{F}(\mathbf{t}_{\boldsymbol{\ell}_n})^\mathsf{T}\right]^\mathsf{T} \tag{3.11}$$

where $\mathbf{s}$ is given by Eq. 3.2.

3.2.4 Appearance Model

Given a set of N appearance vector samples $\{\mathbf{f}_1, \ldots, \mathbf{f}_N\}$ that are extracted using either Eq. 3.8 or Eq. 3.11, we can apply PCA to obtain a parametric statistical linear appearance model. By keeping the first n_a principal components, we end up with

$$\{\bar{\mathbf{a}}, \mathbf{U}_a\} \tag{3.12}$$

where $\mathbf{U}_a \in \mathbb{R}^{m \times n_a}$ is the orthonormal basis and $\bar{\mathbf{a}} \in \mathbb{R}^m$ is the mean appearance vector. This model can be used to generate new appearance instances using the function $\mathcal{A} : \mathbb{R}^{n_a} \longrightarrow \mathbb{R}^m$ as

$$\mathbf{a}_\mathbf{c} = \mathcal{A}(\mathbf{c}) \equiv \bar{\mathbf{a}} + \mathbf{U}_a \mathbf{c} \tag{3.13}$$

where

$$\mathbf{c} = [c_1, c_2, \ldots, c_{n_a}]^\mathsf{T} \tag{3.14}$$

is the $n_a \times 1$ vector of *appearance parameters* that control the linear combination of the eigenvectors.

Figures 3.5 and 3.6 show some exemplar appearance instances generated using the first five principal components of a holistic and a part-based appearance model, respectively. Note that both models are trained on grayscale intensities, in order to make the variance visualization more comprehensive. The figures vary the parameter that corresponds to each component using the values $\left\{-3\sqrt{\lambda_i}, -\frac{3}{2}\sqrt{\lambda_i}, \frac{3}{2}\sqrt{\lambda_i}, 3\sqrt{\lambda_i}\right\}$, $\forall i = 1, \ldots, 5$ where λ_i denotes the corresponding eigenvalue.

3.3 Facial Databases and Evaluation

As explained in Chapter 1, this thesis focuses on the deformable object of human face. Specifically, we utilize all the commonly-used in-the-wild databases that are annotated by Sagonas *et al.* [134, 133, 132] using the standard 68-point annotation mark-up proposed in the CMU MultiPIE database [69]. The employed in-the-wild databases, which contain images downloaded from the web that are captured under totally unconstrained conditions and exhibit large variations in pose, identity, illumination, expressions, occlusion and resolution, include:

- Labeled Face Parts in the Wild (LFPW) [22] (811 training images, 224 testing images)
- Annotated Faces in the Wild (AFW) [185] (337 images)
- HELEN [97] (2000 training images, 330 testing images)
- IBUG [133, 134] (135 images)
- 300W [133, 134, 132] (600 images)

Note that we do not consider the original annotations of LFPW (29 points) or HELEN (194 points), because recent works [183, 181, 128] have shown that these databases have become saturated for the original annotations. Figure 3.4 shows some examples from the employed in-the-wild databases.

The fitting process is commonly initialized by computing the face's bounding box using a face detector and then estimating the global similarity transform that fits the mean shape within the bounding box boundaries. Note that this initial similarity transform only involves a translation and scaling component and not any in-plane rotation. The accuracy of a landmark localization result is measured by the point-to-point RMS error between the fitted shape and the ground-truth annotations, as proposed in [185]. Denoting $\mathbf{s}^f = [x_1^f, y_1^f, x_2^f, y_2^f, \ldots, x_n^f, y_n^f]^\mathsf{T}$ and $\mathbf{s}^g =$

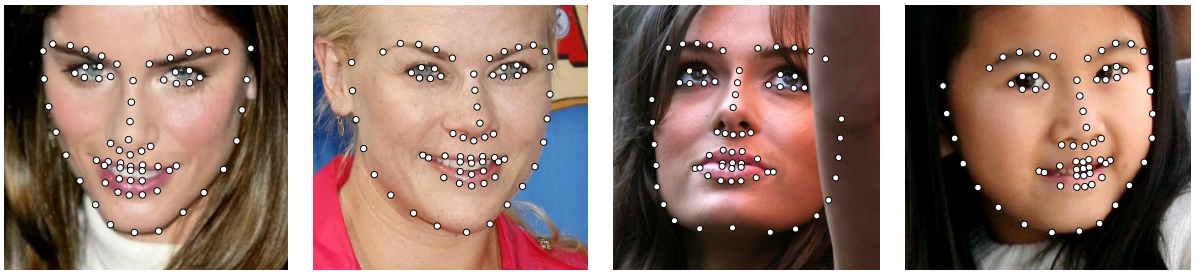
$[x_1^g, y_1^g, x_2^g, y_2^g, \ldots, x_n^g, y_n^g]^\mathsf{T}$ as the fitted shape and the ground-truth shape, respectively, then the error between them is expressed as

$$\text{RMSE} = \frac{\sum_{i=1}^{n} \sqrt{(x_i^f - x_i^g)^2 + (y_i^f - y_i^g)^2}}{cn} \tag{3.15}$$

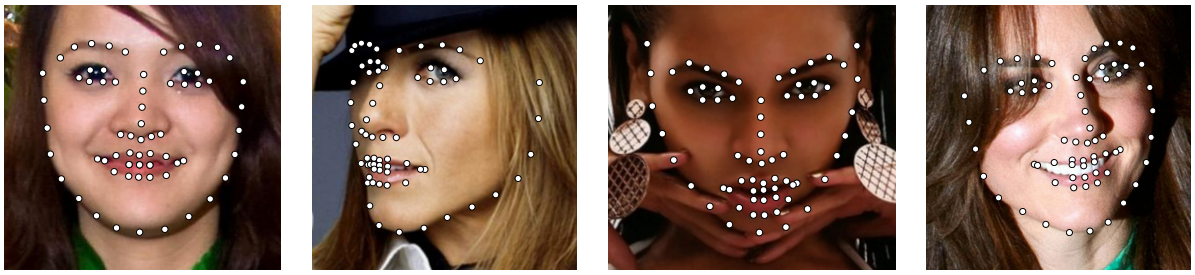
where c is a normalization constant. The interocular distance and the face size, defined as

$$c = \frac{(\max\{x_i^g\}_1^n - \min\{x_i^g\}_1^n + \max\{y_i^g\}_1^n - \min\{y_i^g\}_1^n)}{2} \tag{3.16}$$

are popular normalization choices. These errors are presented in the form of Cumulative Error Distribution (CED) and/or statistical measures.

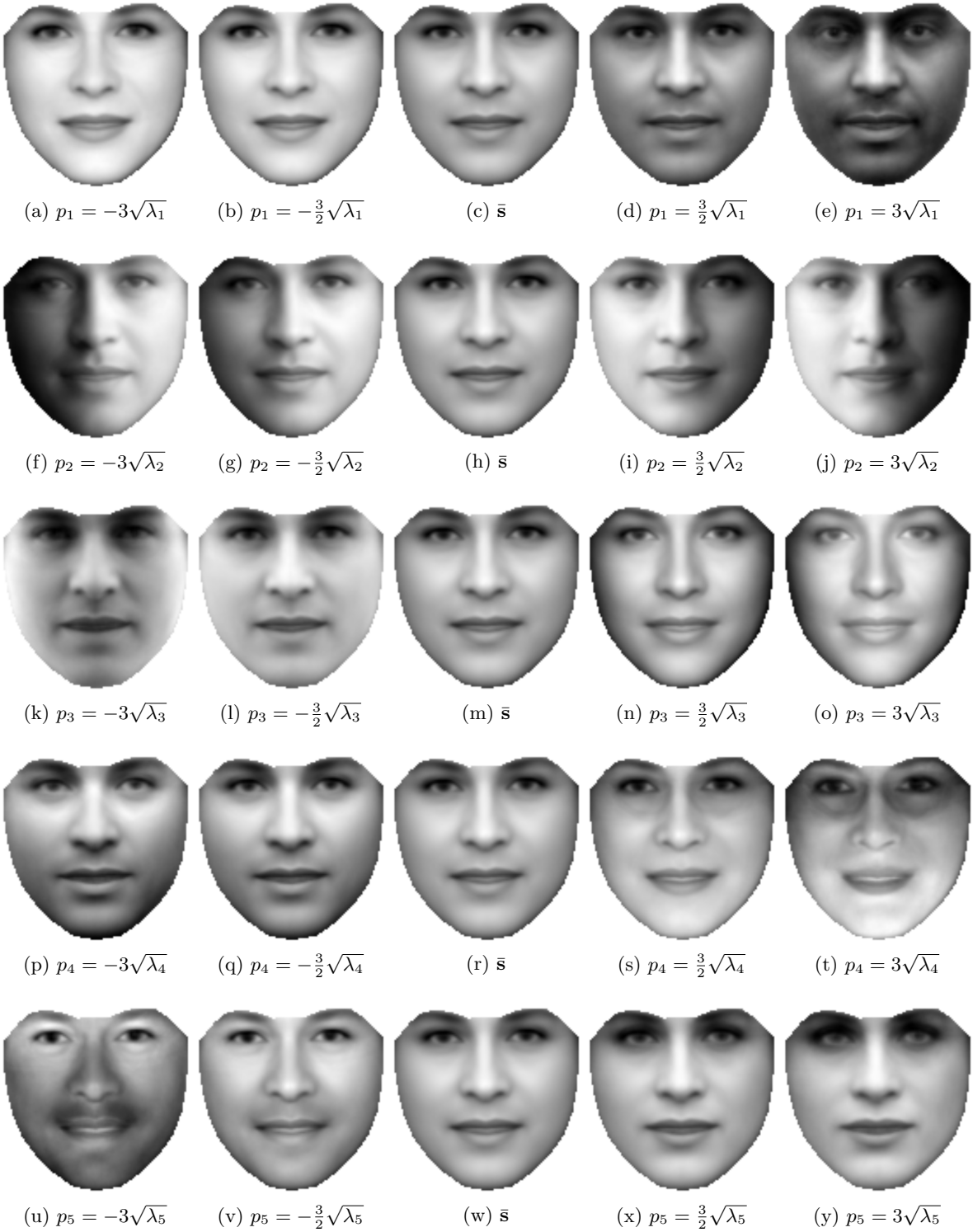


(a) LFPW trainset (b) LFPW testset (c) HELEN trainset (d) HELEN testset



(e) AFW (f) IBUG (g) 300W Indoor (h) 300W Outdoor

Figure 3.4: Exemplar images from the employed in-the-wild databases.



(a) $p_1 = -3\sqrt{\lambda_1}$ (b) $p_1 = -\frac{3}{2}\sqrt{\lambda_1}$ (c) $\bar{\mathbf{s}}$ (d) $p_1 = \frac{3}{2}\sqrt{\lambda_1}$ (e) $p_1 = 3\sqrt{\lambda_1}$

(f) $p_2 = -3\sqrt{\lambda_2}$ (g) $p_2 = -\frac{3}{2}\sqrt{\lambda_2}$ (h) $\bar{\mathbf{s}}$ (i) $p_2 = \frac{3}{2}\sqrt{\lambda_2}$ (j) $p_2 = 3\sqrt{\lambda_2}$

(k) $p_3 = -3\sqrt{\lambda_3}$ (l) $p_3 = -\frac{3}{2}\sqrt{\lambda_3}$ (m) $\bar{\mathbf{s}}$ (n) $p_3 = \frac{3}{2}\sqrt{\lambda_3}$ (o) $p_3 = 3\sqrt{\lambda_3}$

(p) $p_4 = -3\sqrt{\lambda_4}$ (q) $p_4 = -\frac{3}{2}\sqrt{\lambda_4}$ (r) $\bar{\mathbf{s}}$ (s) $p_4 = \frac{3}{2}\sqrt{\lambda_4}$ (t) $p_4 = 3\sqrt{\lambda_4}$

(u) $p_5 = -3\sqrt{\lambda_5}$ (v) $p_5 = -\frac{3}{2}\sqrt{\lambda_5}$ (w) $\bar{\mathbf{s}}$ (x) $p_5 = \frac{3}{2}\sqrt{\lambda_5}$ (y) $p_5 = 3\sqrt{\lambda_5}$

Figure 3.5: Exemplar instances of a holistic statistical appearance model trained on the images of LFPW trainset. Each row shows the variations covered by the first five principal components, where λ_i is the eigenvalue that corresponds to the i-th eigenvector.

(a) $p_1 = -3\sqrt{\lambda_1}$ (b) $p_1 = -\frac{3}{2}\sqrt{\lambda_1}$ (c) $\bar{\mathbf{s}}$ (d) $p_1 = \frac{3}{2}\sqrt{\lambda_1}$ (e) $p_1 = 3\sqrt{\lambda_1}$

(f) $p_2 = -3\sqrt{\lambda_2}$ (g) $p_2 = -\frac{3}{2}\sqrt{\lambda_2}$ (h) $\bar{\mathbf{s}}$ (i) $p_2 = \frac{3}{2}\sqrt{\lambda_2}$ (j) $p_2 = 3\sqrt{\lambda_2}$

(k) $p_3 = -3\sqrt{\lambda_3}$ (l) $p_3 = -\frac{3}{2}\sqrt{\lambda_3}$ (m) $\bar{\mathbf{s}}$ (n) $p_3 = \frac{3}{2}\sqrt{\lambda_3}$ (o) $p_3 = 3\sqrt{\lambda_3}$

(p) $p_4 = -3\sqrt{\lambda_4}$ (q) $p_4 = -\frac{3}{2}\sqrt{\lambda_4}$ (r) $\bar{\mathbf{s}}$ (s) $p_4 = \frac{3}{2}\sqrt{\lambda_4}$ (t) $p_4 = 3\sqrt{\lambda_4}$

(u) $p_5 = -3\sqrt{\lambda_5}$ (v) $p_5 = -\frac{3}{2}\sqrt{\lambda_5}$ (w) $\bar{\mathbf{s}}$ (x) $p_5 = \frac{3}{2}\sqrt{\lambda_5}$ (y) $p_5 = 3\sqrt{\lambda_5}$

Figure 3.6: Exemplar instances of a part-based statistical appearance model trained on the images of LFPW trainset. Each row shows the variations covered by the first five principal components, where λ_i is the eigenvalue that corresponds to the i-th eigenvector.

Part I

Generative Deformable Models

Chapter 4

Feature-based Lucas-Kanade and Active Appearance Models

Contents

4.1 Motivation

As explained in Sec. 2.1, the Lucas-Kanade (LK) algorithm [111, 18] is the most important method for the problem of aligning a given image with a template image. The method's aim is to find the parameter values of a parametric motion model (commonly an affine transform) that minimize the discrepancies between the two images. Active Appearance Models (AAMs) [39] are among the most popular models for the task of face fitting. They are generative Deformable Models of shape and appearance variation. Among the most efficient techniques to optimize AAMs is Gauss-Newton, which recovers the parametric description of a face instance. Gauss-Newton optimization for AAMs is similar to the LK algorithm, with the difference that the registration is obtained between the input image and a parametric appearance model instead of a static template.

The most common choice for both LK and AAMs fitting is the Inverse Compositional (IC) image alignment algorithm [18, 117]. IC is a non-linear, Gauss-Newton optimization technique that aims to minimize the ℓ_2 norm between the warped image texture and a target texture. The target texture is the static template image in the case of LK and a model texture instance in the case of AAMs.

Since IC is a Gauss-Newton method, the registration result is sensitive to initialization and to large appearance variations in terms of illumination, expressions, occlusion, identity, etc. exposed in the input and the target images [16]. Especially, in the case of intensities-based AAMs with the Project-Out IC algorithm [117], the model is incapable of adequately generalizing in order to be robust to outliers. This is the main reason why AAMs have been criticized of not being adequate for generic alignment applications and only being capable of performing well under person specific scenarios.

In this chapter, we propose the employment of highly-descriptive, *dense* appearance features for both LK and holistic AAMs. We show that even though the employment of dense features increases the data dimensionality, there is a small raise in the time complexity and a significant improvement in the alignment accuracy. We show that within the IC optimization, there is no need to compute the dense features at each iteration from the warped image. On the contrary, we extract the dense features from the original image once and then warp the resulting multi-channel image at each iteration. This strategy gives better results, as shown in our motivating experiment of Sec. 4.5.1 and has smaller computational complexity, as explained in Sec. 4.4 and Tab. 4.2. Motivated by this observation, we present very accurate and robust experimental results for both face alignment and fitting with feature-based LK and AAMs, that prove their invariance to illumination and expression changes and their generalization ability to unseen faces.

We apply the above concept for both LK and holistic AAMs by using a great variety of widely-used features, such as Histograms of Oriented Gradients (HOG) [46], Scale-Invariant Feature Transform (SIFT) [109], Image Gradient Orientation kernel (IGO) [158, 157], Edge Structure (ES) [41], Local Binary Patterns (LBP) [120, 121, 122] with variations [169], and Gabor filters [91, 92, 102]. We extensively evaluate the performance and behavior of the proposed framework on the commonly used Yale B Database [65] for LK and on multiple in-the-wild databases (LFPW [22], AFW [185], HELEN [97], iBUG [133]) for AAMs. Finally, we compare with two state-of-the-art discriminative Deformable Models [171, 13] and report more accurate results.

To summarize, the contributions of this work are:

- We propose the incorporation of densely-sampled, highly-descriptive features in the IC gradient descent framework. We show that the combination of *(i)* non-linear least-squares optimization with *(ii)* robust features (*e.g.*, HOG, SIFT) and *(iii)* generative models can achieve excellent performance for the task of face alignment.

- We elaborate on the reasons why it is preferable to warp the features image at each iteration, rather than extracting features at each iteration from the warped image, as it is done in the relevant bibliography.

- Our extended experimental results provide solid comparisons between some of the most successful and widely-used features that exist in the current bibliography for the tasks of interest, by thoroughly investigating the features' accuracy, robustness, and speed of convergence.

- Our proposed HOG and SIFT holistic AAMs outperform state-of-the-art face fitting methods on a series of cross-database challenging in-the-wild experiments.

The content of this chapter is based on the following publications:

- **E. Antonakos**, J. Alabort-i-Medina, G. Tzimiropoulos, and S. Zafeiriou. "HOG Active Appearance Models", *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Paris, France, pp. 224-228, October 2014.

- **E. Antonakos**, J. Alabort-i-Medina, G. Tzimiropoulos, and S. Zafeiriou. "Feature-Based Lucas-Kanade and Active Appearance Models", *IEEE Transactions on Image Processing (T-IP)*, 24(9): pp. 2617-2632, September 2015.

The rest of the chapter is structured as follows: Section 4.2 briefly describes the used features. Section 4.3 elaborates on the intensity-based IC algorithm for LK and AAMs. Section 4.4 explains the strategy to combine the IC optimization with dense features. Finally, Section 4.5 presents extended experiments for LK and AAMs and Section 4.6 draws the conclusions.

4.2 Image Features

A feature-based image representation is achieved with the application of a feature extraction function, as defined in Eq. 3.7. In this work, we require the descriptor function to extract densely-sampled image features, thus compute a feature vector for each pixel location. Given an input image of size $H \times W$ in vectorial form $\mathbf{t}$ with length $L_T = HW$, the descriptor-based image vector is

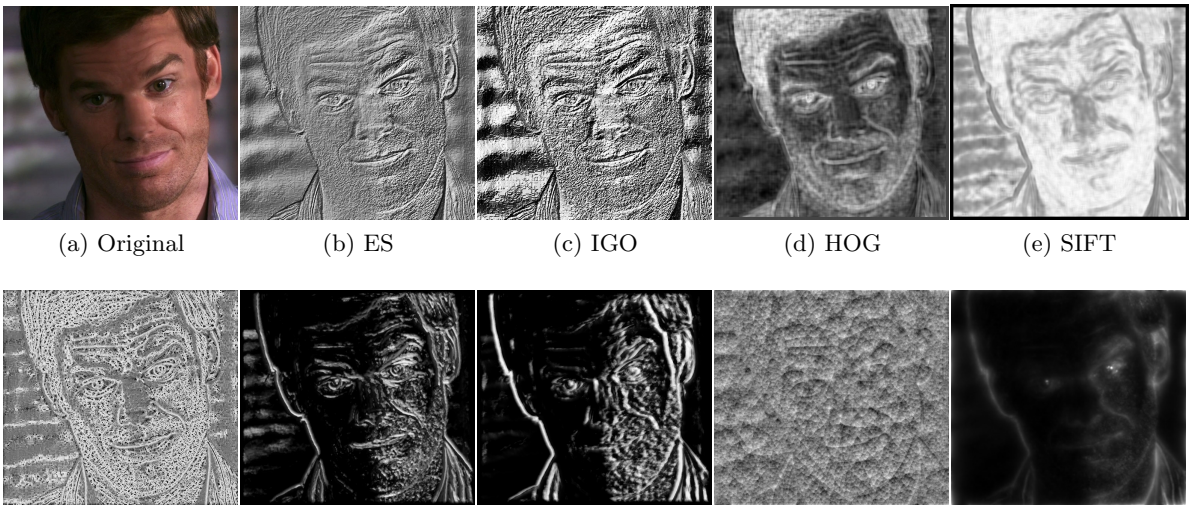
$$\mathbf{f} = \mathcal{F}(\mathbf{t}) \tag{4.1}$$

with size $L_T D \times 1$, where D is the number of channels. In the rest of the chapter, we will denote the images in vectorized form within the equations.

Many robust multi-dimensional image descriptors have been proposed and applied to various tasks. They can be divided in two categories: those extracted based only on the pixel values and those extracted based on larger spatial neighborhoods. They all aim to generate features that are invariant to translation, rotation, scale and illumination changes and robust to local geometric distortion. We select nine of the most powerful and successful descriptors, which are briefly described in the following subsections (4.2.1–4.2.6). Figure 4.1 shows the feature-based image representation for each of the employed feature types. The visualized grayscale images are constructed by summing all the D channels of the feature images. Notice how each descriptor handles the illumination changes and the face's distinctive edges. Table 4.1 summarizes the parameter values, the number of channels and the neighborhood size that gets involved in computing the descriptor at each image location for all features.

4.2.1 Edge Structure (ES)

ES, initially proposed in [41], is a measure which captures the orientation of image structure at each pixel, together with an indication of how accurate the orientation estimate is. The accuracy belief measure penalizes the orientations in flat, noisy regions and favors the ones near strong edges. The first step of the ES features computation involves the estimation of the local gradients with respect to x and y, denoted by $\mathbf{g}_x$ and $\mathbf{g}_y$, and the calculation of the gradient magnitude $\mathbf{g} = \sqrt{\mathbf{g}_x^2 + \mathbf{g}_y^2}$. Then $\mathbf{f} = f(\mathbf{g})[\mathbf{g}_x, \mathbf{g}_y]$ is evaluated, where $f(\mathbf{g}) = |\mathbf{g}|/(|\mathbf{g}| + \bar{g})$ is a non-linear normalization function ($\bar{g}$ is the mean of $\mathbf{g}$). This feature-based representation has $D = 2$ channels and is effective at favoring strong and distinctive edges (Fig. 4.1b).



(a) Original (b) ES (c) IGO (d) HOG (e) SIFT

(f) OLBP (g) TPLBP (h) FPLBP (i) Gabor Angles (j) Gabor Magnitude

Figure 4.1: Examples of the nine employed dense feature types. The feature images have the same height and width as the original image and D channels. In order to visualize them, we compute the sum over all D channels.

4.2.2 Image Gradient Orientation (IGO)

IGO is introduced and successfully applied in [153, 157, 158, 154]. Given the gradients $\mathbf{g}_x$, $\mathbf{g}_y$ of an input image and their orientation $\boldsymbol{\varphi}$, we compute the IGO image as $\mathbf{f} = \frac{1}{\sqrt{L_T}}[\cos\boldsymbol{\varphi}^\mathsf{T}, \sin\boldsymbol{\varphi}^\mathsf{T}]^\mathsf{T}$, where L_T is the length of the input image and $\cos\boldsymbol{\varphi} = [\cos\boldsymbol{\varphi}(1), \ldots, \cos\boldsymbol{\varphi}(L_T)]^\mathsf{T}$ (the same for $\sin\boldsymbol{\varphi}$). The above feature image definition results in $D = 2$ channels. IGO features allow us to estimate the similarity between two images as $s = \mathbf{f}_1^\mathsf{T}\mathbf{f}_2$. This measure becomes $s \approx 0$ for the image areas that are corrupted by outliers (e.g. occlusion) and thus behaves similarly to weighted least-squares kernel without the need of information regarding the structure of outliers. This reveals the advantage of this feature. IGO is robust to outliers while at the same time being low-dimensional compared to other robust features (Fig. 4.1c).

4.2.3 Histograms of Oriented Gradients (HOG)

HOG descriptors [46] cluster the gradient orientations in different bins for localized subwindows of an input image resulting in counting occurrences of the orientations. Thus, the shape and texture of the image are described by histograms of local edge directions, which are also characterized by photometric invariance. The HOG features extraction begins by computing the image gradient. If the image is color, then the gradient with the largest norm between the three channels is kept. Two spatial neighborhoods are used at the region of each

pixel: cells and blocks. A cell is a small sub-window from which we create a histogram of the gradient's orientations weighted by the gradient magnitude. The histogram has N_{bins} bins and trilinear interpolation is applied between the votes of neighboring bin centers with respect to orientation and position. A block is a larger spatial region that consists of $N_{block} \times N_{block}$ cells. We apply contrast normalization between the cells that are grouped within a block, based on the Euclidean norm. The final descriptor vector extracted from each block is composed by concatenating the normalized histograms of the cells, thus it has length $D = N_{bins}N_{block}^2$. In the default HOG formulation, the block can be regarded as a sliding window that scans the locations of an image with a sampling step of either a block (no overlap) or half a block (overlapping windows). On the contrary, the computed feature image in our case is dense, which means that we use a sampling step of one pixel and we extract a descriptor vector from the block centered at each such location. This ends up in a very powerful representation that is descriptive on the important facial parts and flat on the rest of the face (Fig. 4.1d). By using cells of size 8×8 pixels with $N_{block} = 2$ and $N_{bins} = 9$, we have $D = 36$ channels.

4.2.4 Scale-Invariant Feature Transform (SIFT)

SIFT features, originally proposed in [109], are computed locally based on the appearance of particular interest points (keypoints). In the original SIFT formulation, these keypoints are detected as the maxima and minima of the Difference of Gaussians applied in the scale space of an image. The scale space is constructed by convolving the image with Gaussian filters at different scales (and octaves). The keypoints with dominant orientations are kept and the points that have low contrast or lie along an edge are ignored. Then SIFT descriptors are obtained by taking into account neighboring pixels within a radius for a keypoint. Thus, the traditional SIFT framework returns a sparse feature map of an image, which is not useful in our case. Similar to the HOG case, in our framework, we skip the keypoint detection step and extract a SIFT descriptor vector for each image location.

We begin by assigning a dominant orientation to each pixel. Assume that $\mathbf{L}(x, y, \sigma) = \mathbf{G}(x, y, \sigma) * \mathbf{T}(x, y)$ is the Gaussian-smoothed image at the scale σ of the location (x, y). We calculate the gradient magnitude and direction for every pixel in a neighborhood around the point in $\mathbf{L}$ and form an orientation histogram, where each orientation is weighted by the corresponding gradient magnitude and by a Gaussian-weighted circular window with standard deviation proportional to the pixel's σ. Then, we take the orientations that are within a percentage (80%) of the highest bin. If these orientations are more than one, then we create multiple points and assign them each orientation value. Eventually, the final descriptor vector

is created by sampling the neighboring pixels at the image $\mathbf{L}(x, y, \sigma)$ with scale closest to the point's scale, rotating the gradients and coordinates by the previously computed dominant orientation, separating the neighborhood in $N_{block} \times N_{block}$ sub-regions and create a Gaussian-weighted orientations histogram for each sub-region with N_{bins} bins. Finally, the histograms are concatenated in a single vector with length $D = N_{bins}N_{block}^2$ that is normalized to unit length. The SIFT descriptor is similar to the HOG one, with the difference that the orientations histograms are computed with respect to each point's dominant orientation. In general, SIFT are invariant to scale, rotation, illumination and viewpoint (Fig. 4.1e). We use the same parameters as in HOGs ($N_{block} = 2$, $N_{bins} = 9$ and 8×8 cells), thus $D = 36$ channels.

4.2.5 Local Binary Patterns (LBP)

The basic idea behind LBP [120, 121, 122] is to encode the local structure in an image by comparing each pixel's intensity value with the pixel intensities within its neighborhood. For each pixel, we define a neighborhood radius r centered at the pixel and compare the intensities of S circular sample points to its intensity. The sampling is done clockwise or counter-clockwise, starting from a specific angle, and we apply interpolation on sample points that are not discrete. If the center pixel's intensity is greater or equal than the sample's, then we denote it by 1, otherwise by 0. Thus, we end up with a binary number (LBP code) for each pixel, with S digits and 2^S possible combinations, which is converted to decimal. In the original LBP formulation, the output is a descriptor vector describing the whole image with a normalized histogram of the decimal codes. We instead use N_{radius} number of values for the radius parameter, r. Then we sample $N_{samples}$ sets of points S from the circle of each radius value and concatenate the LBP codes in a vector. This means that our dense feature image has $D = N_{radius}N_{samples}$ channels. We also employ the extension of rotation-invariant uniform LBPs. Uniform LBPs are binary codes with at most two circular 0-1 and 1-0 transitions. In the computation of the final LBP patterns, there is a separate label for each uniform code and all the non-uniform codes are labeled with a single label. By setting $r = \{1, 2, \ldots, 8\}$ ($N_{radius} = 8$) and sampling $N_{samples} = 8$ points for each radius value, we end up with $D = 8$ channels.

Moreover, apart from the original LBP, which we denote by OLBP, we also use the variations of Three-Patch LBP (TPLBP) and Four-Patch LBP (FPLBP), introduced in [169]. TPLBP and FPLBP encode in the binary codes the similarities between neighboring patches (for details, please refer to [169]). Thus, the number of channels in this case also depends on the employed number of patches N_{patch} with different sizes, hence $D = N_{radius}N_{samples}N_{patch}$.

Feature Type	*Parameters Values*	*Neighbourhood Size (in pixels)*	*Channels (D)*
IGO, ES	–	–	2
HOG SIFT	$N_{bins} = 9$, $N_{cell} = 2$ $cell = 8 \times 8$ pixels	256	36
OLBP [a]	$N_{radius} = 8$, $N_{samples} = 8$	64	8
TPLBP [a] FPLBP [b]	$N_{radius} = 8$, $N_{samples} = 8$ $N_{patch} = 2$	64	16
Gabor	$N_{sc} = 4$, $N_{or} = 9$	–	36

[a] Radius takes values $\{1, 2, \ldots, 8\}$, patch sizes are 2 and 4 and for each radius we sample a single set of 8 points.

[b] Inner and outer radius are $\{[1, 5], [2, 6], \ldots, [8, 12]\}$, patch sizes are 2 and 4 and for each radius we sample a single set of 8 points.

Table 4.1: Characteristics of the nine employed dense feature types. The characteristics include the features' parameters values, neighborhood size that contributes in each pixel's computation and number of channels.

With the parameters we use, we end up with $D = 16$ channels. The three LBP derivatives are visualized in Figs. 4.1f-4.1h.

4.2.6 Gabor Magnitude and Angle

Herein, we employ the log-Gabor filter (wavelet) [91, 92, 102]. In the log-polar coordinates of the Fourier domain (ρ, θ), this is defined as $G_{(s,o)}(\rho, \theta) = \exp\left(-\frac{1}{2}\left(\frac{\rho-\rho_s}{\sigma_\rho}\right)^2\right) \exp\left(-\frac{1}{2}\left(\frac{\theta-\theta_{(s,o)}}{\sigma_\theta}\right)^2\right)$, where σ_ρ and σ_θ are the bandwidths in ρ and θ respectively and (s, o) are the indexes of each filter's scale and orientation. Thus, by using N_{sc} scales and N_{or} orientations, we have a filterbank of log-Gabor filters with $s = 1, \ldots, N_{sc}$ and $o = 1, \ldots, N_{or}$. The reason why log-Gabor filter is preferred over Gabor is that it has no DC component and its transfer function is extended at a high frequency range. Given an image, we compute its convolution with each log-Gabor filter for all scales and orientations. Then, we create two feature images by concatenating the convolution's magnitude and phase, respectively. Both feature versions have $D = N_{sc}N_{or}$ channels. An example of the Gabor angles and magnitude is shown in Figs. 4.1i and 4.1j, respectively. We use the log-Gabor filters implementation available in [90] with $N_{sc} = 4$ and $N_{or} = 9$, thus $D = 36$.

4.2.7 Features Function Computational Complexity

As mentioned before, the presented features can be separated in two categories:

1. Features that are computed in a pixel-based fashion (*e.g.*, ES, IGO).
2. Features that are computed in a window-based mode, thus they depend on the values of a larger spatial neighborhood for each location (*e.g.*, HOG, SIFT, LBP).

Given an image $\mathbf{t}$ in vectorial form with length L_T, the computational cost of extracting dense D-channel features of the first category is $\mathcal{O}(L_T D)$. Respectively, the complexity of extracting the features of the second category, using a window of size $h \times w$ for each pixel, is $\mathcal{O}(L_T L_w D)$, where $L_w = hw$ is the window's area. However, since the window's dimensions h and w take values of the same order as D, hence $hw \approx D^2$, the cost of the second case can also be expressed as

$$\mathcal{O}(L_T D^3) \tag{4.2}$$

This gives an intuition on the complexity difference between the two cases. In the following sections, we will use the window-based features complexity of Eq.4.2 as the worst-case scenario, since it is more expensive than the pixel-based one.

4.3 Inverse-Compositional Alignment Algorithm

The optimization technique that we employ for both LK and AAMs is the efficient Gauss-Newton Inverse Compositional (IC) Image Alignment [18, 117]. In this section, we firstly refer to the problem of LK (4.3.1) and then elaborate on holistic AAMs (4.3.2). In both cases, Gauss-Newton aims to minimize an ℓ_2 norm with respect to a parametric motion model, as defined in Chapter 3, Sec. 3.2.2. The motion model utilized in this work is Piecewise Affine Warp (PWA) [43, 18], denoted as $\mathcal{W}(\mathbf{p})$, where $\mathbf{p}$ is the n_s number of parameters (Eq. 3.5). In order to explain the IC algorithm, we first present the forward-additive (FA) and forward-compositional (FC) ones. Note that all the algorithms in this section are presented based on pixel intensities, thus we assume that we have images with a single channel.

4.3.1 Lucas-Kanade Optimization

Herein, we first define the optimization techniques for the LK face alignment problem, in order to describe the IC optimization for AAMs in the following Sec. 4.3.2. The aim of image alignment is to find the location of a constant template $\bar{\mathbf{a}} \in \mathbb{R}^m$ in an input vectorized image

$\mathbf{t}$, where m is the number of pixels inside the reference shape as also defined in Sec. 3.2.2. This is mathematically expressed as minimizing the ℓ_2-norm cost function

$$\underset{\mathbf{p}}{\operatorname{argmin}} \|\bar{\mathbf{a}} - \mathbf{t}(\mathcal{W}(\mathbf{p}))\|^2 \tag{4.3}$$

with respect to the n_s motion model parameters $\mathbf{p}$. The proposed Gauss-Newton optimization techniques [16, 18] are categorized as:

- Forward
- Inverse

depending on the direction of the motion parameters estimation and

- Additive
- Compositional

depending on the way the motion parameters are updated.

Forward-Additive

Lucas and Kanade proposed the FA gradient descent in [111]. By using an additive iterative update of the parameters, *i.e.*

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p} \tag{4.4}$$

and having an initial estimate of $\mathbf{p}$, the cost function of Eq. 4.3 is expressed as minimizing

$$\underset{\Delta\mathbf{p}}{\operatorname{argmin}} \|\bar{\mathbf{a}} - \mathbf{t}(\mathcal{W}(\mathbf{p} + \Delta\mathbf{p}))\|^2 \tag{4.5}$$

with respect to $\Delta\mathbf{p}$. The solution is given by first linearizing around $\mathbf{p}$, thus using first order Taylor series expansion at $\mathbf{p} + \Delta\mathbf{p} = \mathbf{p} \Rightarrow \Delta\mathbf{p} = \mathbf{0}$. This gives

$$\mathbf{t}(\mathcal{W}(\mathbf{p} + \Delta\mathbf{p})) \approx \mathbf{t}(\mathcal{W}(\mathbf{p})) + \mathbf{J_t}|_{\mathbf{p}=\mathbf{p}}\Delta\mathbf{p} \tag{4.6}$$

where $\mathbf{J_t}|_{\mathbf{p}=\mathbf{p}} = \nabla\mathbf{t}\frac{\partial\mathcal{W}}{\partial\mathbf{p}}$ is the *image Jacobian*, consisting of the *image gradient* evaluated at $\mathcal{W}(\mathbf{p})$ and the *warp jacobian* evaluated at $\mathbf{p}$. The final solution is given by

$$\Delta\mathbf{p} = \mathbf{H}^{-1}\mathbf{J_t^\mathsf{T}}|_{\mathbf{p}=\mathbf{p}}\left[\bar{\mathbf{a}} - \mathbf{t}(\mathcal{W}(\mathbf{p}))\right] \tag{4.7}$$

where

$$\mathbf{H} = \mathbf{J_t^\mathsf{T}}|_{\mathbf{p}=\mathbf{p}}\mathbf{J_t}|_{\mathbf{p}=\mathbf{p}} \tag{4.8}$$

is the Gauss-Newton approximation of the *Hessian matrix*. This method is forward because the warp projects into the image coordinate frame and additive because the iterative update of the motion parameters is computed by estimating a $\Delta\mathbf{p}$ incremental offset from the current parameters. The algorithm is very slow with computational complexity $\mathcal{O}({n_s}^3+{n_s}^2 m)$, because the computationally costly Hessian matrix and its inverse depend on the warp parameters $\mathbf{p}$ and need to be evaluated at each iteration.

Forward-Compositional

Compared to the FA version, in the FC gradient descent we have the same warp direction for computing the parameters, but a compositional update of the form

$$\mathcal{W}(\mathbf{p}) \leftarrow \mathcal{W}(\mathbf{p}) \circ \mathcal{W}(\Delta\mathbf{p}) \tag{4.9}$$

The minimization cost function in this case takes the form

$$\underset{\Delta\mathbf{p}}{\operatorname{argmin}} \left\|\bar{\mathbf{a}} - \mathbf{t}\left(\mathcal{W}(\mathbf{p}) \circ \mathcal{W}(\Delta\mathbf{p})\right)\right\|^2 \tag{4.10}$$

and the linearization is

$$\left\|\bar{\mathbf{a}} - \mathbf{t}(\mathcal{W}(\mathbf{p})) - \mathbf{J_t}|_{\Delta\mathbf{p}=\mathbf{0}}\Delta\mathbf{p}\right\|^2 \tag{4.11}$$

where the composition with the identity warp is $\mathcal{W}(\mathbf{p}) \circ \mathcal{W}(\mathbf{0}) = \mathcal{W}(\mathbf{p})$. The image Jacobian in this case is expressed as $\mathbf{J_t}|_{\mathbf{p}=\mathbf{0}} = \nabla\mathbf{t}(\mathcal{W}(\mathbf{p}))\left.\frac{\partial\mathcal{W}}{\partial\mathbf{p}}\right|_{\mathbf{p}=\mathbf{0}}$. Thus, with this formulation, the warp Jacobian is constant and can be precomputed, because it is evaluated at $\mathbf{p} = \mathbf{0}$. This precomputation slightly improves the algorithm's computational complexity compared to the FA case, even though the compositional update is more expensive than the additive one.

Inverse-Compositional

In the IC optimization, the direction of the warp is reversed compared to the two previous techniques and the incremental warp is computed with respect to the template $\bar{\mathbf{a}}$ [18, 17]. Compared to Eq. 4.3 the goal in this case is to minimize

$$\underset{\Delta\mathbf{p}}{\operatorname{argmin}} \left\|\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}}(\mathcal{W}(\Delta\mathbf{p}))\right\|^2 \tag{4.12}$$

with respect to $\Delta\mathbf{p}$. The incremental warp $\mathcal{W}(\Delta\mathbf{p})$ is computed with respect to the template $\bar{\mathbf{a}}$, but the current warp $\mathcal{W}(\mathbf{p})$ is still applied on the input image. By linearizing around $\Delta\mathbf{p} = \mathbf{0}$ and using the identity warp, we have

$$\left\|\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}} - \mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}}\Delta\mathbf{p}\right\|^2 \tag{4.13}$$

where $\mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}} = \nabla\bar{\mathbf{a}} \left.\frac{\partial \mathcal{W}}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{0}}$. Consequently, similar to the FC case, the increment is

$$\Delta\mathbf{p} = \mathbf{H}^{-1}\mathbf{J}_{\bar{\mathbf{a}}}^{\mathsf{T}}|_{\mathbf{p}=\mathbf{0}} \left[\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}}\right] \tag{4.14}$$

where the Hessian matrix is

$$\mathbf{H} = \mathbf{J}_{\bar{\mathbf{a}}}^{\mathsf{T}}|_{\mathbf{p}=\mathbf{0}}\mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}} \tag{4.15}$$

The compositional motion parameters update at each iteration is

$$\mathcal{W}(\mathbf{p}) \leftarrow \mathcal{W}(\mathbf{p}) \circ \mathcal{W}(\Delta\mathbf{p})^{-1} \tag{4.16}$$

Since the gradient is always taken at the template, the warp Jacobian $\left.\frac{\partial \mathcal{W}}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{0}}$ and thus the Hessian matrix's inverse remain constant and can be precomputed. This makes the IC algorithm both fast and efficient with a total computational complexity of $\mathcal{O}(n_s^2 + n_s m)$.

4.3.2 Active Appearance Models Optimization

AAMs are statistical Deformable Models of shape and appearance that recover a parametric description of a certain object through optimization. Their shape and appearance models are linear statistical models built as explained in Secs. 3.1 and 3.2, respectively. These models can be used to generate new shape and appearance instances as shown in Eqs. 3.4 and 3.13, respectively. Note that the appearance model utilized in this chapter employs a holistic appearance representation.

The basic difference between the IC algorithm employed for LK and AAMs is that the template image $\bar{\mathbf{a}}$ is not static, but it includes a linear appearance variation controlled by the appearance parameters $\mathbf{c}$ as shown in Eq. 3.13. Consequently, the minimization cost function of Eq. 4.3 now becomes

$$\underset{\mathbf{p},\mathbf{c}}{\operatorname{argmin}} \|\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}} - \mathbf{U}_a\mathbf{c}\|^2 \tag{4.17}$$

We present three algorithms for solving the optimization problem: Simultaneous, Alternating and Project-Out.

Project-Out Inverse-Compositional

The Project-Out IC (POIC) algorithm [117] decouples shape and appearance by solving Eq. 4.17 in a subspace orthogonal to the appearance variation. This is achieved by "projecting-out" the appearance variation, thus working on the orthogonal complement of the appearance subspace $\hat{\mathbf{U}}_a = \mathbf{E} - \mathbf{U}_a\mathbf{U}_a^{\mathsf{T}}$. The cost function of Eq. 4.17 takes the form

$$\underset{\Delta\mathbf{p}}{\operatorname{argmin}} \|\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}}(\mathcal{W}(\Delta\mathbf{p}))\|^2_{\mathbf{E}-\mathbf{U}_a\mathbf{U}_a^{\mathsf{T}}} \tag{4.18}$$

and the first-order Taylor expansion over $\Delta\mathbf{p} = \mathbf{0}$ is

$$\bar{\mathbf{a}}(\mathcal{W}(\Delta\mathbf{p})) \approx \bar{\mathbf{a}} + \mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}}\Delta\mathbf{p} \tag{4.19}$$

The incremental update of the warp parameters is computed as

$$\Delta\mathbf{p} = \mathbf{H}^{-1}\mathbf{J}_{POIC}^{\mathsf{T}}[\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}}] \tag{4.20}$$

where

$$\mathbf{J}_{POIC} = (\mathbf{E} - \mathbf{U}_a\mathbf{U}_a^{\mathsf{T}})\mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}} \tag{4.21}$$

and

$$\mathbf{H}^{-1} = \mathbf{J}_{POIC}^{\mathsf{T}}\mathbf{J}_{POIC} \tag{4.22}$$

The appearance parameters can be retrieved at the end of the iterative operation as

$$\mathbf{c} = \mathbf{U}_a^{\mathsf{T}}[\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}}] \tag{4.23}$$

in order to reconstruct the appearance vector. The POIC algorithm is very fast with $\mathcal{O}(n_s m + n_s^2)$ computational complexity, because the Jacobian, the Hessian matrix and its inverse are constant and can be precomputed. However, the algorithm is not robust, especially in cases with large appearance variation or outliers.

Simultaneous Inverse-Compositional

In the Simultaneous IC (SIC) [68] we aim to optimize simultaneously for $\mathbf{p}$ and $\mathbf{c}$ parameters. Similar to the Eq. 4.12 of the LK-IC case, the cost function of Eq. 4.17 now becomes

$$\underset{\Delta\mathbf{p},\Delta\mathbf{c}}{\operatorname{argmin}} \left\|\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}}\left(\mathcal{W}(\Delta\mathbf{p})\right) - \mathbf{U}_a\left(\mathcal{W}(\Delta\mathbf{p})\right)(\mathbf{c} + \Delta\mathbf{c})\right\|^2 \tag{4.24}$$

We denote by

$$\Delta\mathbf{q} = \left[\Delta\mathbf{p}^{\mathsf{T}}, \Delta\mathbf{c}^{\mathsf{T}}\right]^{\mathsf{T}} \tag{4.25}$$

the vector of concatenated parameters increments with length $n_s + n_a$. As in Eq. 4.13, the linearization of the model term around $\Delta\mathbf{p} = \mathbf{0}$ consists of two parts: the mean appearance vector approximation

$$\bar{\mathbf{a}}(\mathcal{W}(\Delta\mathbf{p})) \approx \bar{\mathbf{a}} + \mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}}\,\Delta\mathbf{p} \tag{4.26}$$

and the linearized basis

$$\mathbf{U}_a(\mathcal{W}(\Delta\mathbf{p})) \approx \mathbf{U}_a + \left[\mathbf{J}_{\mathbf{u}_1}|_{\mathbf{p}=\mathbf{0}}\Delta\mathbf{p}, \ldots, \mathbf{J}_{\mathbf{u}_{n_a}}|_{\mathbf{p}=\mathbf{0}}\Delta\mathbf{p}\right] \tag{4.27}$$

where $\mathbf{J}_{\mathbf{u}_i}|_{\mathbf{p}=\mathbf{0}} = \nabla \mathbf{u}_i \left.\frac{\partial \mathcal{W}}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{0}}$ denotes the Jacobian with respect to the i^{th} eigentexture at $\Delta\mathbf{p} = \mathbf{0}$. Then the final solution at each iteration is

$$\Delta\mathbf{q} = \mathbf{H}^{-1}\mathbf{J}_{SIC}^{\mathsf{T}}\left[\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}} - \mathbf{U}_a\mathbf{c}\right] \tag{4.28}$$

where the Jacobian is given by

$$\mathbf{J}_{SIC} = \left[\mathbf{J}_{\mathbf{a_c}}|_{\mathbf{p}=\mathbf{0}}, \mathbf{U}_a\right] \tag{4.29}$$

with

$$\mathbf{J}_{\mathbf{a_c}}|_{\mathbf{p}=\mathbf{0}} = \mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}} + \sum_{i=1}^{n_a} c_i \mathbf{J}_{\mathbf{u}_i}|_{\mathbf{p}=\mathbf{0}} \tag{4.30}$$

and the Hessian matrix is

$$\mathbf{H} = \mathbf{J}_{SIC}^{\mathsf{T}}\mathbf{J}_{SIC} \tag{4.31}$$

At every iteration, we apply the compositional motion parameters update of Eq. 4.16 of the LK-IC and an additive appearance parameters update

$$\mathbf{c} \leftarrow \mathbf{c} + \Delta\mathbf{c} \tag{4.32}$$

The individual Jacobians $\mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}}$ and $\mathbf{J}_{\mathbf{u}_i}|_{\mathbf{p}=\mathbf{0}}$, $\forall i = 1, \ldots, n_a$ are constant and can be pre-computed. However, the total Jacobian $\mathbf{J}_{\mathbf{a_c}}|_{\mathbf{p}=\mathbf{0}}$ and hence the Hessian matrix depend on the current estimate of the appearance parameters $\mathbf{c}$, thus they need to be computed at every iteration. This makes the algorithm very slow with a total cost of $\mathcal{O}((n_s+n_a)^2 m + (n_s+n_a)^3)$.

Alternating Inverse-Compositional

The Alternating IC (AIC) algorithm, proposed in [124, 155], instead of minimizing the cost function simultaneously for both shape and appearance as in the SIC algorithm, it solves two separate minimization problems, one for the shape and one for the appearance parameters, in an alternating fashion. That is

$$\begin{cases} \underset{\Delta\mathbf{p}}{\operatorname{argmin}} \|\mathbf{t}(\mathcal{W}(\mathbf{p})) - \mathbf{a_c}(\mathcal{W}(\Delta\mathbf{p}))\|^2_{\mathbf{E}-\mathbf{U}_a\mathbf{U}_a^{\mathsf{T}}} \\ \underset{\Delta\mathbf{c}}{\operatorname{argmin}} \|\mathbf{t}(\mathcal{W}(\mathbf{p})) - \mathbf{a}_{\mathbf{c}+\Delta\mathbf{c}}(\mathcal{W}(\Delta\mathbf{p}))\|^2 \end{cases} \tag{4.33}$$

The minimization in every iteration is achieved by first using a fixed estimate of $\mathbf{c}$ to compute the current estimate of the increment $\Delta\mathbf{p}$ and then using the fixed estimate of $\mathbf{p}$ to compute the increment $\Delta\mathbf{c}$. More specifically, similar to the previous cases and skipping the linearization

steps, given the current estimate of $\mathbf{c}$, the warp parameters increment is computed from the first cost function as

$$\Delta\mathbf{p} = \mathbf{H}^{-1}\mathbf{J}_{AIC}^{\mathsf{T}}\left[\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}} - \mathbf{U}_a\mathbf{c}\right] \tag{4.34}$$

where

$$\mathbf{J}_{AIC} = (\mathbf{E} - \mathbf{U}_a\mathbf{U}_a^{\mathsf{T}})\left[\mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}} + \sum_{i=1}^{n_a} c_i\mathbf{J}_{\mathbf{u}_i}|_{\mathbf{p}=\mathbf{0}}\right] \tag{4.35}$$

and

$$\mathbf{H}^{-1} = \mathbf{J}_{AIC}^{\mathsf{T}}\mathbf{J}_{AIC} \tag{4.36}$$

Then, given the current estimate of the motion parameters $\mathbf{p}$, AIC computes the appearance parameters as the least-squares solution of the second cost function of Eq. 4.33, thus

$$\Delta\mathbf{c} = \mathbf{U}_a^{\mathsf{T}}\left[\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}}(\mathcal{W}(\Delta\mathbf{p})) - \mathbf{U}_a(\mathcal{W}(\Delta\mathbf{p}))\mathbf{c}\right] \tag{4.37}$$

This alternating optimization is repeated at each iteration. The motion parameters are compositionally updated as in Eq. 4.16 and the appearance parameters are updated in an additive mode, *i.e.*

$$\mathbf{c} \leftarrow \mathbf{c} + \Delta\mathbf{c} \tag{4.38}$$

AIC algorithm is slower than POIC, but more accurate as it also optimizes with respect to the appearance variance. Although the individual Jacobians $\mathbf{J}_{\mathbf{u}_i}|_{\mathbf{p}=\mathbf{0}}$, $\forall i = 1, \ldots, n_a$ and $\mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}}$ can be precomputed, the total Jacobian $\mathbf{J}_{AIC}$ and the Hessian need to be evaluated at each iteration. Following the Hessian matrix computation technique proposed in [124], which improves the cost from $\mathcal{O}(n_s^2 m)$ to $\mathcal{O}(n_s^2 n_a^2)$ (usually $m > n_a^2$) and taking into account the Hessian inversion ($\mathcal{O}(n_s^3)$), the total cost at each iteration is $\mathcal{O}(n_s^2 n_a^2 + (n_s + n_a)m + n_s^3)$.

Recently it was shown that AIC and SIC are theoretically equivalent (*i.e.*, Eqs. 4.34, 4.37 are exactly the same as Eq. 4.28) and that the only difference is their computational costs [155]. That is the SIC algorithm requires to invert the Hessian of the concatenated shape and texture parameters ($O((n_s + n_a)^3)$). However, using the fact that

$$min_{x,y} f(x, y) = min_x\left(min_y f(x, y)\right) \tag{4.39}$$

and solving first for the texture parameter increments, it was shown that

1. the complexity of SIC can be reduced dramatically, and
2. SIC is equivalent to AIC algorithm [155] (similar results can be shown by using the Schur's complement of the Hessian of texture and shape parameters).

4.4 Feature-Based Optimization

In this section we describe the combination of the IC algorithm with the feature-based appearance of Eq. 4.1. The keypoint of this combination is that there are two different ways of conducting the composition of the features function $\mathcal{F}$ and the warp function $\mathcal{W}$ on an image. Given an image $\mathbf{t}$ and the warp parameters $\mathbf{p}$, the warped feature-based image $\mathbf{f}$ can be obtained with the two following composition directions:

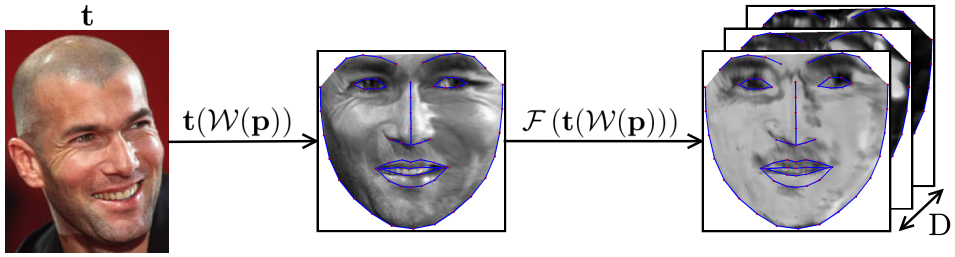
- *Features from warped image:*

$$\mathbf{f} = \mathcal{F}\left(\mathbf{t}(\mathcal{W}(\mathbf{p}))\right) \tag{4.40}$$

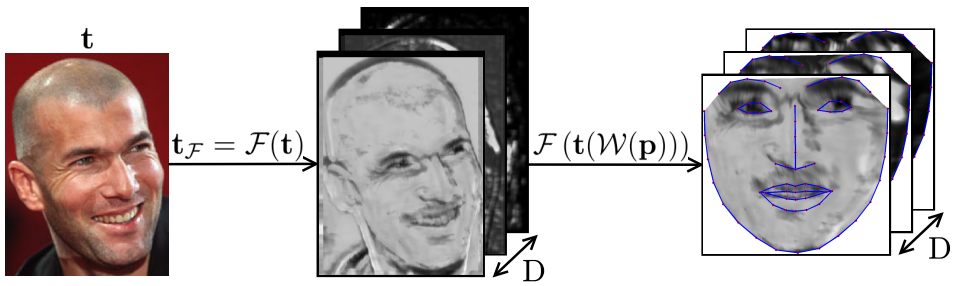
- *Warping on features image:*

$$\mathbf{f} = \mathbf{t}_{\mathcal{F}}(\mathcal{W}(\mathbf{p})) \text{ where } \mathbf{t}_{\mathcal{F}} = \mathcal{F}(\mathbf{t}) \tag{4.41}$$

The composition order of these two cases is shown in Fig. 4.2. In the following subsections we present the incorporation of these two functions compositions in the IC algorithm and explain why the second one is preferable. For simplicity, we use the LK-IC algorithm (Sec. 4.3.1) for face alignment that does not include appearance variation.



(a) Features from warped image.



(b) Warping on features image.

Figure 4.2: The two possible composition directions of the feature extraction function $\mathcal{F}$ and the warp function $\mathcal{W}(\mathbf{p})$.

4.4.1 Warp Function Computational Complexity

As shown in Sec. 4.2.7, the computational cost of the feature extraction function $\mathcal{F}(\mathbf{t})$ is $\mathcal{O}(L_T D^3)$, where $L_T = HW$ is the resolution of the image $\mathbf{t}$. Regarding the warp function, we need to consider that the warping of a D-channel image, $\mathbf{t}(\mathcal{W}(\mathbf{p}))$, includes the three following steps:

1. Synthesis of the shape model instance $\mathbf{s}$, generated as in Eq. 3.4 using the weights $\mathbf{p}$, which has a cost of $\mathcal{O}(2n_s n)$.

2. Computation of the mapping of each pixel in the mean shape $\bar{\mathbf{s}}$ to the synthesized shape instance. This firstly involves the triangulation of the shape instance in N_{tr} number of triangles (same as the number of triangles of the mean shape) using Delaunay triangulation [100]. Then, six affine transformation parameters are computed for each triangle based on the coordinates of the corresponding triangles' vertexes. Finally, the transformed location of each point within each triangle is evaluated. Thus, the complexity of this step is $\mathcal{O}(6N_{tr}\frac{m}{N_{tr}}) = \mathcal{O}(6m)$.

3. Copying the values of all channels D for all pixels from the input image to the reference frame $\bar{\mathbf{s}}$ ($\mathcal{O}(Dm)$).

Consequently, taking into account that $(6+D)m \gg 2n_s n$, the overall computational complexity of warping a multi-channel image is $\mathcal{O}((6+D)m)$.

4.4.2 Optimization with Features from Warped Image

From Eqs. 4.12 and 4.40 we get the cost function of minimizing

$$\underset{\Delta\mathbf{p}}{\operatorname{argmin}} \left\| \mathcal{F}(\mathbf{t}(\mathcal{W}(\mathbf{p}))) - \mathcal{F}(\bar{\mathbf{a}}(\mathcal{W}(\Delta\mathbf{p}))) \right\|^2 \tag{4.42}$$

with respect to $\Delta\mathbf{p}$. Thus, the first-order Taylor expansion of this expression around $\Delta\mathbf{p} = \mathbf{0}$ is

$$\mathcal{F}(\bar{\mathbf{a}}(\mathcal{W}(\Delta\mathbf{p}))) \approx \mathcal{F}(\bar{\mathbf{a}}) + \frac{\partial \mathcal{F}}{\partial \bar{\mathbf{a}}} \nabla\bar{\mathbf{a}} \left.\frac{\partial \mathcal{W}}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{0}} \Delta\mathbf{p} \tag{4.43}$$

Since it is not possible to compute $\frac{\partial \mathcal{F}}{\partial \bar{\mathbf{a}}}$, we make the approximation $\frac{\partial \mathcal{F}}{\partial \bar{\mathbf{a}}} \nabla\bar{\mathbf{a}} \approx \nabla\mathcal{F}(\bar{\mathbf{a}})$ and the linearization becomes

$$\mathcal{F}(\bar{\mathbf{a}}(\mathcal{W}(\Delta\mathbf{p}))) \approx \mathcal{F}(\bar{\mathbf{a}}) + \nabla\mathcal{F}(\bar{\mathbf{a}}) \left.\frac{\partial \mathcal{W}}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{0}} \Delta\mathbf{p} \tag{4.44}$$

Consequently, in every IC repetition step, the warping is performed on the intensities image ($D = 1$) with the current parameters estimate ($\mathcal{O}(7m)$) and is followed by the feature extraction ($\mathcal{O}(mD^3)$), ending up to a cost of $\mathcal{O}(m(7 + D^3))$ per iteration. Hence, by applying k iterations of the algorithm and given that $D^3 \gg 7$, the overall complexity of warping and features extraction is

$$\mathcal{O}(kmD^3) \tag{4.45}$$

Note that this is only a part of the final cost, as the IC algorithm complexity also needs to be taken into account. Moreover, in the AAMs case, it is difficult to extract window-based features (*e.g.*, HOG, SIFT, LBP) from the mean shape template image, as required from the above procedure. This is because, we have to pad the warped texture in order to compute features on the boundary, which requires extra triangulation points.

4.4.3 Optimization with Warping on Features Image

The combination of Eqs. 4.12 and 4.41 gives the cost function

$$\underset{\Delta\mathbf{p}}{\operatorname{argmin}} \|\mathbf{t}_{\mathcal{F}}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}}_{\mathcal{F}}(\mathcal{W}(\Delta\mathbf{p}))\|^2 \tag{4.46}$$

where $\mathbf{t}_{\mathcal{F}} = \mathcal{F}(\mathbf{t})$ and $\bar{\mathbf{a}}_{\mathcal{F}} = \mathcal{F}(\bar{\mathbf{a}})$ are the multi-channel feature-based representations of the input and the template images respectively. The linearization around $\Delta\mathbf{p} = \mathbf{0}$ has the same form as in Eq. 4.44 of the previous case. However, in contrast with the previous case, the warping is performed on the feature-based image. This means that the feature extraction is performed *once* on the input image and the resulting multi-channel image is warped during each iteration. Hence, the computational complexity of feature extraction and warping is $\mathcal{O}((6+D)m)$ per iteration and $\mathcal{O}(k(6+D)m+L_T D^3)$ overall per image for k iterations, where L_T is the resolution of the input image.

The above cost greatly depends on the input image dimensions L_T. In order to override this dependency, we firstly resize the input image with respect to the scaling factor between the face detection bounding box and the mean shape resolution. Then, we crop the resized image in a region slightly bigger than the bounding box. Thus, the resulting input image has resolution approximately equal to the mean shape resolution m, which leads to an overall complexity of

$$\mathcal{O}(km(6 + D) + mD^3) \tag{4.47}$$

for k iterations. Another reason for resizing the input image is to have correspondence on the scales on which the features are extracted, so that they describe the same neighborhood.

The computational complexities of Eqs. 4.45 and 4.47 are approximately equal for small number of channels D (e.g. for ES and IGO). However, this technique of warping the features image has much smaller complexity for large values of D (*e.g.*, HOG, SIFT, LBP, Gabor). This is because $k(D+6) < D^3$ for large values of D, so $km(6+D)$ can be eliminated in Eq. 4.47. Consequently, since $kmD^3 \gg mD$, it is more advantageous to compute the features image once and then warp the multi-channel image at each iteration. In the experiments (Sec. 4.5), we report the timings that prove the above conclusion. Finally, we carried out an extensive experiment comparing the two methods for face alignment (LK) in Sec. 4.5.1 (Fig. 4.4). The results indicate that warping the multi-channel features image performs better, which is an additional reason to choose this composition direction apart from the computational complexity.

4.5 Experimental Results

Herein, we present extended experiments for both face alignment (LK, Sec. 4.5.1) and face fitting (holistic AAMs, Secs. 4.5.2 and 4.5.3) using the IC framework. We employ all the dense features described in Sec. 4.2 with the parameters of Tab. 4.1.

Note that commonly LK and AAMs fitting is performed using an image pyramid with progressively increasing the number of shape and appearance parameters as the image resolution increases [18, 117, 124, 157]. However, in the following experiments of this chapter, the image pyramid is not employed in order to facilitate and simplify the comparisons. Using multiple fitting scales would make it difficult to derive any conclusions about the various features and approaches, such as the representation power, number of appearance and shape eigenvectors, convergence rate, etc. Nevertheless, a multi-level pyramid fitting framework is employed in the rest of this thesis, as also explained in individual Chapters 5, 6 and 7.

4.5.1 Face Alignment (Lucas-Kanade)

In this section, we conduct experiments for the task of face alignment using the LK-IC algorithm. In Sec. 4.5.1 we show a motivating experiment in which we compare the performance of IC with warping the features image at each iteration vs. extracting features from the warped image. In Sec. 4.5.1, we compare the performance of IC with warping the features image for all features types. For both experiments, we use the Yale Face Database B [65], which consists of 10 subjects with 576 images per subject under different viewing conditions. We select 1 template image and 10 testing images for each subject (100 image pairs) that are corrupted with extreme illumination conditions (Fig. 4.3).

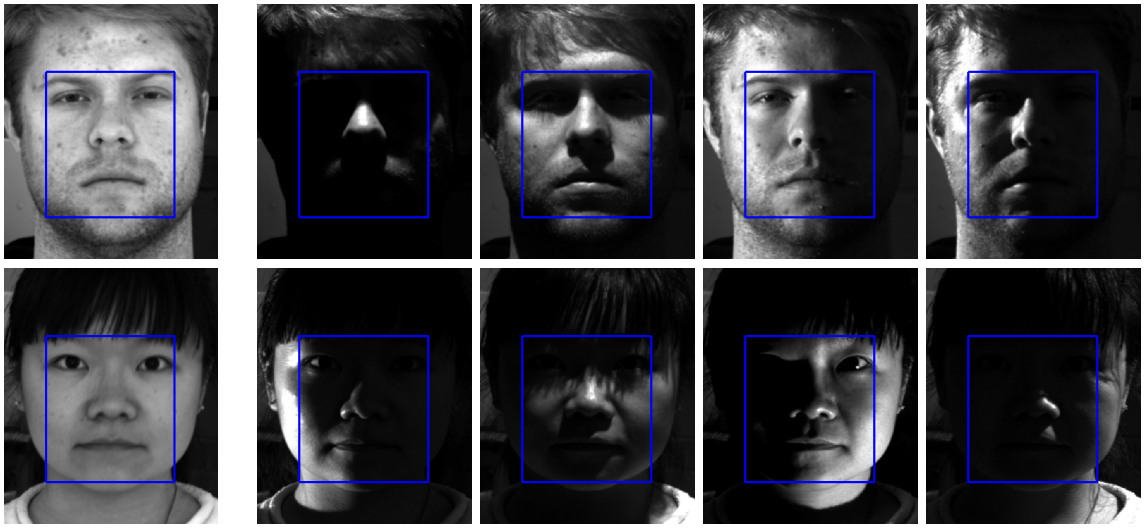


Figure 4.3: Yale B Database images examples. The template image (left) is corrupted with extreme illumination in the testing images for each subject.

We use the evaluation framework proposed in [18]. Specifically, we define three canonical points within a region of interest for each image. These points are randomly perturbed using a Gaussian distribution with standard deviation $\sigma = \{1, 2, \ldots, 9\}$. Then, we create the affine distorted image based on the affine warp defined between the original and perturbed points. After applying 30 iterations of the IC optimization algorithm, we compute the RMS error between the estimated and the correct locations of the three canonical points. The optimization is considered to have converged if the final RMS error is less than 3 pixels. Additionally, for each value of σ, we perform 100 experiments with different randomly perturbed warps. We evaluate the performance by plotting the average frequency of convergence and the average mean RMS error of the converged cases with respect to each value of σ. The results are averaged over the 100 experiment repetitions with different random warps.

Warping of features image vs Features from warped image

In the experiment of Fig. 4.4 we compare the performance of the two possible combination techniques between the features extraction function and the warp function, as presented in Sec. 4.4. The figure shows only HOG, SIFT, IGO and LBP cases, though we get the same results with the rest of features types. The comparison indicates that the method of extracting the features from the original image outperforms the one of extracting the features from the warped image, especially for large values of σ. The reason behind this behavior is that the warping of an image provokes some distortion on the texture which partly destroys the local structure. This has negative consequences on the computation of all the employed features,

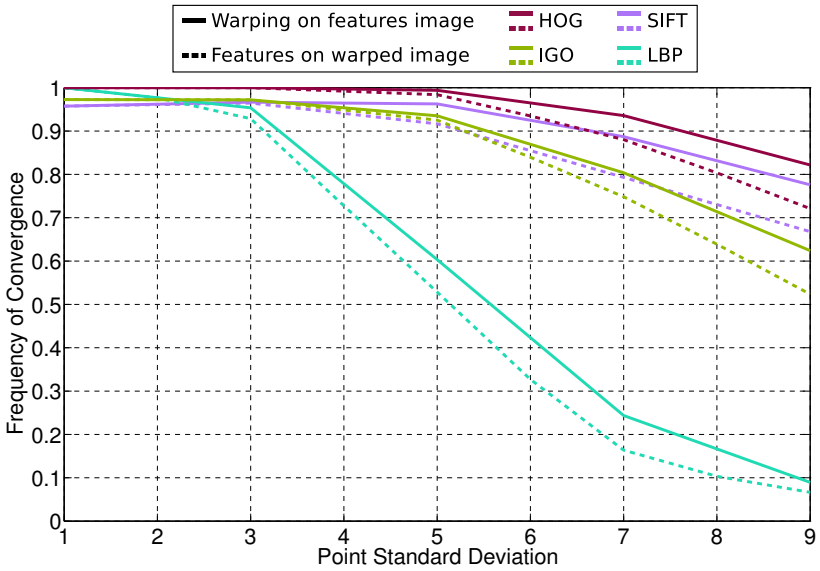


Figure 4.4: Comparison between the techniques of warping the features image and extracting features from the warped image. The plot shows results for HOG, SIFT, IGO and LBP features, however the rest of the features demonstrate the same behaviour.

because the descriptor of each pixel depends on the structure of its neighborhood.

Features Comparison

Figure 4.5 provides an evaluation of the robustness of each feature by showing the average frequency of convergence with respect to each value of σ. This experiment clearly indicates that Intensities or Gabor Magnitude features are totally inappropriate for such a task. HOG is the most robust feature with remarkable convergence frequency, followed by SIFT, IGO and ES. Finally, the LBPs family and Gabor Angles are not robust, but they can achieve decent results when the initialization is good.

4.5.2 Face Fitting (Active Appearance Models)

In this section we compare the performance of the selected features using AAMs for the task of face fitting with cross-database experiments. We investigate *which* features are more suitable for the task by comparing them with respect to their accuracy (Sec. 4.5.2), speed of convergence (Sec. 4.5.2) and computational cost (Sec. 4.5.2). We also shed light on *why* some features perform better by comparing them with respect to the number of appearance components (Sec. 4.5.2), the neighborhood size per pixel (Sec. 4.5.2) and the smoothness of their cost function (Sec. 4.5.2).

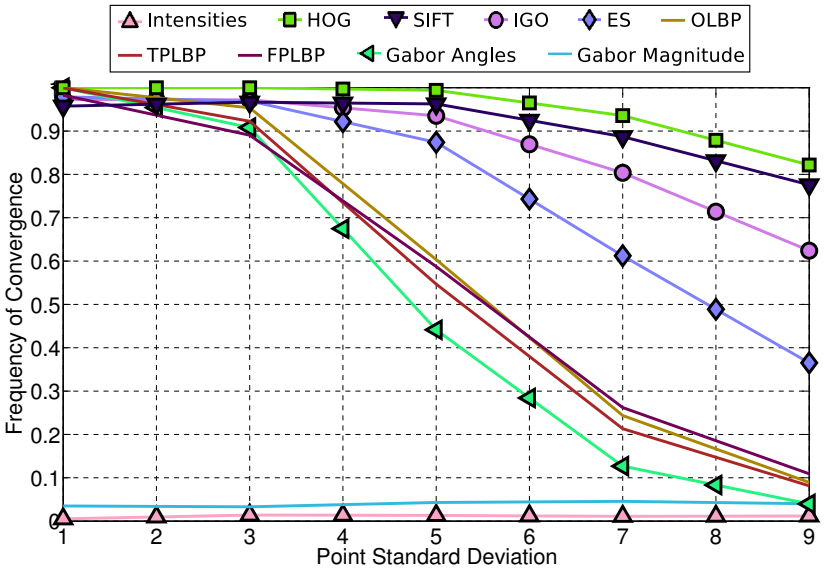


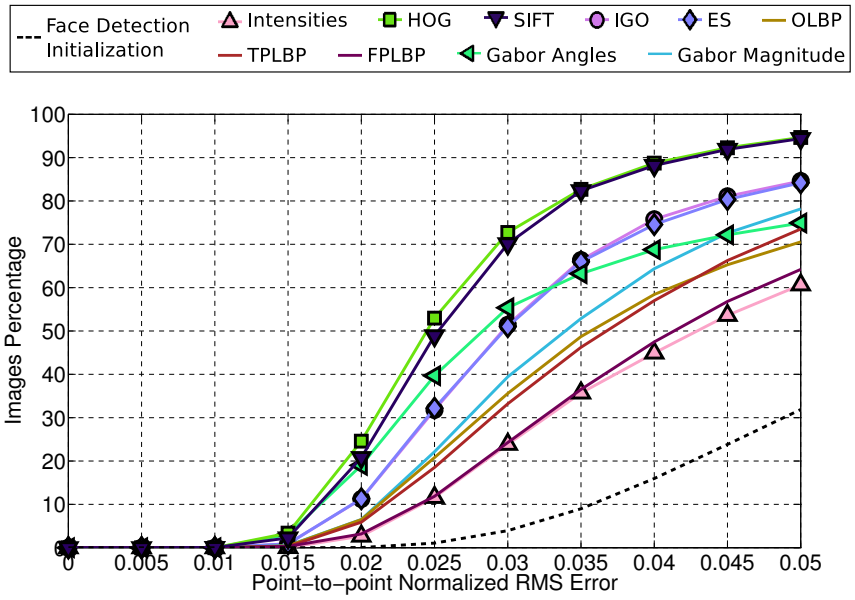
Figure 4.5: Face alignment (Lucas-Kanade) results on Yale B database using the inverse compositional framework. The figure shows the frequency of convergence with respect to the standard deviation σ.

As explained in Sec.4.3.2, AIC and SIC algorithms are theoretically equivalent and the only difference between them is that SIC is significantly slower. Specifically, the updates of SIC (Eq. 4.28) and AIC (Eqs. 4.34 and 4.37) are theoretically guaranteed to be the same [155]. Thus, herein we employ the AIC and POIC algorithms.

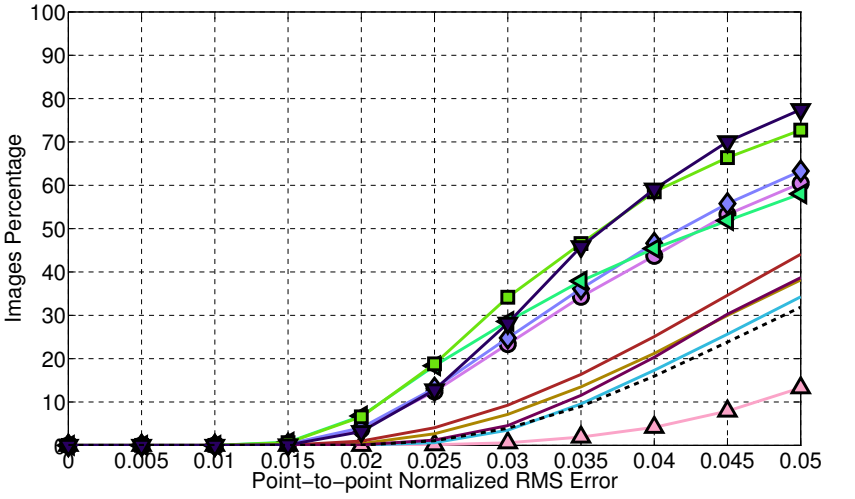
We use the in-the-wild databases presented in Sec. 3.3. Specifically, we use the 811 image of the LFPW trainset [22] for training. The testing is performed on AFW [185], LFPW testing set [22], Helen training and testing set [97] and iBUG [134], thus 3026 in-the-wild images in total. The fitting process is always initialized by computing the face's bounding box using Cascade Deformable Part Models (CDPM) face detector [123]. The fitting error is computed with the RMSE of Eq. 3.15 normalized with the face size of Eq. 3.16.

Accuracy

Figures 4.6a and 4.6b compare the accuracy of AIC and POIC respectively on all the databases (3026 testing images) for all the features types. The fitting procedure is performed using the methodology of Sec. 4.4.3 and keeping $n_s = 15$ eigenshapes and $n_a = 100$ eigentextures, regardless of the feature type. The results are plotted in the form of Cumulative Error Distributions (CED). Note that this experiment intends to make a fair comparison of the accuracy between the various features by letting the fitting procedure converge for all feature types. The results indicate that HOG and SIFT features are the most appropriate for the task. HOG



(a) Alternating IC



(b) Project-Out IC

Figure 4.6: Face fitting (AAMs) accuracy on in-the-wild databases (3026 test images) using the alternating and project-out inverse compositional frameworks, evaluated on 68 landmark points.

features perform better in the case of AIC and the SIFT ones are more robust for POIC, however the differences between them are very small. IGO and ES features have a sufficiently good performance. Moreover, similar to the face alignment case, Gabor Angles are not robust, but they achieve very accurate fitting result when they converge, especially in the POIC case. On the contrary, even though Gabor Magnitude features demonstrate a decent performance in the AIC, they completely diverge in the POIC case. This observation, combined with their

performance with the LK algorithm, indicates that they are unsuitable for image alignment without a linear appearance variation model. The same fact stands for intensities as well. Finally, the LBPs family has relatively poor performance. Figure 4.17 shows some indicative fitting examples from the very challenging iBUG database for all features with AIC.

Convergence

Herein, we examine the frequency of convergence achieved by each feature type. We assume that a fitting procedure has converged when either the cost function error incremental or the landmarks mean displacement are very small.

The cost incremental criterion is defined as

$$\frac{abs(error_{k-1} - error_k)}{error_{k-1}} < \epsilon \tag{4.48}$$

where $error_k$ is the cost function error from Eq. 4.17 at current iteration k and $\epsilon = 10^{-5}$. The mean displacement criterion is defined as the mean point-to-point normalized Euclidean distance between the shapes of current and previous iterations, thus

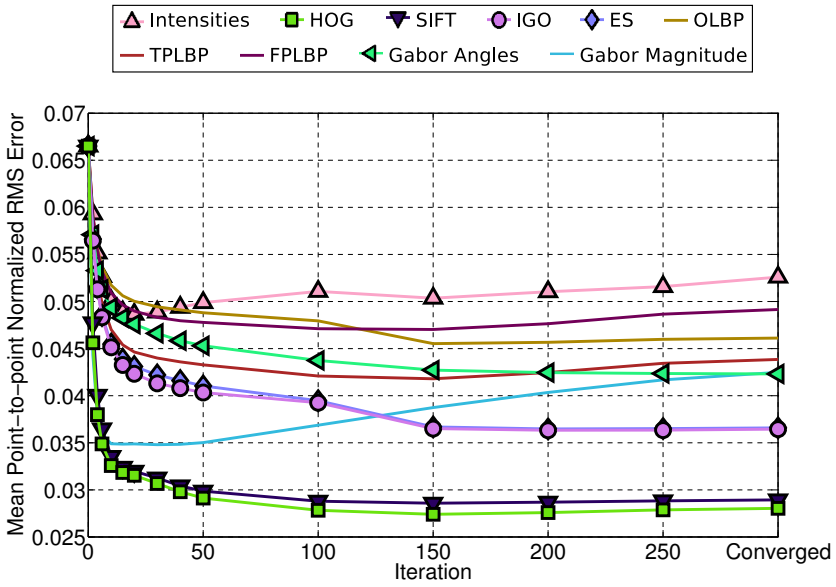
$$\frac{\sum_{i=1}^{n} \sqrt{(x_i^k - x_i^{k-1})^2 + (y_i^k - y_i^{k-1})^2}}{cn} < \epsilon \tag{4.49}$$

with $\epsilon = 10^{-4}$.

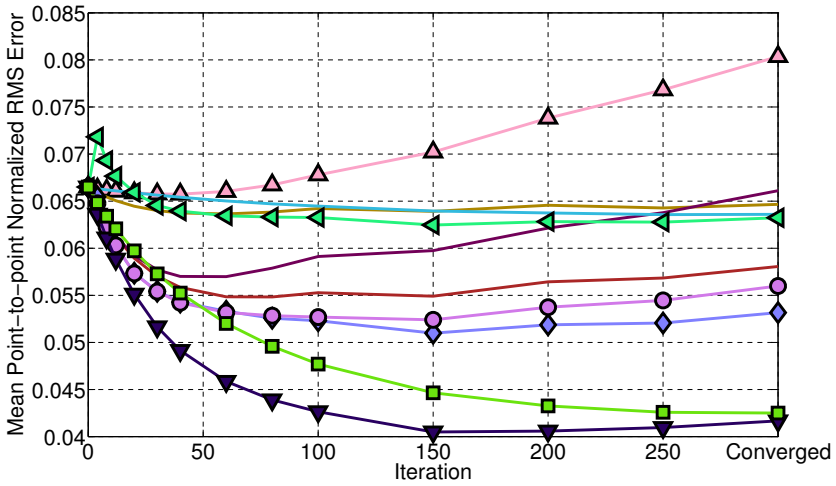
Figure 4.7 shows the mean point-to-point normalized RMS fitting error overall 3026 images with respect to the iteration number by allowing the optimization procedure to converge. The results indicate that HOG and SIFT features converge faster to a more accurate optimum compared to all the other feature types. Indicative examples of the convergence speed of each feature are shown in Fig. 4.8. Specifically, these plots show how fast the parameter value that corresponds to the 1$^{\text{st}}$ eigenvector of the shape subspace $\mathbf{U}_s$ moves towards its ideal (ground-truth) value. This eigenshape controls the face's pose over the yaw angle. These examples demonstrate the advantages of HOG and SIFT features, which reach the ideal value in very few iterations. Note that in all these experiments we want the algorithms to converge, thus we let them execute many iterations. However, this is not necessary in a practical application, because as the iterations advance, the improvements in the fitted shape get much smaller.

Timings

Table 4.2 reports the timings for each feature type using the two compositional scenarios explained in Sec. 4.4 within the AAMs optimization framework. It presents the computational



(a) Alternating IC



(b) Project-Out IC

Figure 4.7: Mean point-to-point normalized RMS fitting error with respect to iteration number on in-the-wild databases (3026 test images). The plot aims to compare the speed of convergence of each feature type. Please refer to Table 4.2 (columns 5-10) for the computational cost of each feature-based method.

cost per iteration and the total cost of running the optimization for 50 and 100 iterations. Note that the AAMs framework used for those experiments is developed without any code optimization. The reference frame (mean shape $\bar{\mathbf{s}}$) has size 170×170.

The table justifies the computational analysis presented in Sec. 4.4. As expected, it is faster to compute the features once and warp the features image (Eq. 4.47) rather than extracting

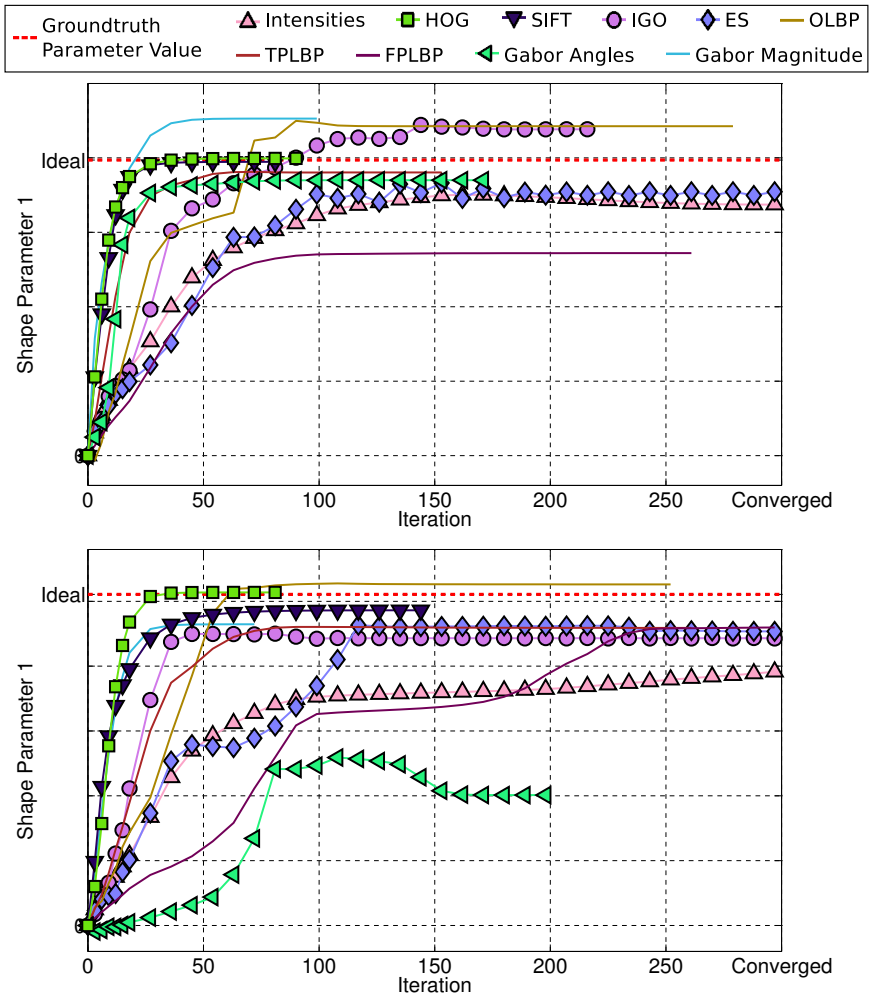


Figure 4.8: Indicative examples of the speed of convergence of each feature. The plots show how fast the 1st parameter value of the shape model moves towards its ideal (groundtruth) value. The example images are `image_0022.png` (*left*) and `image_0028.png` (*right*) from LFPW testing set.

features from each warped image at each iteration (Eq. 4.45). This is because, in most features cases, it is more expensive to extract features than warp a multi-channel image ($\mathcal{O}(\mathcal{F}) > \mathcal{O}(\mathcal{W})$). This happens with all the multi-channel features. The only exception is the SIFT features case, because the optimized implementation of [160] is faster than the unoptimized warping of the 36 channels ($\mathcal{O}(\mathcal{F}) < \mathcal{O}(\mathcal{W})$). Moreover, the combination of Tab. 4.2 with Fig. 4.7 suggests that even though high-dimensional features like HOG and SIFT converge really fast, their computational cost is quite similar to features with less channels that require multiple iterations until convergence.

The AAM fitting used in these experiments is implemented in Matlab using the Moore-Penrose pseudoinverse, which, despite the fact that it ensures robustness, it is computationally

				Warping on features image					
				Alternating IC			*Project-Out IC*		
Feature	*Channels*	*Feature function*	*Warp function*	*number of iterations*			*number of iterations*		
Type		*Cost ($\mathcal{F}$)*	*Cost ($\mathcal{W}$)*	*1*	*50*	*100*	*1*	*50*	*100*
Intensities	1	–	0.01	0.02	1.0	2.0	0.02	1.0	2.0
IGO, ES	2	0.01	0.01	0.05	2.0	4.0	0.04	1.5	3.0
OLBP	8	0.07	0.03	0.2	6.6	13.1	0.17	5.1	10.1
TPLBP	16	1.25	0.05	1.48	12.8	24.3	1.43	10.3	19.3
FPLBP		1.82		2.05	13.3	24.8	2.0	10.8	19.8
HOG	36	1.32	0.11	1.84	27.3	53.3	1.72	21.3	41.3
SIFT		0.07		0.59	26.1	52.1	0.47	20.1	40.1
Gabor		0.12		0.64	26.1	52.1	0.52	20.1	40.1
				Features from warped image					
				Alternating IC			*Project-Out IC*		
Feature	*Channels*	*Feature function*	*Warp function*	*number of iterations*			*number of iterations*		
Type		*Cost ($\mathcal{F}$)*	*Cost ($\mathcal{W}$)*	*1*	*50*	*100*	*1*	*50*	*100*
Intensities	1	–	0.01	0.02	1.0	2.0	0.02	1.0	2.0
IGO, ES	2	0.01	0.01	0.04	2.0	4.0	0.03	1.5	3.0
OLBP	8	0.07	0.03	0.18	9.0	18.0	0.15	7.5	15.0
TPLBP	16	1.25	0.05	1.44	72.0	144.0	1.39	69.5	139.0
FPLBP		1.82		2.01	100.5	201.0	1.96	98.0	196.0
HOG	36	1.32	0.11	1.74	87.7	174.0	1.62	81.0	162.0
SIFT		0.07		0.49	24.5	49.0	0.37	18.5	37.0
Gabor		0.12		0.54	27.0	54.0	0.42	21.0	42.0

Table 4.2: Computational costs of the feature extraction functions, the warp function and the AAM fitting using both composition ways of the two functions for all feature types. All the reported times are measured in seconds.

expensive. Additionally, as mentioned before, the fitting is not performed using an image pyramid. These two factors make the fitting procedure reported in Tab. 4.2 slower than expected. However, note that the aim of these experiments is to make a fair comparison of the computational complexity between the different feature types. It is not in the scope of this work to provide an optimized implementation of AAMs or features. Faster AAM optimization can be achieved with the framework proposed in [124, 155]. One could also use GPU or parallel programming to achieve faster performance and eliminate the cost difference between various

features and also between the two composition scenarios of $\mathcal{F}$ and $\mathcal{W}$. Finally, by applying a multi-scale fitting using an image pyramid greatly speeds up the fitting procedure, since convergence is achieved in less iterations, as shown in Chapter 5 (Sec. 5.3) and Chapter 7.

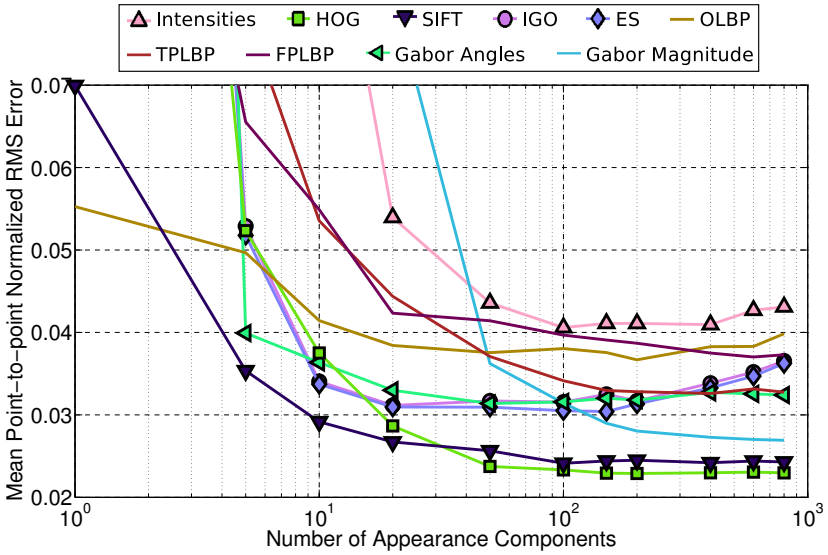


Figure 4.9: Mean point-to-point normalized RMS fitting error with respect to number of appearance components on the LFPW testset in-the-wild database. Note that we use logarithmic scale on the horizontal axis.

Number of Appearance Components

Figure 4.9 shows the mean point-to-point normalized RMS fitting error with respect to the number of appearance components, i.e. n_a, for LFPW testset using logarithmic scale on the horizontal axis. The results indicate that for most features, except IGO, ES and Intensities, the fitting performance is improved by increasing the number of appearance components. SIFT features can achieve very accurate results by using very few appearance components (even less than 10), thus with small computational cost. Additionally, note that Gabor Magnitude features can achieve significantly good accuracy (close to HOG and SIFT) if one keeps their whole eigenspectrum.

Neighborhood Size

Figure 4.10 plots the mean point-to-point normalized RMS fitting error with respect to the neighborhood size from which the feature value of each pixel is computed. For HOG and SIFT this is done by changing the cell size. In the case of the LBPs family, we alter the radius values (N_{radius}). For the rest of features (IGO, ES, Gabor, Intensities), we simply downscale the image. This experiment proves that the spatial neighborhood covered by each feature

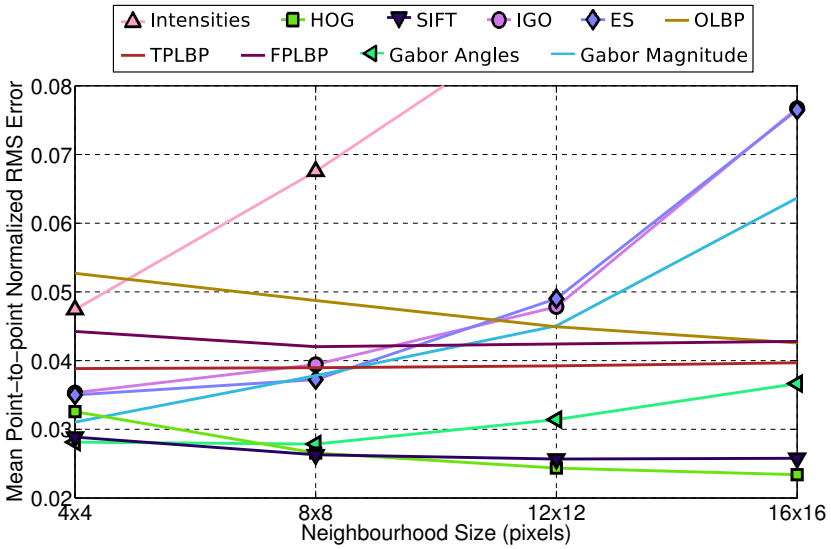
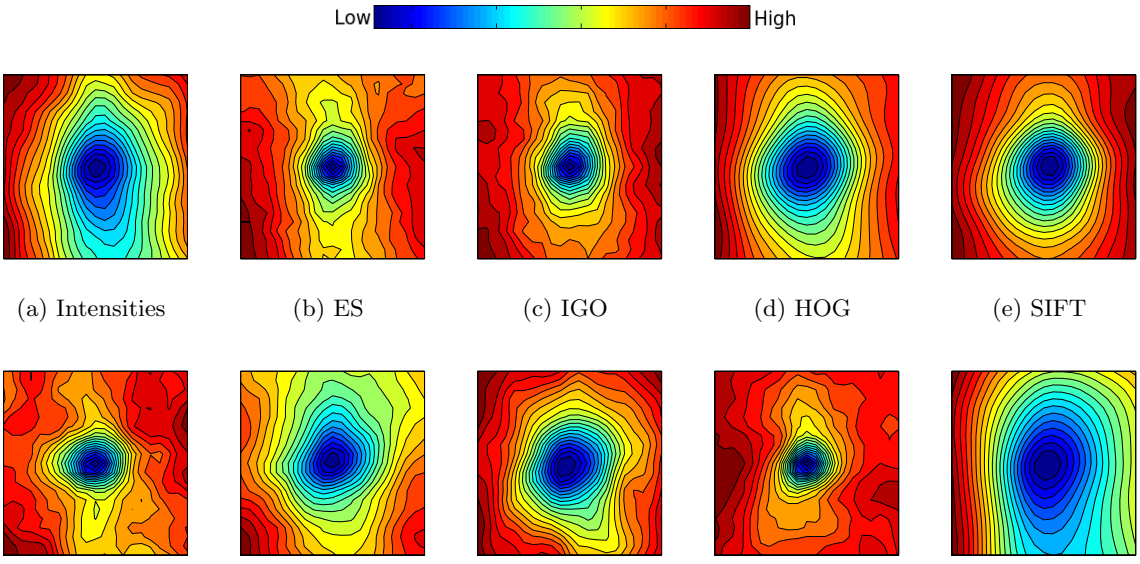


Figure 4.10: Mean point-to-point normalized RMS fitting error with respect to neighbourhood size on the LFPW testset in-the-wild database.

does not massively affect its performance. HOG, SIFT and LBP features are more accurate when applied to largest regions, as more information is accumulated to their channels. On the contrary, ES, IGO and Gabor features are not assisted by increasing the neighborhood size.



(a) Intensities (b) ES (c) IGO (d) HOG (e) SIFT

(f) OLBP (g) TPLBP (h) FPLBP (i) Gabor Angles (j) Gabor Magnitude

Figure 4.11: Contour plots of the cost function for each feature. The plots show the mean cost function over 100 images after translating the ground-truth shape over the x and y axis by $\pm 15\%$ (pixels) of the face size.

Cost Function

Figure 4.11 illustrates the cost function for each feature type in 2D contour plots. The plots are generated by translating the ground-truth shape of an image within a grid of ±15% (pixels) of the face size along the x and y axis and evaluating the cost of Eq. 4.17, where $\mathbf{c}$ are the projection parameters $\mathbf{c} = \mathbf{U}_a^\mathsf{T}(\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}})$. The plotted costs are averaged over 100 images. For each feature we use $n_a = 100$ appearance components, so that the experiment is fair and can be combined with the accuracy results of Sec. 4.5.2. These plots are very informative. The cost functions of IGO, ES and Gabor Angles have a very narrow region of small errors, which means that they can be accurate only when their initialization is close to the global optimum. On the contrary, Gabor Magnitude features have a very broad low error region, which means that they can quickly reach a small error but they will get stuck to a local minimum that is probably far from the global optimum. This can also be observed in Fig. 4.7a, where Gabor Magnitude features converge very fast to a low error but then start to diverge, due to the multiple local minima of their cost function. Finally, HOG and SIFT features have a smooth cost and the region of minimum values is large enough to facilitate fast and accurate convergence.

4.5.3 Comparison with state-of-the-art Face Fitting Methods

Herein we compare the performance of our proposed feature-based AAMs (both AIC and POIC) against two state-of-the-art facial trackers: Supervised Descent Method (SDM) [171] and Robust Discriminative Response Map Fitting (DRMF) for Constrained Local Models (CLMs) [13]. For our feature-based AAMs, we employ the HOG and SIFT features because they proved to be the most accurate and robust for both face alignment and fitting. We use the same initialization and experimental setup as in the previous section (Sec. 4.5.2). Specifically, the AAMs are trained on the 811 images of the LFPW trainset, keeping $n_s = 15$ eigenshapes and $n_a = 100$ eigentextures. For the other two methods, we used the implementations provided online by their authors with their pre-trained models. Note that both these methods are trained on thousands of images, much more than the 811 used to train our AAMs. All methods are initialized using the CDPM face detector [123]. In this experiment we report results evaluated on 49 landmark points shape mask instead of 68 points. This is because the SDM framework computes and returns only these 49 points. The 49-point mask occurs by removing the 17 points of the boundary (jaw) and the 2 points the mouth's corners from the 68 points shape mask of [69]. Thus this evaluation scheme emphasizes on the internal facial areas (eyebrows, eyes, nose, mouth).

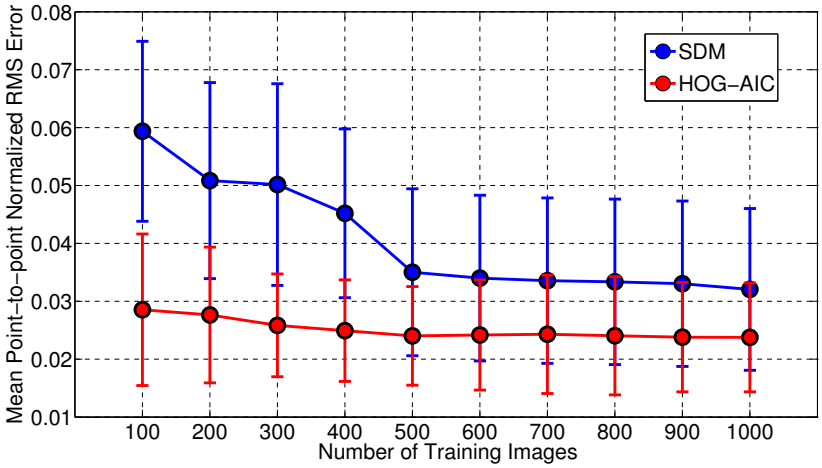


Figure 4.12: Performance (mean and standard deviation) of SIFT-AIC and SDM with respect to the number of training images. The performance is evaluated on Helen testset and is measured with the mean and standard deviation of the normalized RMS error. In this experiment we use our SDM implementation [1].

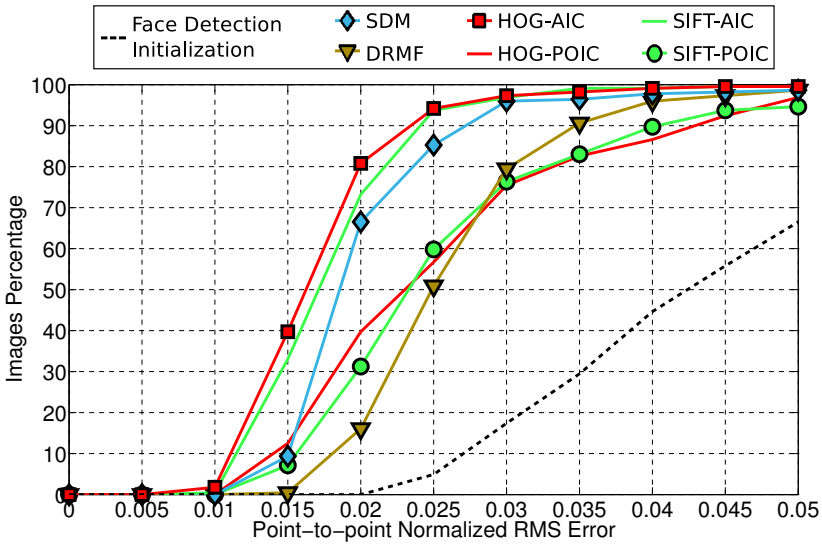


Figure 4.13: Comparison between our proposed HOG and SIFT AAMs and two state-of-the-art methods (SDM [171] and DRMF [13]) on LFPW testset. The evaluation is based on 49 points mask, which means it does not include the face boundary (jaw). For SDM and DRMF we use the code provided by their authors.

Figures 4.13-4.16 show the results on LFPW testset, AFW, iBUG and Helen train and test databases, respectively (3026 images in total). A main difference between these two methods and AAMs is that due to their discriminative nature, they both require many data in order to generalize well, whilst the generative shape and appearance models of AAMs perform well with much fewer training images. This is shown in Fig. 4.12 which plots the performance of

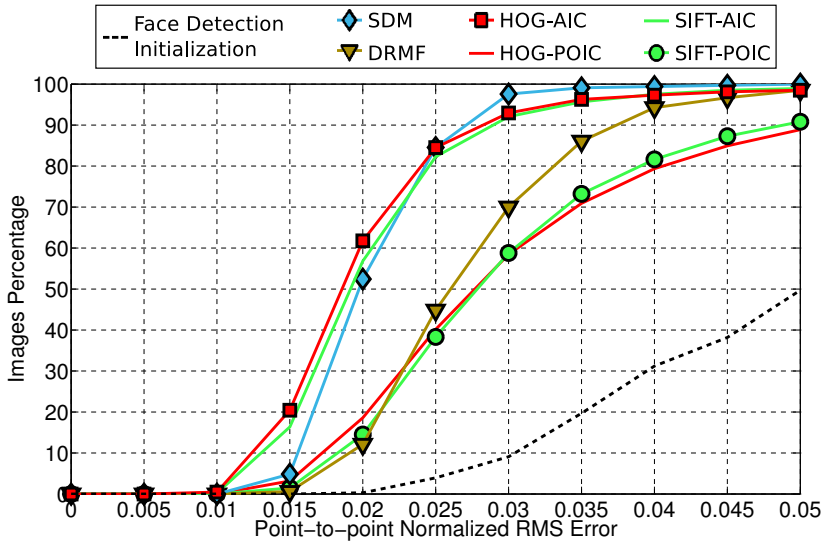


Figure 4.14: Comparison between our proposed HOG and SIFT AAMs and two state-of-the-art methods (SDM [171] and DRMF [13]) on Helen trainset and testset. The evaluation is based on 49 points mask, which means it does not include the face boundary (jaw). For SDM and DRMF we use the code provided by their authors.

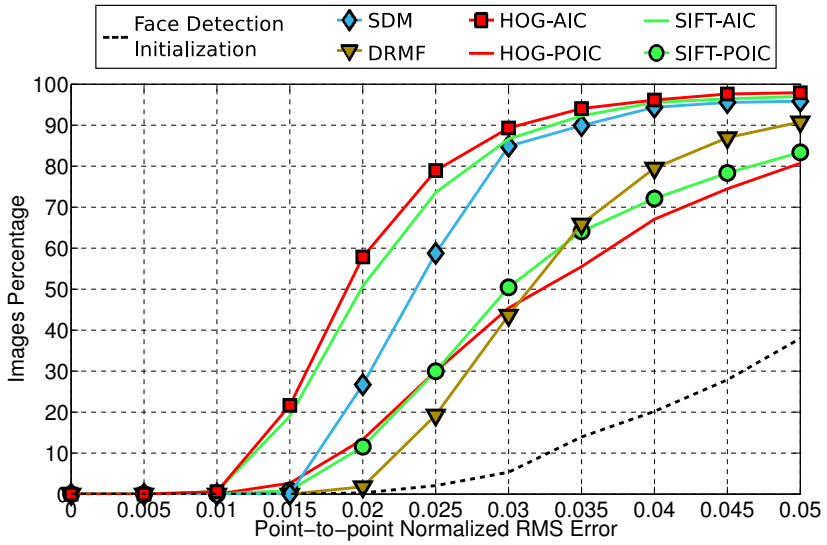


Figure 4.15: Comparison between our proposed HOG and SIFT AAMs and two state-of-the-art methods (SDM [171] and DRMF [13]) on AFW. The evaluation is based on 49 points mask, which means it does not include the face boundary (jaw). For SDM and DRMF we use the code provided by their authors.

HOG-AIC and SDM with respect to the number of training images. Since SDMs's authors do not provide any training code [171], for this small experiment we employ our SDM version developed in the Menpo Project [1]. The training images are randomly selected from the 2811 images of LFPW and Helen trainsets and the evaluation is applied on Helen testing set. The

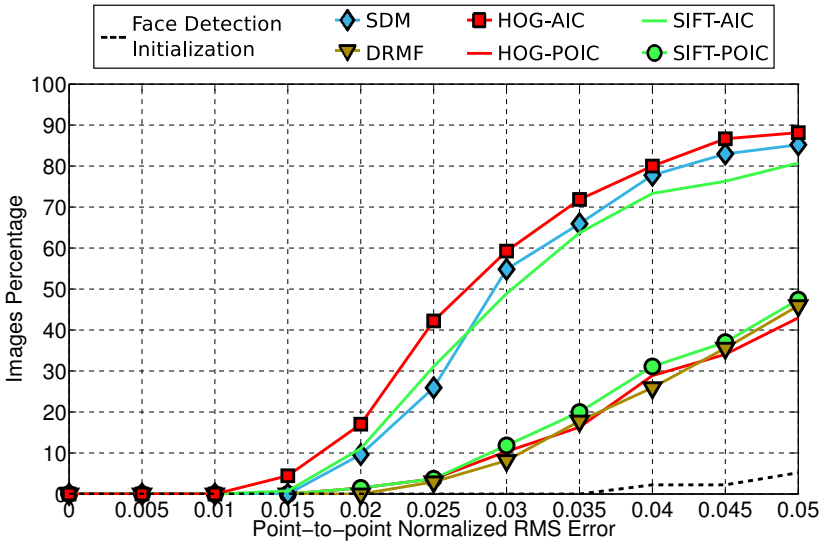


Figure 4.16: Comparison between our proposed HOG and SIFT AAMs and two state-of-the-art methods (SDM [171] and DRMF [13]) on iBUG. The evaluation is based on 49 points mask, which means it does not include the face boundary (jaw). For SDM and DRMF we use the code provided by their authors.

graph shows that SDM keeps improving as the number of training images increases whilst the SIFT AAMs performance remains almost the same. Finally, Fig. 4.17 shows some indicative fitting results using all the features employed in this work.

The results indicate that HOG-AIC and SIFT-AIC significantly outperform DRMF and are also more accurate than SDM. They are more accurate especially when they converge as can be seen from the percentage of images with error less or equal than 0.02. Even though SDM and DRMF have smaller computational complexities compared to Tab. 4.2, we find these results remarkable, considering that our feature-based AAMs are trained using much fewer training images. Finally, the results show that the HOG and SIFT POIC models have a similar performance as DRMF.

4.5.4 Results Interpretation and Discussion

In general, it is very difficult to find a strict theoretical difference between the various employed non-linear features, such as HOG, SIFT, LBP etc., because the design of features still remains mainly an empirical art rather than an exact science. Nevertheless, we can sketch the difference between the magnitude of Gabor filters in various scales and orientations and SIFT features. Gabor features have been used before in literature [112, 63], however our experiments prove that they are not efficient for generic face alignment and are probably more suitable for person-specific settings [167, 52].

The difference between the complex response (i.e., having both the magnitude and the phase) of Gabor filters and other employed features is that the former are produced by the convolution of a bank of linear filters, hence they are not robust to the facial appearance changes [112]. This is the reason why we prefer to extract non-linear features from the responses, i.e. the magnitude (modulus) and the phase. Moreover, the difference between the magnitude of Gabor filters in various scales and orientations and SIFT features can be explained using the theory on invariant scattering networks [34], according to which SIFT features can be very well approximated by the modulus of the coefficients of the wavelet transform using a particular family of wavelets (i.e. partial derivatives of a Gaussian) (for more details please refer to Section 2.3 of [34]). Convolution with Gabor filters with different scales and orientations does not constitute a proper wavelet image transform. In general Gabor filter expansion is not applied in building a wavelet transform, since this requires computation of bi-orthogonal wavelets, which may be very time-consuming. Therefore, usually a filter bank consisting of Gabor filters with various scales and rotations [167, 52], as we do in this work, is created and applied for feature extraction. In general, the results suggest that large-scale features are very robust and have a high convergence frequency even with initializations that are too far from ground-truth. However, when the initialization is close to the optimal solution, higher-frequency features tend to be more accurate. For example the phase filter information may have excellent localization properties when the deformation is small, but it is very sensitive to noise and small perturbations.

Finally, we believe that the advantages of the employed features, especially the multi-channel gradient based ones such as HOG and SIFT, are excellently coupled with the generalization ability of generative models. In fact, we believe that the most important experimental result shown in the previous section is that the combination of

1. non-linear least-squares optimization, with
2. robust features, and
3. generative models

can achieve very good performance without the need of large training datasets, which emphasizes the main advantage of the proposed framework over discriminative methods.

4.6 Conclusions

In this chapter, we presented a novel formulation of LK and holistic AAMs alignment algorithms which employs dense feature descriptors for the appearance representation. We showed, both theoretically and experimentally, that by extracting the features from the input image once and then warping the features image has better performance and lower computational complexity than computing features from the warped image at each iteration. This allows us to take advantage of the descriptive qualities of various features in order to achieve robust and accurate performance for the problems of face alignment and fitting. Our LK experiments prove that feature-based face alignment is invariant to person ID and extreme lighting variations. Our face fitting experiments on challenging in-the-wild databases show that the feature-based AAMs have the ability to generalize well to unseen faces and demonstrate invariance to expression, pose and lighting variations. The presented experiments also provide a comparison between various features and prove that HOG and SIFT are the most powerful. Finally, we report face fitting results using AAMs with HOG and SIFT features that outperform discriminative state-of-the-art methods trained on thousands of images. We believe that the experimental results are among the major contributions of this work, as they emphasize that the combination of highly-descriptive features with efficient optimization techniques leads to deformable models with remarkable performance.

(a) Face Detection Initialization

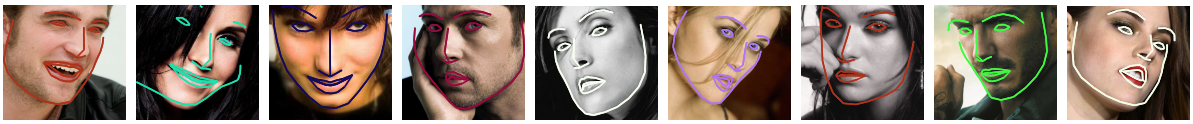
(b) HOG

(c) SIFT

(d) IGO

(e) ES

(f) Gabor Angles



(g) Gabor Magnitude

(h) OLBP

(i) TPLBP (similar for FPLBP)

(j) Intensities

Figure 4.17: Fitting examples using feature-based AIC on very challenging images from iBUG database.

Chapter 5

Active Pictorial Structures

Contents

5.1 Motivation

As explained in Chapter 1, one of the most well-studied deformable models are AAMs [39, 117]. In the previous chapter (Chapter 4), we showed that the combination of the Simultaneous [68] and Alternating [124, 153] inverse compositional algorithms with powerful features can achieve very accurate and robust performance. On the other hand, the Project-Out inverse compositional (POIC) [117] algorithm has a real-time complexity but is inaccurate, which makes it unsuitable for generic settings. Therefore, AAMs have two disadvantages:

1. They are slow and inappropriate for real-time applications.
2. By employing PCA the appearance of the object is modeled with a single multivariate normal distribution, which, as it will be shown in this chapter, restricts the fitting accuracy (Fig. 5.1).

Mainly due to the high complexity when using a holistic appearance representation, many existing methods employ a part-based one. This means that a local patch is extracted from the neighborhood around each landmark, as shown in Sec. 3.2.3. Among the most important

part-based deformable models are Pictorial Structures (PS) [61, 60, 7], their discriminative descendant Deformable Part Model (DPM) [58, 185] and their extensions like Deformable Structures [187]. PS learn a patch expert for each part and model the shape of the object using spring-like connections between parts based on a tree structure. Thus, a different distribution is assumed for each pair of parts connected with an edge, as opposed to the PCA shape model of AAMs that assumes a single multivariate normal distribution for all parts. The optimization aims to find a tree-based shape configuration for which the patch experts have a minimum cost and is performed using a dynamic programming algorithm based on the distance transform [59]. PS are successfully used for various tasks, such as human pose estimation [179] and face detection [185, 116]. Their biggest advantage is that they find the global optimum, thus they are not dependent neither require initialization. The dynamic programming technique computes all the responses for all the possible configurations of the parts and selects the one with the minimum cost. However, in practice, PS have two important disadvantages:

1. Inference is very slow.

2. Because the tree structure restricts too much the range of possible realizable shape configurations, the global optimum, even though it is the best solution in the span of the model, it does not always correspond to the shape that best describes the object in reality.

The method proposed in this chapter takes advantage of the strengths, and overcomes the disadvantages, of both AAMs and PS. We are motivated by the tree-based structure of PS and we further expand on this concept. Our model can formulate the relations between parts using any graph structure; not only trees. From AAMs we borrow the use of the Gauss-Newton algorithm in combination with a statistical shape model. Our weighted inverse compositional algorithm with fixed Jacobian and Hessian provides close to real-time cost with state-of-the-art performance. Thus, the proposed model shares characteristics from both AAMs and PS, hence the name Active Pictorial Structures (APS).

The idea of substituting the PCA shape model with a piece-wise linear model has also been proposed for 3D facial models in [147]. The most closely related method to the proposed APS is the Gauss-Newton Deformable Part Model (GN-DPM) [156]. It is a part-based AAM that takes advantage of the efficient inverse alternating Gauss-Newton technique proposed in [155] and reports very accurate performance. The two most important differences between the proposed APS and GN-DPM are that: *(i)* APS do not model the appearance of an object

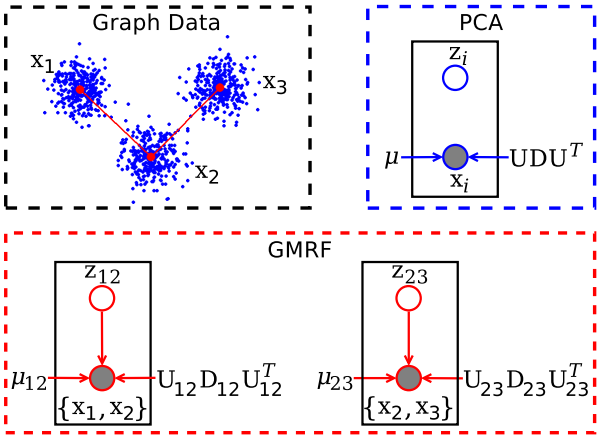


Figure 5.1: A simple visualization motivating the main idea behind APS. We propose to model the appearance of an object using multiple pairwise distributions based on the edges of a graph (GMRF) and show that this outperforms the commonly used PCA model under an inverse Gauss-Newton optimization framework.

using PCA but assume a different distribution for each pair of connected parts that proves to perform better, *(ii)* APS employ a weighted inverse compositional algorithm with fixed Jacobian and Hessian, which is by definition at least an order of magnitude faster than the alternating one.

In summary, the contributions of this work are:

- The proposed model combines the advantages of PS (graph-based relations between parts) and AAMs (weighted inverse Gauss-Newton optimization with statistical shape model).
- We show that it is more accurate to model the appearance of an object with multiple graph-based normal distributions, thus using a Gaussian Markov Random Field [131] structure, rather than a single multidimensional normal distribution (PCA), as is commonly done in literature. We also prove that this is not beneficial for modeling an object's shape, because the resulting covariance matrix has high rank and the shape subspace has too many dimensions to be optimized. We also show that employing a tree structure for the shape model, as done in PS [60, 58, 185], limits the model's descriptiveness and hampers the performance.
- We use the spring-like shape model of PS and DPM as a shape prior in the Gauss-Newton optimization. This deformation term makes the model more robust as it manages to restrict non-realistic instances of the object's shape.

- We propose, to the best of our knowledge, the best performing weighted inverse compositional Gauss-Newton algorithm with fixed Jacobian and Hessian. As it will be shown, its computational cost reduces to a single matrix multiplication per iteration and is independent of the employed graph structure. We test the proposed method on the task of face alignment, because of the plethora of annotated facial data. However, it can also be applied to other objects, such as eyes, cars etc. Our experiments show that APS outperform the current state-of-the-art methods.

The content of this chapter is based on the following publication:

- **E. Antonakos**, J. Alabort-i-Medina, and S. Zafeiriou. "Active Pictorial Structures", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, pp. 5435-5444, June 2015.

The rest of the chapter is structured as follows: Section 5.2 explains the training and fitting of the proposed method. Section 5.3 presents extended experimental results on the human face and other deformable objects (eyes, cars). Finally, Section 5.4 summarizes the outcomes of this chapter and draws conclusions.

5.2 Method

In the problem of object alignment in-the-wild, the sparse shape of the object is described using n landmark points that are usually located on semantic parts of the object, as explained by Eq. 3.2. The relative location of a landmark point i with respect to a landmark point j is defined as

$$\left.\begin{array}{l} \boldsymbol{\ell}_i = [x_i, y_i]^\mathsf{T} \\ \boldsymbol{\ell}_j = [x_j, y_j]^\mathsf{T} \end{array}\right\} \Rightarrow \begin{array}{l} dx_{ij} = x_i - x_j \\ dy_{ij} = y_i - y_j \\ d\boldsymbol{\ell}_{ij} = \boldsymbol{\ell}_i - \boldsymbol{\ell}_j = [dx_{ij}, dy_{ij}]^\mathsf{T} \end{array} \tag{5.1}$$

Furthermore, we employ the part-based appearance representation of Eq. 3.11. To facilitate notation, let us define a function $\mathcal{A} : \mathbb{R}^{2n} \longrightarrow \mathbb{R}^{mn}$ that extracts a feature-based image vector given a shape instance, as

$$\mathcal{A}(\mathbf{s}) = \left[\mathcal{F}(\boldsymbol{\ell}_1)^\mathsf{T}, \mathcal{F}(\boldsymbol{\ell}_2)^\mathsf{T}, \ldots, \mathcal{F}(\boldsymbol{\ell}_n)^\mathsf{T}\right]^\mathsf{T} \tag{5.2}$$

The function concatenates all the vectorized feature-based image patches that correspond to the n landmarks of the shape instance in a vector of length mn.

5.2.1 Graphical Model

Let us define an undirected graph between the n landmark points of an object as

$$G = (V, E) \tag{5.3}$$

where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of n vertexes and there is an edge $(v_i, v_j) \in E$ for each pair of connected landmark points. Moreover, let us assume that we have a set of random variables

$$X = \{X_i\}\,,\ \forall i : v_i \in V \tag{5.4}$$

which represent an abstract feature vector of length k extracted from each vertex v_i, *i.e.*, $\mathbf{x}_i, i : v_i \in V$ (e.g. the location coordinates, appearance vector etc.). We model the likelihood probability of two random variables that correspond to connected vertexes with a normal distribution

$$\begin{aligned} p(X_i = \mathbf{x}_i, X_j = \mathbf{x}_j | G) \sim \mathcal{N}(\boldsymbol{\mu}_{ij}, \boldsymbol{\Sigma}_{ij}), \\ \forall i, j : (v_i, v_j) \in E \end{aligned} \tag{5.5}$$

where $\boldsymbol{\mu}_{ij}$ is the $2k \times 1$ mean vector and $\boldsymbol{\Sigma}_{ij}$ is the $2k \times 2k$ covariance matrix. Consequently, the cost of observing a set of feature vectors $\{\mathbf{x}_i\}, \forall i : v_i \in V$ can be computed using a Mahalanobis distance per edge, *i.e.*

$$\sum_{\forall i,j:(v_i,v_j)\in E} \left(\begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix} - \boldsymbol{\mu}_{ij} \right)^{\mathsf{T}} \boldsymbol{\Sigma}_{ij}^{-1} \left(\begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix} - \boldsymbol{\mu}_{ij} \right) \tag{5.6}$$

In practice, the computational cost of computing Eq. 5.6 is too expensive because it requires looping over all the graph's edges. Especially in the case of a complete graph, it makes it impossible to perform inference in real time.

Inference can be much faster if we convert this cost to an equivalent matrical form as

$$(\mathbf{x} - \boldsymbol{\mu})^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \tag{5.7}$$

This is equivalent to modeling the set of random variables X with a Gaussian Markov Random Field (GMRF) [131]. A GMRF is described by an undirected graph, where the vertexes stand for random variables and the edges impose statistical constraints on these random variables. Thus, the GMRF models the set of random variables with a multivariate normal distribution

$$p(X = \mathbf{x} | G) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{5.8}$$

where

$$\boldsymbol{\mu} = \left[\boldsymbol{\mu}_1^{\mathsf{T}}, \ldots, \boldsymbol{\mu}_n^{T} \right]^{\mathsf{T}} = \left[E(X_1)^{\mathsf{T}}, \ldots, E(X_n)^{\mathsf{T}} \right]^{\mathsf{T}} \tag{5.9}$$

is the $nk \times 1$ mean vector and $\mathbf{\Sigma}$ is the $nk \times nk$ overall covariance matrix. We denote by $\mathbf{Q}$ the block-sparse precision matrix that is the inverse of the covariance matrix, *i.e.*,

$$\mathbf{Q} = \mathbf{\Sigma}^{-1} \tag{5.10}$$

By applying the GMRF we make the assumption that the random variables satisfy the three Markov properties (pairwise, local and global) and that the blocks of the precision matrix that correspond to disjoint vertexes are zero, *i.e.*,

$$\mathbf{Q}_{ij} = \mathbf{0}_{k \times k}, \ \forall i, j : (v_i, v_j) \notin E \tag{5.11}$$

By defining $\mathcal{G}_i = \{(i-1)k+1, (i-1)k+2, \ldots, ik\}$ to be a set of indices for sampling a matrix and by equalizing Eqs. 5.6 and 5.7 we can prove that the structure of the precision matrix is

$$\mathbf{Q} = \begin{cases} \sum\limits_{\forall j:(v_i,v_j)\in E} \mathbf{\Sigma}_{ij}^{-1}(\mathcal{G}_1, \mathcal{G}_1) + \\ \sum\limits_{\forall j:(v_j,v_i)\in E} \mathbf{\Sigma}_{ji}^{-1}(\mathcal{G}_2, \mathcal{G}_2), \ \forall v_i \in V, & \text{at } (\mathcal{G}_i, \mathcal{G}_i) \\ \mathbf{\Sigma}_{ij}^{-1}(\mathcal{G}_1, \mathcal{G}_2), \ \forall i, j : (v_i, v_j) \in E, & \text{at } (\mathcal{G}_i, \mathcal{G}_j) \\ & \text{and } (\mathcal{G}_j, \mathcal{G}_i) \\ 0, & \text{elsewhere} \end{cases} \tag{5.12}$$

Using the same assumptions and given a directed graph (cyclic or acyclic) $G = (V, E)$, where $(v_i, v_j) \in E$ denotes the relation of v_i being the parent of v_j, we can show that

$$\begin{aligned} &(\mathbf{x} - \boldsymbol{\mu})^{\mathsf{T}} \mathbf{Q} (\mathbf{x} - \boldsymbol{\mu}) = \\ &= \sum_{\forall i,j:(v_i,v_j)\in E} \left(\mathbf{x}_i - \mathbf{x}_j - \boldsymbol{\mu}_{ij}\right)^{\mathsf{T}} \mathbf{\Sigma}_{ij}^{-1} \left(\mathbf{x}_i - \mathbf{x}_j - \boldsymbol{\mu}_{ij}\right) \end{aligned} \tag{5.13}$$

is true if

$$\mathbf{Q} = \begin{cases} \sum\limits_{\forall j:(v_i,v_j)\in E} \mathbf{\Sigma}_{ij}^{-1} + \\ \sum\limits_{\forall j:(v_j,v_i)\in E} \mathbf{\Sigma}_{ji}^{-1}, \ \forall v_i \in V, & \text{at } (\mathcal{G}_i, \mathcal{G}_i) \\ -\mathbf{\Sigma}_{ij}^{-1}, \ \forall i, j : (v_i, v_j) \in E, & \text{at } (\mathcal{G}_i, \mathcal{G}_j) \\ & \text{and } (\mathcal{G}_j, \mathcal{G}_i) \\ 0, & \text{elsewhere} \end{cases} \tag{5.14}$$

where $\boldsymbol{\mu}_{ij} = E(X_i - X_j)$ and

$$\boldsymbol{\mu} = \left[\boldsymbol{\mu}_1^{\mathsf{T}}, \ldots, \boldsymbol{\mu}_n^{\mathsf{T}}\right]^{\mathsf{T}} = \left[E(X_1)^{\mathsf{T}}, \ldots, E(X_n)^{\mathsf{T}}\right]^{\mathsf{T}} \tag{5.15}$$

In this case, if G is a tree, then we have a Bayesian network. Please refer to Appendix A.1 for detailed proofs of Eqs. 5.12 and 5.14.

5.2.2 Model Training

APS differ from most existing generative object alignment methods because they assume a GMRF structure in order to model the appearance and the deformation of an object. As we show in the experiments, this assumption is the key that makes the proposed method efficient and accurate.

In order to train APS, assume that we have a set of N training images $\{\mathbf{I}^1, \dots, \mathbf{I}^N\}$ with the corresponding ground truth (manually annotated) shapes $\{\mathbf{s}^1, \dots, \mathbf{s}^N\}$.

Shape Model

APS use a statistical shape model built using PCA, similar to the PDM employed in most existing parametric methods such as AAMs, CLMs and GN-DPMs. As explained in Sec. 3.1, the procedure involves the alignment of the training shapes with respect to their rotation, translation and scaling (similarity transform) using Procrustes analysis, the subtraction of the mean shape and the application of PCA. We further augment the acquired subspace with four eigenvectors that control the global similarity transform of the object, re-orthonormalize [117] and keep the first n_s eigenvectors. Thus, we end up with a linear shape model $\{\bar{\mathbf{s}}, \mathbf{U} \in \mathbb{R}^{2n \times n_s}\}$, where $\bar{\mathbf{s}} = \left[E(\boldsymbol{\ell}_1)^\mathsf{T}, \dots, E(\boldsymbol{\ell}_n)^\mathsf{T}\right]^\mathsf{T}$ is the $2n \times 1$ mean shape vector and $\mathbf{U}$ denotes the orthonormal basis.

Let us define a function $\mathcal{S} \in \mathbb{R}^{2n}$ with slightly different signature than Eq. 3.4. Specifically, it generates a shape instance given the linear model's basis, an input shape and a parameters' vector (weights) as

$$\mathcal{S}(\mathbf{U}, \mathbf{s}, \mathbf{p}) = \mathbf{s} + \mathbf{U}\mathbf{p} \tag{5.16}$$

where $\mathbf{p} = [p_1, p_2, \dots, p_{n_s}]^\mathsf{T}$ are the parameters' values. Similarly, we define the set of functions $\mathcal{S}_i \in \mathbb{R}^2,\ \forall i = 1, \dots, n$ that return the coordinates of the i^{th} landmark of the shape instance as

$$\mathcal{S}_i(\mathbf{U}, \mathbf{s}, \mathbf{p}) = \mathbf{s}_{2i-1,2i} + \mathbf{U}_{2i-1,2i}\mathbf{p},\ \forall i = 1, \dots, n \tag{5.17}$$

where $\mathbf{s}_{2i-1,2i}$ denotes the coordinates' vector of the i^{th} landmark point, *i.e.*, $\boldsymbol{\ell}_i = [x_i, y_i]^\mathsf{T}$, and $\mathbf{U}_{2i-1,2i}$ denotes the $2i-1$ and $2i$ row vectors of the shape subspace $\mathbf{U}$. Note that from now onwards, for simplicity, we will write $\mathcal{S}(\mathbf{s}, \mathbf{p})$ and $\mathcal{S}_i(\mathbf{s}, \mathbf{p})$ instead of $\mathcal{S}(\mathbf{U}, \mathbf{s}, \mathbf{p})$ and $\mathcal{S}_i(\mathbf{U}, \mathbf{s}, \mathbf{p})$ respectively.

Another way to build the shape model is by using the GMRF structure (Fig. 5.1). Specifically, given an undirected graph $G^s = (V^s, E^s)$ and assuming that the pairwise locations' vector of two connected landmarks follows a normal distribution as in Eq. 5.5, *i.e.*,

$$[\boldsymbol{\ell}_i^\mathsf{T}, \boldsymbol{\ell}_j^\mathsf{T}]^\mathsf{T} \sim \mathcal{N}(\boldsymbol{\mu}_{ij}^s, \boldsymbol{\Sigma}_{ij}^s), \ \forall i, j : (v_i^s, v_j^s) \in E^s \tag{5.18}$$

we formulate a GMRF. Following Eq. 5.8 and using the shape vector of Eq. 3.2, this can be expressed as

$$p(\mathbf{s}|G^s) \sim \mathcal{N}(\bar{\mathbf{s}}, \boldsymbol{\Sigma}^s) \tag{5.19}$$

where the precision matrix $\mathbf{Q}^s$ is structured as shown in Eq. 5.12 with $\mathbf{x}_i = \boldsymbol{\ell}_i$ and $k = 2$. Then, after constructing the precision matrix, we can invert it and apply PCA on the resulting covariance matrix $\boldsymbol{\Sigma}^s = (\mathbf{Q}^s)^{-1}$ in order to obtain a linear shape model. Even though, as we show below, the GMRF-based modeling creates a more powerful appearance model representation, it does not do the same for the shape model. Our experiments suggest that the single Gaussian PCA shape model is more beneficial than any other model that assumes a GMRF structure. This can be explained by the fact that $\boldsymbol{\Sigma}^s$ ends up having a high rank, especially if G^s has many edges. As a result, most of its eigenvectors correspond to non-zero eigenvalues and they express a small percentage of the whole data variance. This means that during fitting we need to employ a large number of eigenvectors ($n_s \approx 2n$), much more than in the case of a single multivariate distribution, which makes the Gauss-Newton optimization very unstable and ineffective.

Appearance Model

In most AAM-like formulations, the appearance model is built by warping all textures to a reference frame, vectorizing and building the PCA model. In this work, we propose to model the appearance of an object using a GMRF graphical model, as presented in Sec. 5.2.1. In contrast to the shape model case, the GMRF-based appearance model is more powerful than its PCA counterpart. Specifically, given an undirected graph $G^a = (V^a, E^a)$ and assuming that the concatenation of the appearance vectors of two connected landmarks can be described by a normal distribution (Eq. 5.5), *i.e.*,

$$\left[\mathcal{F}(\boldsymbol{\ell}_i)^\mathsf{T}, \mathcal{F}(\boldsymbol{\ell}_j)^\mathsf{T}\right]^\mathsf{T} \sim \mathcal{N}(\boldsymbol{\mu}_{ij}^a, \boldsymbol{\Sigma}_{ij}^a), \ \forall i, j : (v_i^a, v_j^a) \in E^a \tag{5.20}$$

we form a GMRF that, using Eq. 5.2, can be expressed as

$$p(\mathcal{A}(\mathbf{s})|G^a) \sim \mathcal{N}(\bar{\mathbf{a}}, \boldsymbol{\Sigma}^a) \tag{5.21}$$

where $\bar{\mathbf{a}} = \left[E(\mathcal{F}(\boldsymbol{\ell}_1))^\mathsf{T}, \ldots, E(\mathcal{F}(\boldsymbol{\ell}_n))^\mathsf{T}\right]^\mathsf{T}$ is the $mn \times 1$ mean appearance vector and $\mathbf{Q}^a = (\boldsymbol{\Sigma}^a)^{-1}$ is the $mn \times mn$ precision matrix that is structured as shown in Eq. 5.12 with $\mathbf{x}_i = \mathcal{F}(\boldsymbol{\ell}_i)$ and $k = m$. During the training of the appearance model, we utilize the low rank representation of each edgewise covariance matrix $\boldsymbol{\Sigma}^a_{ij}$ by using the first n_a singular values of its SVD factorization. Given $\bar{\mathbf{a}}$ and $\mathbf{Q}^a$, the cost of an observed appearance vector $\mathcal{A}(\mathbf{s})$ corresponding to a shape instance $\mathbf{s} = \mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})$ in an image is

$$\begin{aligned} &\|\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})) - \bar{\mathbf{a}}\|^2_{\mathbf{Q}^a} = \\ &= \left[\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})) - \bar{\mathbf{a}}\right]^\mathsf{T} \mathbf{Q}^a \left[\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})) - \bar{\mathbf{a}}\right] \end{aligned} \tag{5.22}$$

Our experiments show that all the tested GMRF-based appearance models greatly outperform the PCA-based one.

Deformation Prior

Apart from the shape and appearance models, we also employ a deformation prior that is similar to the deformation models used in [60, 185]. Specifically, we define a directed (cyclic or acyclic) graph between the landmark points as $G^d = (V^d, E^d)$ and model the relative locations between the parent and child of each edge with the GMRF of Eq. 5.13. We assume that the relative location between the vertexes of each edge, as defined in Eq. 5.1, follows a normal distribution

$$\boldsymbol{\ell}_i - \boldsymbol{\ell}_j \sim \mathcal{N}(\boldsymbol{\mu}^d_{ij}, \boldsymbol{\Sigma}^d_{ij}), \ \forall (i,j) : (v^d_i, v^d_j) \in E^d \tag{5.23}$$

and model the overall structure with a GMRF that has a $2n \times 2n$ precision matrix $\boldsymbol{Q}^d$ given by Eq. 5.14 with $k = 2$. The mean relative locations vector used in this case is the same as the mean shape $\bar{\mathbf{s}}$, because $\boldsymbol{\mu}^d_{ij} = E(\boldsymbol{\ell}_i - \boldsymbol{\ell}_j) = E(\boldsymbol{\ell}_i) - E(\boldsymbol{\ell}_j)$. As mentioned in [60], the normal distribution of each edge's relative locations vector in some sense controls "the stiffness of a spring connecting the two parts". In practice, this spring-like model manages to constrain extreme shape configurations that could be evoked during fitting with very bad initialization, leading the optimization process towards a better result. Given $\bar{\mathbf{s}}$ and $\mathbf{Q}^d$, the cost of observing a shape instance $\mathbf{s} = \mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})$ is

$$\begin{aligned} &\|\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}) - \bar{\mathbf{s}}\|^2_{\mathbf{Q}^d} = \|\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}) - \mathcal{S}(\bar{\mathbf{s}}, \mathbf{0})\|^2_{\mathbf{Q}^d} = \\ &= \mathcal{S}(\mathbf{0}, \mathbf{p})^\mathsf{T} \mathbf{Q}^d \mathcal{S}(\mathbf{0}, \mathbf{p}) \end{aligned} \tag{5.24}$$

where we used the properties $\mathcal{S}(\bar{\mathbf{s}}, \mathbf{0}) = \bar{\mathbf{s}} + \mathbf{U0} = \bar{\mathbf{s}}$ and $\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}) - \bar{\mathbf{s}} = \bar{\mathbf{s}} + \mathbf{Up} - \bar{\mathbf{s}} = \mathcal{S}(\mathbf{0}, \mathbf{p})$.

5.2.3 Gauss-Newton Optimization

The trained shape, appearance and deformation models can be combined to localize the landmark points of an object in a new testing image $\mathbf{I}$. Specifically, given the appearance and deformation costs of Eqs. 5.22 and 5.24, the cost function to be optimized is

$$\underset{\mathbf{p}}{\operatorname{argmin}} \|\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})) - \bar{\mathbf{a}}\|^2_{\mathbf{Q}^a} + \|\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}) - \bar{\mathbf{s}}\|^2_{\mathbf{Q}^d} \tag{5.25}$$

We minimize the cost function with respect to the shape parameters $\mathbf{p}$ using a variant of the Gauss-Newton algorithm [71, 117, 18]. The optimization procedure can be applied in two different ways, depending on the coordinate system in which the shape parameters are updated: *(i) forward* and *(ii) inverse*. Additionally, the parameters update can be carried out in two manners: *(i) additive* and *(ii) compositional*, which we show that in the case of our model they are identical. However, the forward additive algorithm is very slow compared to the inverse one. This is the reason why herein we only present and experiment with the inverse case. Please refer to Appendix A.2 for a derivation of the forward case.

Inverse-Compositional

The compositional update has the form

$$\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}) \leftarrow \mathcal{S}(\mathbf{s}, \mathbf{p}) \circ \mathcal{S}(\bar{\mathbf{s}}, \Delta\mathbf{p})^{-1} \tag{5.26}$$

As also shown in [156], by expanding this expression we get

$$\mathcal{S}(\mathbf{s}, \mathbf{p}) \circ \mathcal{S}(\bar{\mathbf{s}}, \Delta\mathbf{p})^{-1} = \mathcal{S}(\mathcal{S}(\bar{\mathbf{s}}, -\Delta\mathbf{p}), \mathbf{p}) = \mathcal{S}(\bar{\mathbf{s}}, \mathbf{p} - \Delta\mathbf{p}) \tag{5.27}$$

Consequently, due to the translational nature of our motion model, the compositional parameters update is reduced to the parameters subtraction, as

$$\mathbf{p} \leftarrow \mathbf{p} - \Delta\mathbf{p} \tag{5.28}$$

which is equivalent to the additive update. By using this compositional update of the parameters and having an initial estimate of $\mathbf{p}$, the cost function of Eq. 5.25 is expressed as minimizing

$$\begin{aligned} \underset{\Delta\mathbf{p}}{\operatorname{argmin}} \|\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})) - \bar{\mathbf{a}}(\mathcal{S}(\bar{\mathbf{s}}, \Delta\mathbf{p}))\|^2_{\mathbf{Q}^a} + \\ + \|\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}) - \mathcal{S}(\bar{\mathbf{s}}, \Delta\mathbf{p})\|^2_{\mathbf{Q}^d} \end{aligned} \tag{5.29}$$

with respect to $\Delta\mathbf{p}$. With some abuse of notation due to $\bar{\mathbf{a}}$ being a vector, $\bar{\mathbf{a}}(\mathcal{S}(\bar{\mathbf{s}}, \Delta\mathbf{p}))$ can be described as

$$\bar{\mathbf{a}}(\mathcal{S}(\bar{\mathbf{s}}, \Delta\mathbf{p})) = \begin{bmatrix} \boldsymbol{\mu}^a_1(\mathcal{S}_1(\bar{\mathbf{s}}, \Delta\mathbf{p})) \\ \vdots \\ \boldsymbol{\mu}^a_n(\mathcal{S}_n(\bar{\mathbf{s}}, \Delta\mathbf{p})) \end{bmatrix} \tag{5.30}$$

where $\boldsymbol{\mu}_i^a = E(\mathcal{F}(\boldsymbol{\ell}_i)), \forall i = 1, \ldots, n$. This formulation gives the freedom to each landmark point of the mean shape to slightly move within its reference frame. The reference frame of each landmark is simply the $h \times w$ patch neighborhood around it, in which $\boldsymbol{\mu}_i^a$ is defined. In order to find the solution we need to linearize around $\Delta\mathbf{p} = \mathbf{0}$ as

$$\begin{cases} \bar{\mathbf{a}}(\mathcal{S}(\bar{\mathbf{s}}, \Delta\mathbf{p})) \approx \bar{\mathbf{a}} + \mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}} \Delta\mathbf{p} \\ \mathcal{S}(\bar{\mathbf{s}}, \Delta\mathbf{p}) \approx \bar{\mathbf{s}} + \mathbf{J}_{\mathcal{S}}|_{\mathbf{p}=\mathbf{0}} \Delta\mathbf{p} \end{cases} \tag{5.31}$$

where

$$\mathbf{J}_{\mathcal{S}}|_{\mathbf{p}=\mathbf{0}} = \mathbf{J}_{\mathcal{S}} = \frac{\partial \mathcal{S}}{\partial \mathbf{p}} = \mathbf{U} \tag{5.32}$$

is the $2n \times n_s$ shape Jacobian and $\mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}} = \mathbf{J}_{\bar{\mathbf{a}}}$ is the $mn \times n_s$ appearance Jacobian

$$\mathbf{J}_{\bar{\mathbf{a}}} = \nabla\bar{\mathbf{a}} \frac{\partial \mathcal{S}}{\partial \mathbf{p}} = \nabla\bar{\mathbf{a}}\mathbf{U} = \begin{bmatrix} \nabla\boldsymbol{\mu}_1^a \mathbf{U}_{1,2} \\ \vdots \\ \nabla\boldsymbol{\mu}_n^a \mathbf{U}_{2n-1,2n} \end{bmatrix} \tag{5.33}$$

where $\mathbf{U}_{2i-1,2i}$ denotes the $2i-1$ and $2i$ row vectors of the basis $\mathbf{U}$. Note that we make an abuse of notation by writing $\nabla\boldsymbol{\mu}_i^a$ because $\boldsymbol{\mu}_i^a$ is a vector. However, it represents the gradient of the mean patch-based appearance that corresponds to landmark i and it has size $m \times 2$. By substituting, taking the partial derivative with respect to $\Delta\mathbf{p}$, equating it to $\mathbf{0}$ and solving for $\Delta\mathbf{p}$ we get

$$\Delta\mathbf{p} = \mathbf{H}^{-1}[\mathbf{J}_{\bar{\mathbf{a}}}{}^{\mathsf{T}} \mathbf{Q}^a \left(\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})) - \bar{\mathbf{a}}\right) + \mathbf{H}_{\mathcal{S}}\mathbf{p}] \tag{5.34}$$

where

$$\left.\begin{array}{l} \mathbf{H}_{\bar{\mathbf{a}}} = \mathbf{J}_{\bar{\mathbf{a}}}{}^{\mathsf{T}} \mathbf{Q}^a \mathbf{J}_{\bar{\mathbf{a}}} \\ \mathbf{H}_{\mathcal{S}} = \mathbf{J}_{\mathcal{S}}{}^{\mathsf{T}} \mathbf{Q}^d \mathbf{J}_{\mathcal{S}} = \mathbf{U}^{\mathsf{T}} \mathbf{Q}^d \mathbf{U} \end{array}\right\} \Rightarrow \mathbf{H} = \mathbf{H}_{\bar{\mathbf{a}}} + \mathbf{H}_{\mathcal{S}} \tag{5.35}$$

is the combined $n_s \times n_s$ Hessian matrix and we use the property $\mathbf{J}_{\mathcal{S}}{}^{\mathsf{T}} \mathbf{Q}^d \left(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}) - \bar{\mathbf{s}}\right) = \mathbf{U}^{\mathsf{T}} \mathbf{Q}^d \mathbf{U}\mathbf{p} = \mathbf{H}_{\mathcal{S}}\mathbf{p}$. Note that $\mathbf{J}_{\bar{\mathbf{a}}}$, $\mathbf{H}_{\bar{\mathbf{a}}}$, $\mathbf{H}_{\mathcal{S}}$ and $\mathbf{H}^{-1}$ of Eq. 5.34 can be precomputed. The computational cost per iteration is only $\mathcal{O}(mnn_s)$. The cost is practically reduced to a multiplication between a $n_s \times mn$ matrix and a $n_s \times 1$ vector that leads to a close to real-time performance, similar to the one of the very fast SDM method [171].

Derivation of Existing Methods

The APS model shown in the cost function of Eq. 5.25 is an abstract formulation of a generative model from which many existing models from the literature can be derived.

PS [60], **DPM** [185] As explained in Sec. 5.1, the proposed model is partially motivated by PS [60, 185]. In the original formulation of PS, the cost function to be optimized has the

form

$$\begin{aligned}&\underset{\mathbf{s}}{\operatorname{argmin}} \sum_{i=1}^{n} m_i(\boldsymbol{\ell}_i) + \sum_{i,j:(v_i,v_j)\in E} d_{ij}(\boldsymbol{\ell}_i, \boldsymbol{\ell}_j) = \\ &= \underset{\mathbf{s}}{\operatorname{argmin}} \sum_{i=1}^{n} [\mathcal{A}(\boldsymbol{\ell}_i) - \boldsymbol{\mu}_i^a]^\mathsf{T} (\boldsymbol{\Sigma}_i^a)^{-1} [\mathcal{A}(\boldsymbol{\ell}_i) - \boldsymbol{\mu}_i^a] + \sum_{i,j:(v_i,v_j)\in E} [\boldsymbol{\ell}_i - \boldsymbol{\ell}_j - \boldsymbol{\mu}_{ij}^d]^\mathsf{T} (\boldsymbol{\Sigma}_{ij}^d)^{-1} [\boldsymbol{\ell}_i - \boldsymbol{\ell}_j - \boldsymbol{\mu}_{ij}^d]\end{aligned} \tag{5.36}$$

where $\mathbf{s} = [\boldsymbol{\ell}_1^\mathsf{T}, \ldots, \boldsymbol{\ell}_n^\mathsf{T}]^\mathsf{T}$ is the vector of landmark coordinates ($\boldsymbol{\ell}_i = [x_i, y_i]^\mathsf{T},\ \forall i = 1, \ldots, n$), $\mathcal{A}(\boldsymbol{\ell}_i)$ is a feature vector extracted from the image location $\boldsymbol{\ell}_i$ and we have assumed a tree $G = (V, E)$. $\{\boldsymbol{\mu}_i^a, \boldsymbol{\Sigma}_i^a\}$ and $\{\boldsymbol{\mu}_{ij}^d, \boldsymbol{\Sigma}_{ij}^d\}$ denote the mean and covariances of the appearance and deformation, respectively. In Eq. 5.36, $m_i(\boldsymbol{\ell}_i)$ is a function measuring the degree of mismatch when part v_i is placed at location $\boldsymbol{\ell}_i$ in the image. Moreover, $d_{ij}(\boldsymbol{\ell}_i, \boldsymbol{\ell}_j)$ denotes a function measuring the degree of deformation of the model when part v_i is placed at location $\boldsymbol{\ell}_i$ and part v_j is placed at location $\boldsymbol{\ell}_j$. The authors show an inference algorithm based on distance transform [59] that can find a global minimum of Eq. 5.36 without any initialization. However, this algorithm imposes two important restrictions:

1. The appearance of each part is independent of the rest of them.
2. G must always be acyclic (a tree).

Additionally, the computation of $m_i(\boldsymbol{\ell}_i)$ for all parts ($i = 1, \ldots, n$) and all possible image locations (response maps) has a high computational cost, which makes the algorithm very slow. Finally, in [185], the authors only use a diagonal covariance for the relative locations (deformation) of each edge of the graph, which restricts the flexibility of the model.

In the proposed APS, we aim to minimize the cost function of Eq. 5.25 which can be expanded as

$$\begin{aligned}&\underset{\mathbf{p}}{\operatorname{argmin}} \|\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})) - \bar{\mathbf{a}}\|^2_{\mathbf{Q}^a} + \|\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}) - \bar{\mathbf{s}}\|^2_{\mathbf{Q}^d} = \\ &= \underset{\mathbf{p}}{\operatorname{argmin}} [\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})) - \bar{\mathbf{a}}]^\mathsf{T} \mathbf{Q}^a [\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})) - \bar{\mathbf{a}}] + [\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}) - \bar{\mathbf{s}}]^\mathsf{T} \mathbf{Q}^d [\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}) - \bar{\mathbf{s}}]\end{aligned} \tag{5.37}$$

There are two main differences between APS and PS:

1. We employ a statistical shape model and optimize with respect to its parameters.
2. We use the efficient Gauss-Newton optimization technique.

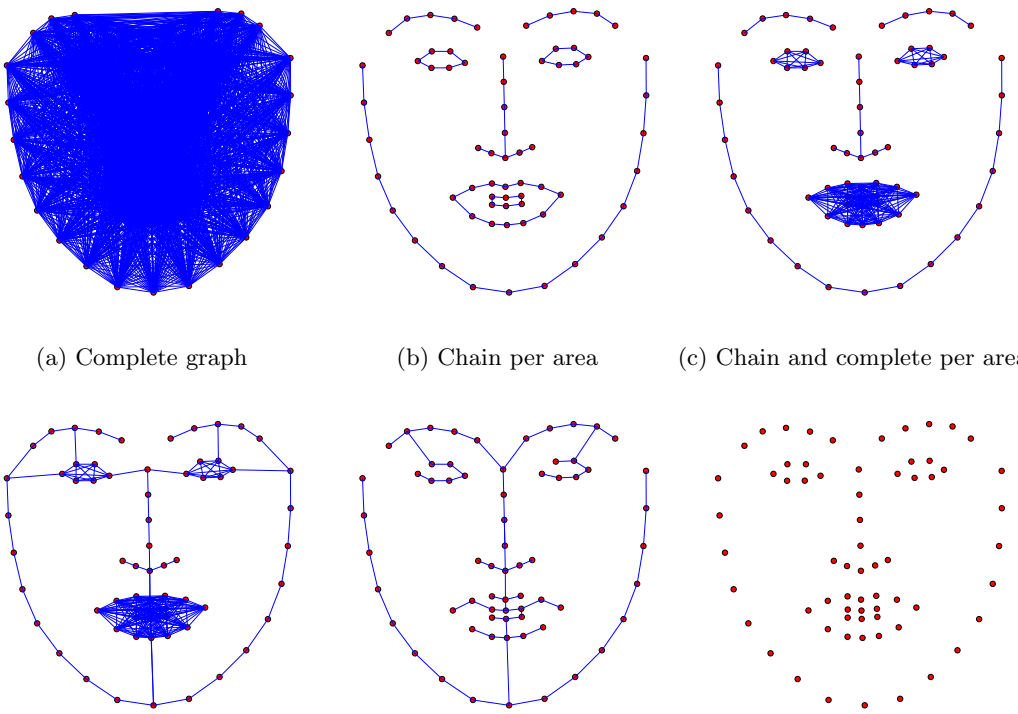
However, these differences introduce some important advantages. The proposed formulation allows to define a graph (not only tree) between the object's parts. This means that we can assume dependencies between any pair of landmarks for both the appearance and the deformation, as opposed to PS that assumes independence for the appearance and a tree structure for the deformation. As shown in the experimental results of Sec. 5.3.1, this lack of restriction is very beneficial. Finally, even though the efficient Gauss-Newton APS optimization does not find a global optimum, it handles the cost function in its matricial form (not in sums as in Eq. 5.36) and with an inverse-compositional manner, which ends up in much faster computational time that does not get affected by the graph structure.

AAM-POIC [117]. By removing the deformation prior from Eq. 5.25 and using a single multidimensional normal distribution in the shape and appearance models, the proposed APS are equivalent to AAMs. After performing an eigenanalysis on the appearance covariance matrix ($\mathbf{\Sigma}^a = \mathbf{W}\mathbf{D}\mathbf{W}^\mathsf{T}$), the POIC optimization of an AAM can be derived from the presented inverse algorithm by using as precision matrix the complement of the texture subspace, *i.e.*, $\mathbf{Q}^a = \mathbf{E} - \mathbf{W}\mathbf{W}^\mathsf{T}$. The part-based AAM of [156] uses an alternating optimization similar to [153]. Its project-out equivalent can be derived by using the above precision matrix.

BAAM-POIC [3]. Similar to the AAM-POIC, the Bayesian AAM can be formulated by replacing the precision matrix with $\mathbf{Q}^a = \mathbf{W}\mathbf{D}^{-1}\mathbf{W}^\mathsf{T} + \frac{1}{\sigma^2}(\mathbf{E} - \mathbf{W}\mathbf{W}^\mathsf{T})$. This precision matrix is derived by applying the Woodbury formula on the covariance matrix $\mathbf{W}\mathbf{D}\mathbf{W}^\mathsf{T} + \sigma^2\mathbf{E}$, where σ^2 is the variance of the noise in the appearance subspace $\mathbf{W}$. The above highlight the flexibility and strengths of the proposed model. As shown in Sec 5.3.2, the proposed GMRF-based appearance model makes our inverse technique, to the best of our knowledge, the best performing one among all inverse algorithms with fixed Jacobian and Hessian (*e.g.*, POIC).

5.3 Experimental Results

In this section we present a comprehensive evaluation of the different ways in which APS can be used to model the shape and appearance of an object and compare their performance against state-of-the-art Deformable Models. In all presented cases, the proposed APS are built using a two-level pyramid. We keep about 92% of the shape variance and set $n_a = 150$ for both levels that corresponds to about 80% of the appearance variance. The appearance is represented either by pixel intensities or dense SIFT [109] with 8 channels and the extracted patch size is 17×17. The accuracy of the fitting results is measured by the point-to-point RMS error between the fitted shape and the ground truth annotations, normalized by the face



(a) Complete graph (b) Chain per area (c) Chain and complete per area

(d) Chain and complete per area with connections between them (e) Minimum spanning tree (f) Empty graph

Figure 5.2: Employed GMRF graph structures.

size, as proposed in [185]. Note that our Python implementation of APS runs at $50ms$ per frame, which is very close to real-time. We believe that with further code optimization, APS are likely to be capable of running in real-time on high end desktop/laptop machine. Their time complexity is independent of the graph structure that is employed.

5.3.1 Internal Experimental Analysis

Herein, we present three experiments as a proof of concept regarding the formulation of APS. Specifically, we aim to examine the contribution of each one of the shape, appearance and deformation models and evaluate various graph structures. The model is trained using the 811 images of LFPW [22] train set and tested on the corresponding test set. We use the annotations provided by the 300W competition [133, 134, 132] and evaluate using 66 landmark points which are derived by removing landmarks 61 and 65 from the 68-points mark-up. In this set of experiments, we don't extract any appearance features and only use pixel intensities. Figure 5.2 shows the graph structures that we employ for the purpose of these experiments.

Note that the minimum spanning tree (MST) is computed as shown in [60]. The fitting process of the presented experiments is initialized by adding Gaussian noise to the global similarity transform retrieved from the ground truth annotations (without in-plane rotation) and applying it to the mean shape $\bar{\mathbf{s}}$. We set the standard deviation of the random noise to 0.04, which generates very challenging initializations.

Graph type G^a	*mean* $\pm$ *std*	*median*	≤ 0.04
Fig. 5.2a	0.0399 $\pm$ 0.0227	0.0324	68.3%
Fig. 5.2b	**0.0391 $\pm$ 0.0243**	**0.0298**	**69.6%**
Fig. 5.2c	0.0506 $\pm$ 0.0371	0.0370	58.9%
Fig. 5.2d	0.0492 $\pm$ 0.0373	0.0354	58.9%
Fig. 5.2e	0.0413 $\pm$ 0.0257	0.0316	65.2%
Fig. 5.2f	0.0398 $\pm$ 0.0246	0.0319	66.5%
PCA	0.0716 $\pm$ 0.0454	0.0595	25.5%
Initialization	0.0800 $\pm$ 0.0280	0.0768	4.0%

Table 5.1: Comparison of the GMRF-based and the PCA-based appearance model of APS.

Beginning with the appearance model, Tab. 5.1 reports the performance when using a GMRF with the graph structures of Fig. 5.2 and when using a single multivariate normal distribution through PCA. The performance is reported in the form of statistical measures (mean, median and standard deviation) and as the percentage of the testing images that achieved a final error ≤ 0.04 (value at which the result is considered adequately good by visual inspection). For this experiment, we use a PCA shape model and a deformation prior with the MST. The improvement is significantly high. Even the empty graph, which generates a block diagonal precision matrix $\mathbf{Q}^a$, thus it assumes independence between all parts, greatly outperforms the PCA case. The most appropriate graph structure is the one of Fig. 5.2b, which suggests that, for the case of faces, it is better to connect the landmarks of each facial area (eyes, mouth, nose etc.) between them and avoid relating the areas between each other.

Table 5.2 presents the same experiment for the shape model and the results are opposite to those of the appearance model. However, this is a well expected result. As mentioned in Sec. 5.2.2, the appearance model utilizes directly the constructed block sparse precision matrix. On the contrary, we need to decompose the covariance matrix ($\mathbf{\Sigma}^s = (\mathbf{Q}^s)^{-1}$) of the shape model in order to learn a parametric subspace that will be used during optimization. However, due to the block sparse formulation, the resulting covariance matrix has high (in some cases full) rank. Most eigenvalues are non-zero and they represent a small percentage

Graph type G^s	*mean* ± *std*	*median*	≤ 0.04
Fig. 5.2a	0.0495 ± 0.0273	0.0420	45.5%
Fig. 5.2b	0.0496 ± 0.0276	0.0438	45.5%
Fig. 5.2c	0.0503 ± 0.0262	0.0433	44.2%
Fig. 5.2d	0.0495 ± 0.0257	0.0434	44.6%
Fig. 5.2e	0.0519 ± 0.0306	0.0437	43.8%
Fig. 5.2f	0.0492 ± 0.0249	0.0437	42.9%
PCA	**0.0412 ± 0.0295**	**0.0301**	**65.6%**
Initialization	0.0800 ± 0.0280	0.0768	4.0%

Table 5.2: Comparison of the GMRF-based and the PCA-based shape model of APS.

of the data variance. Thus by keeping more than 90% of the total variance, the model ends up with too many modes of variation (about 100 in the case of 68 vertexes and depending on the graph structure). Consequently, it is very hard to apply a robust optimization in such a parametric space, as the search space is too large.

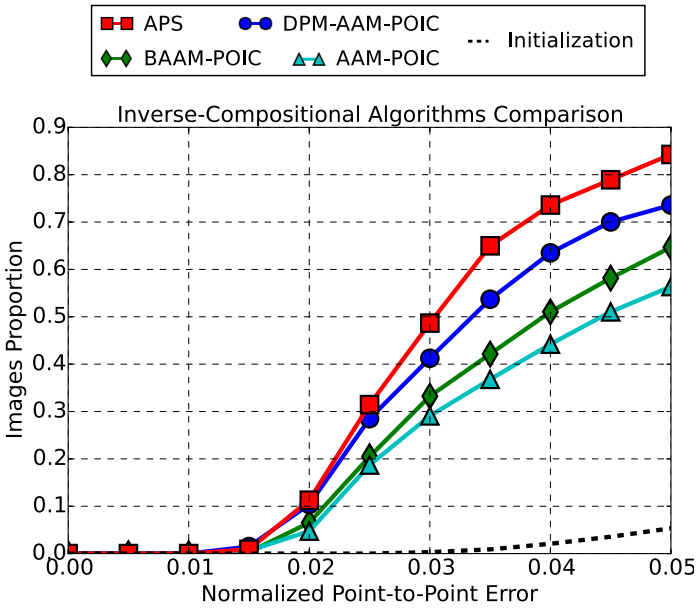
Deformation prior G^d	Shape model G^s	
	Fig. 5.2a	*PCA*
No prior	0.1327 ± 0.0857	0.0429 ± 0.0267
Fig. 5.2b	0.0524 ± 0.0256	0.0430 ± 0.0240
Fig. 5.2e	**0.0495 ± 0.0273**	**0.0391 ± 0.0243**

Table 5.3: Comparison of the GMRF-based and the PCA-based deformation prior of APS in combination with the GMRF-based and the PCA-based shape model.

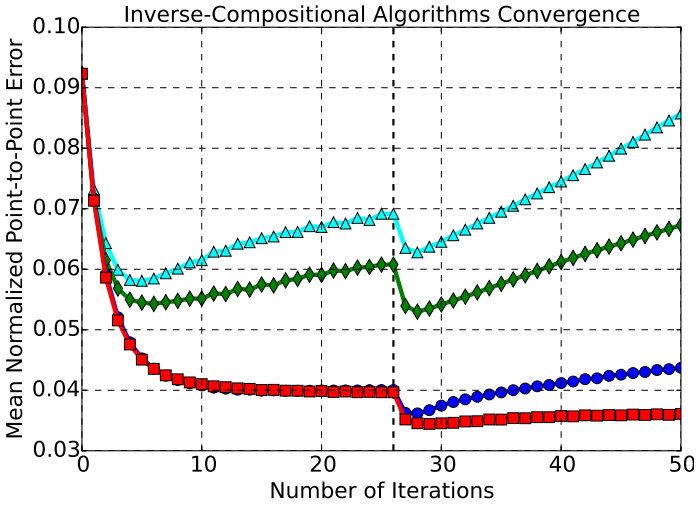
Finally, Tab. 5.3 examines the contribution of the deformation prior of Eq. 5.25. We use the graph of Fig. 5.2b for the appearance model and we test for two cases of the shape model: PCA and GMRF with a complete graph (Fig. 5.2a). The results prove that the prior plays an important role in both cases, as it improves the result. Especially in the case of the GMRF, the improvement is significant. Given the previous analysis about the non robust behavior of a GMRF shape model, this result is expected because the prior term will prevent the shape model from generating non-realistic instances of the face.

5.3.2 Comparison with State-of-the-Art Methods

Figures 5.3a and 5.3b aim to compare the accuracy and convergence speed of APS against the other existing inverse compositional techniques with fixed Jacobian and Hessian (POIC)



(a) Accuracy



(b) Convergence

Figure 5.3: Comparison of APS accuracy and convergence with other inverse compositional methods with fixed Jacobian and Hessian on AFW database. The dashed vertical black line in *(b)* denotes the transition from lower to higher pyramidal level.

mentioned in 5.2.3. AAM-POIC [117] and BAAM-POIC [3] denote the POIC optimization of an AAM and a Bayesian AAM. AAM-DPM-POIC refers to the inverse algorithm that can be combined with the AAM part-based model of [156]. All methods are trained on LFPW database in the same manner, using the same pyramid and extracting dense SIFT features with 8 channels. For all of them we keep $n_s = 5$ and $n_s = 15$ shape components for the low and high levels respectively, that correspond to about 92% of the total shape variance, and

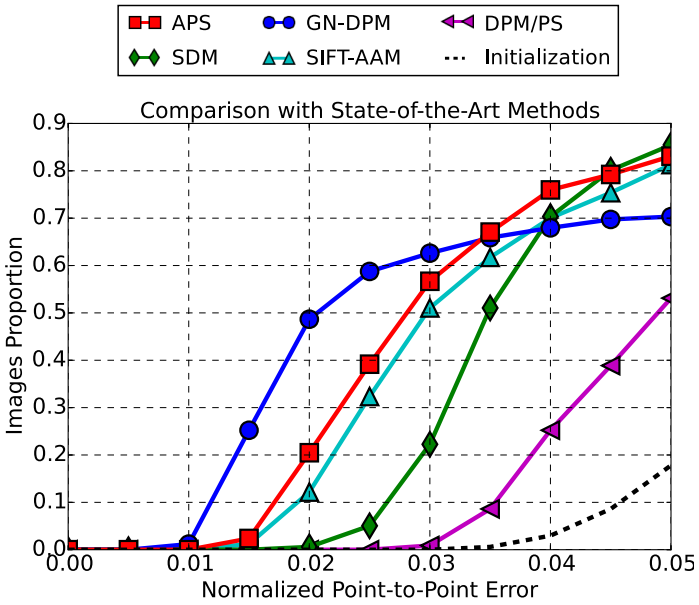


Figure 5.4: Comparison of APS with current state-of-the-art methods on AFW database.

APS	*SDM*	*SIFT-AAM*	*GN-DPM*	*DPM/PS*
0.0415	0.0453	0.0423	0.0686	0.0585

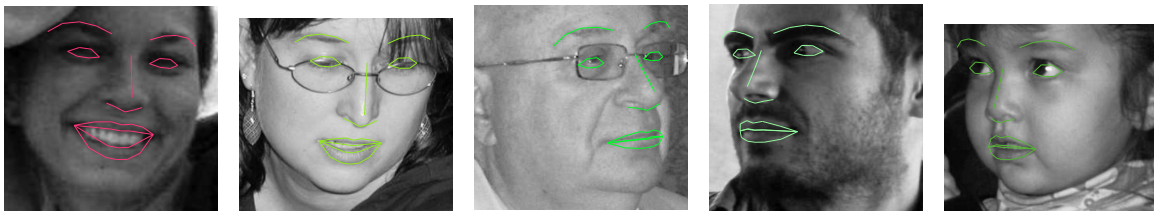
Table 5.4: Mean values of the cumulative error curves reported in Fig. 5.4.

$n_a = 150$ appearance components for both levels. The results, which are computed using 66 landmark points, are reported on the challenging AFW [185] database and indicate that the proposed method outperforms all existing inverse-compositional techniques by a significant margin. Most importantly, APS need very few number of iterations in order to converge (less than 10 at the first pyramidal level and no more than 4 at the second), which highlights their close to real-time computational complexity.

Figure 5.4 compares APS against the current state-of-the-art techniques: SDM [171], the recently proposed GN-DPM [156] and SIFT-AAM [8, 9]. The initialization for all methods is done using the bounding box of the landmark points returned by DPM [185] (the black dashed line). For all the methods we used the pre-trained implementations provided by their authors, except SIFT-AAM which we trained using the Menpo Project [1]. Note that all competing methods are trained on much more data than the 811 LFPW images that we use. The result is reported on the AFW database and computed based on 49 points, which is the mark-up that both SDM and GN-DPM return. Table 5.4 reports the mean values of the cumulative error curves of Fig. 5.4. These results show that APS outperform all methods and are more robust.

Note that GN-DPM is very accurate when the initialization is close to the ground-truth but is not robust against bad initializations, as indicated by its large mean error value. Finally, Fig. 5.5 shows some indicative fitting examples.

(a) Initialization



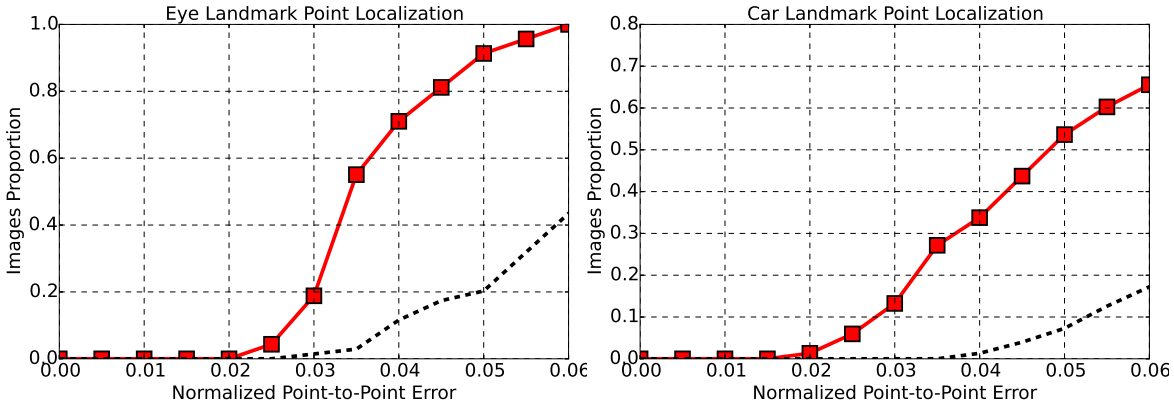
(b) Final fitting

Figure 5.5: Fitting results on the AFW facial database. These are indicative results that correspond to the curve of Fig. 5.4.

5.3.3 Results on Other Deformable Objects

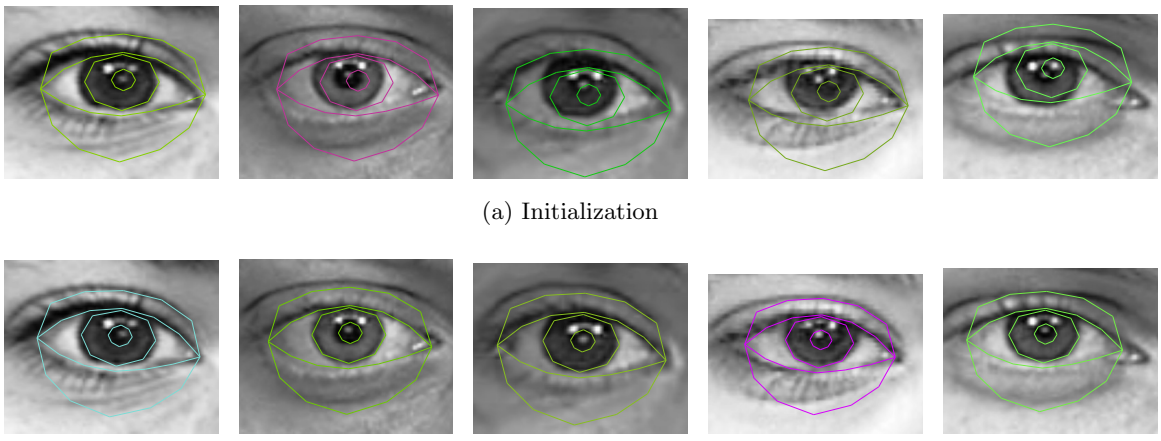
Note that APS is a flexible patch-based Deformable Model that can also be applied to the landmark localization of other objects. Herein, we show indicative results for the case of eyes and cars. In the case of cars, we employ the sideview (view 2) images from CMU database [28, 104], which we split in 450 and 151 training and testing images, respectively. For eyes, we use our in-house annotated database that consists of 38 manually annotated landmarks and it has 600 and 400 training and testing images respectively. Figure 5.6 shows the cumulative fitting error curves for both objects. For the initialization, we add Gaussian noise to the global similarity transform retrieved from the ground-truth annotations (without in-plane rotation) and apply it to the mean shape of the object. The standard deviation of the noise is set to 0.06.

Finally, Figs. 5.7 and 5.8 show some indicative fitting examples for both objects. Note that in the case of human eyes, most of the error is accumulated by the should be the upper and lower sclera, because it is a region without any distinctive features.



(a) Eyes (b) Cars

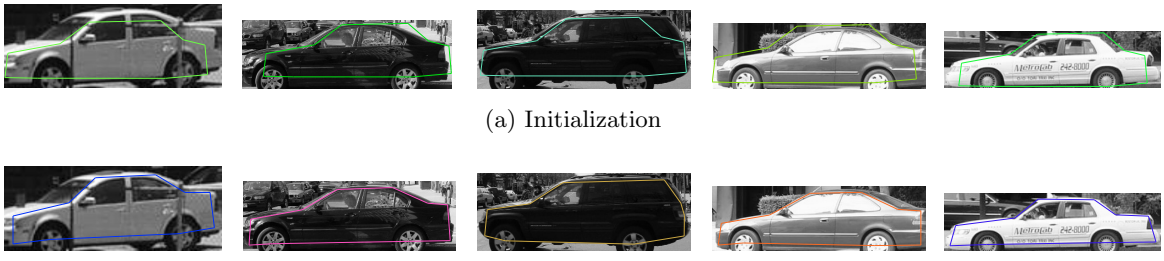
Figure 5.6: Fitting results of APS for human eyes and cars.



(a) Initialization

(b) Final fitting

Figure 5.7: Fitting results on open eyes. These are indicative results that correspond to the curve of Fig. 5.6a.



(a) Initialization

(b) Final fitting

Figure 5.8: Fitting results on cars sideview. These are indicative results that correspond to the curve of Fig. 5.6b.

5.4 Conclusions

In this chapter, we proposed a powerful part-based generative model that combines the main ideas behind PS and AAMs. APS employ a graph-based modeling of the appearance and use a variant of the Gauss-Newton technique to optimize with respect to the parameters of a statistical shape model. Our experiments show that modeling the patch-based appearance of an object with a GMRF structure is more beneficial than applying a PCA model. APS also introduce a spring-like deformation prior term that makes them robust to bad initializations. The method has a close to real-time fitting performance, which is the same independent of the graph structure that is employed, and as shown in our experiments needs only a few iterations to converge. Even though we show experiments only for the task of face alignment, we believe that the method is also suitable for other object classes, especially articulated objects (*e.g.*, hands, body pose) for which the combination of patch-based appearance with the deformation prior can make a significant difference.

Part II

Combining Generative and Discriminative Models

Chapter 6

Automatic Construction of Deformable Models

Contents

6.1 Motivation

In order to train Deformable Models with good generalization ability, a large amount of carefully annotated data is needed. Developing useful datasets and benchmarks that can contribute in the progress of an application domain is a highly time consuming and costly procedure. It requires both careful selection of the images, so that they can model the vast amount of an object's variability, and careful annotation of the various parts of the object (or landmarks). The amount of annotation that is required depends on both the object and the application. In faces, for example, where many landmark points are needed in tasks such as facial expression analysis, motion capture and expression transfer, usually more than 60 points are annotated [22, 97, 185, 133, 134, 132]. To illustrate how much time consuming careful face annotation is, according to our experience, a trained annotator may need an average of 5 minutes per image for the manual annotation of 68 landmarks. This highly depends on many factors such as the image's illumination and resolution, the presence of occlusions and the face's pose and expression. Thus, the annotation of 1000 images requires a total of about 83 hours. Note that it is very difficult to consecutively annotate for more than 4 hours. Furthermore, in many

cases, fatigue can cause errors on the accuracy and consistency of annotations and they may require correction.

In this chapter, we deal with the problem of automatically constructing a robust Deformable Model using

1. A simple bounding box object detector, and
2. A shape by means of a Point Distribution Model (PDM) (Sec. 3.1)

The detector can be as simple as the Viola-Jones object detector [162, 163, 164] which returns only a bounding box of a detected object. Such detectors are widely employed in commercial products (*e.g.*, even the cheapest digital camera has a robust face detector). Other detectors that can be used are efficient sub-window search [95] and DPM [185]. The annotations that are needed to train the object detector can be acquired very quickly, since only a bounding box containing the object is required. Specifically, after selecting the images that are going to be used, the annotation procedure takes a couple of seconds per image. The statistical shape model can be created by using only 40-50 shape examples, which can be produced by either drawing possible shape variations of the 2D shape of the object or projecting 3D CAD model instances of the object on the 2D camera plane (such an example is shown in [186] for cars). Even the annotation of the shape examples is not a time consuming task, due to their small number. Furthermore, there are unsupervised techniques to learn the shape prior (model) directly from images [77, 88].

The two most closely related works to the proposed method are the automatic construction of AAMs [19] and the so-called RASL methodology [126] for person-specific face alignment. There are two main differences between our framework and [19]:

1. We use a predefined statistical shape model instead of trying to find both the shape and appearance models. We believe that with the current available optimization techniques, it is extremely difficult to simultaneously optimize for both the texture and shape parameters.
2. We employ the robust component analysis of [158] for the appearance which deals with outliers.

Thus, even though our method is similar in concept to [19], these two differences make the problem feasible to solve. In particular, the methodology in [19] fails to create a generic

model even in controlled recording conditions, due to extremely high dimensionality of the parameters to be found and to the sensitivity of the subspace method to outliers. This was probably one of the reasons why the authors demonstrate very limited and only person-specific experiments. Furthermore, our methodology bypasses some of the limitations of [126], which requires the presence of only one low-rank subspace, hence it has been shown to work only for the case of congealing images of a single person. Finally, we argue that in order for an automatically constructed AAM methodology to be robust to both within-class and out-of-class outliers[1], which cannot be avoided in totally unsupervised settings, statistical component analysis techniques should be employed [19].

To summarize, the contributions of this work are as follows:

- We propose the first, to the best of our knowledge, methodology for automatic construction of both a generative and a discriminative AAM given only a dataset of images with the respective bounding boxes and a statistical shape model (PDM). Even though our method uses a similar texture model to [153], it is considerably different, since in that work an AAM is built using only annotated data, while our technique constructs the texture model in a fully automatic manner.

- We propose a discriminatively trained AAM methodology using the robust component analysis in [158]. Inspired by the recent success in applying a cascade of regressors [50, 171, 36, 136] to discriminatively learn a model for face alignment, we follow a similar line of research. The proposed discriminative AAM uses the robust component analysis [158] due to the fact it is trained on automatically annotated data, hence it needs to be robust to all kinds of outliers.

- Overall, the proposed methodology constructs a very powerful model, by iteratively training a generative fully automatically built AAM and then a discriminative AAM learned from the fitted shapes of the generative AAM. The method can be applied to the detection of any deformable object and thus to automatic classification/recognition applications. This is the first, to the best of our knowledge, fully automatic methodology for creating deformable model that outperforms state-of-the-art methodologies that were trained directly on the manually annotated data.

The content of this chapter is based on the following publication:

[1] Within-class outliers refer to outliers present in the image of an object such as occlusion. Out-of-class outliers refer to images of irrelevant objects or to background.

- **E. Antonakos**, and S. Zafeiriou. "Automatic Construction of Deformable Models In-The-Wild", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, pp. 1813-1820, June 2014.

The rest of the chapter is structured as follows: Section 6.2 gives an overview of the proposed method where Sections 6.2.1 and 6.2.2 elaborate on the generative and discriminative models, respectively. Section 6.3 shows extended experimental results. Finally, Section 6.4 draws conclusions.

6.2 Method

Assuming the existence of a statistical shape model of an object (PDM), our method automatically trains a generative AAM and in extension a discriminative AAM, by only using a dataset of totally unconstrained in-the-wild images containing the object and the corresponding bounding boxes. This is achieved by alternatingly constructing a generative and a discriminative Deformable Model. At each iteration, the training of each of the two models utilizes the fitted shapes computed with the other already trained model. This iterative procedure is demonstrated in Fig. 6.1.

Specifically, we separate our set of images and the corresponding bounding boxes in two disjoint equally-sized datasets, referred to as the *generative* and the *discriminative* that are used for the training of the respective models. The first generative model is trained on the initial shapes extracted by initializing the PDM mean shape in the bounding boxes. At each iteration, the currently trained generative model is used to find the fitted shapes on the discriminative database's images. Then, a discriminative model is trained on these shapes. At the next iteration, the currently trained discriminative model is applied on the images of the generative database to extract the shapes estimations. A new version of the generative model is then trained based on these extracted shapes of the generative dataset. At the end of this iterative procedure, we train a final generative and discriminative AAM on the unified database of both datasets.

This alternating training of each model followed by the supply of updated shapes to the other and vice versa manages to continuously improve the fitted shapes, leading to more accurate models. The role of the discriminative model is especially crucial, as it moves the generative model from the local optimum that it stuck. Next, in Sec. 6.2.1 and 6.2.2 we present the generative and discriminative models, respectively.

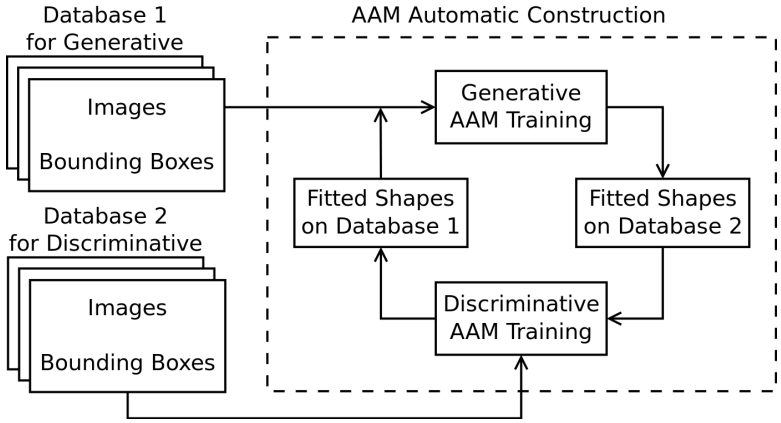


Figure 6.1: Automatic construction of deformable models. Given two sets of disjoint in-the-wild images and the object detector bounding boxes, our method automatically trains an AAM by training a generative and a discriminative model in an alternating manner.

6.2.1 Automatic Construction of a Generative AAM

The generative model employed in this work is no different than the holistic AAM presented in Chapter 4. However, in this work, the appearance model is trained by employing the robust subspace analysis proposed in [158], which uses the image gradient orientations (IGO features). Given an image $\mathbf{t}$ in vectorial form with size $L_T \times 1$, the so-called *normalized gradients* feature extraction function $\mathcal{F}(\mathbf{t})$ involves the computation of the image gradients $\mathbf{g}_x$, $\mathbf{g}_y$ and the corresponding gradient orientation $\boldsymbol{\varphi} = \arctan(\mathbf{g}_y / \mathbf{g}_x)$ as

$$\mathcal{F}(\mathbf{t}) = \frac{1}{\sqrt{L_T}} [\cos\boldsymbol{\varphi},\ \sin\boldsymbol{\varphi}]^\mathsf{T} \tag{6.1}$$

where $\cos\boldsymbol{\varphi} = [\cos\varphi(1), \ldots, \cos\varphi(L_T)]$ and $\sin\boldsymbol{\phi} = [\sin\varphi(1), \ldots, \sin\varphi(L_T)]$. Similar to Eq. 3.8, we denote the feature-based warped appearance vector as

$$\mathbf{a}(\mathbf{p}) = \mathbf{t}_\mathcal{F}(\mathcal{W}(\mathbf{p})) \ \text{ with } \ \mathbf{t}_\mathcal{F} = \mathcal{F}(\mathbf{t}) \tag{6.2}$$

that has size $2m \times 1$, where m is the number of pixels inside the reference (*i.e.*, mean) shape. Remember from Sec. 3.2 that an appearance model is then trained by performing PCA on a set of training appearance vectors that results in a subspace of n_a eigenvectors $\mathbf{U}_a \in \mathbb{R}^{2m \times n_a}$ and the mean appearance $\bar{\mathbf{a}}$. This model can be used to synthesize shape-free texture instances using Eq. 3.13.

The employment of the robust kernel of Eq. 6.1 has a key role in the successful performance of the proposed method, because it cancels-out both within-class and out-of-class outliers [158]. This is shown in the "toy" example of Fig. 6.2. In this experiment we have a dataset of 50 aligned face images. We replace 20% of these with the same baboon image and apply PCA on

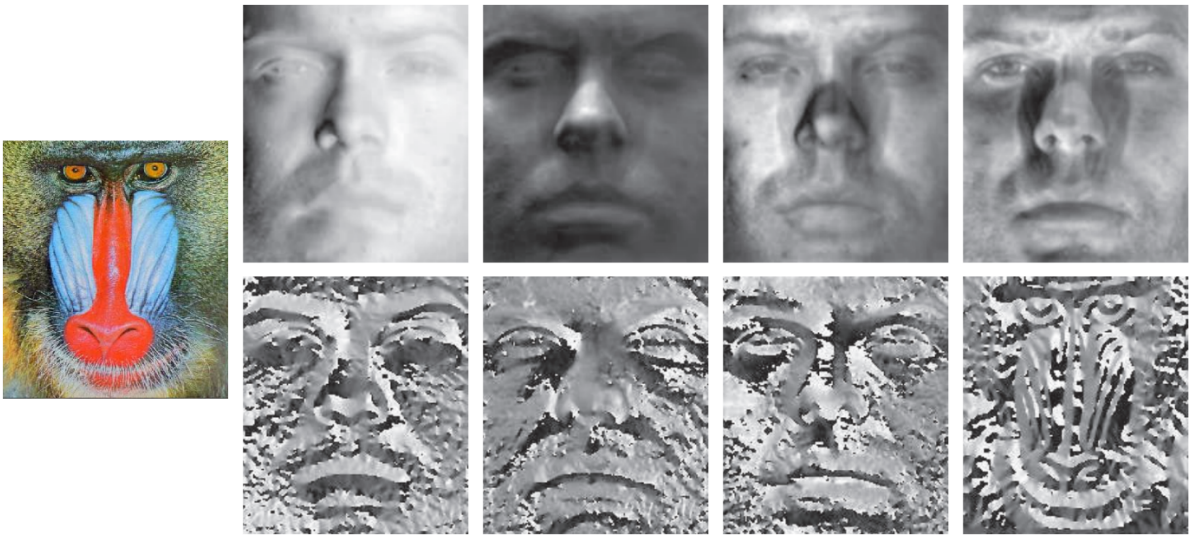


Figure 6.2: Robust kernel. Having a face dataset with 20% of the images replaced by the baboon, the top and bottom rows show 4 principal components of the PCA on intensities and normalized gradients respectively. Note that contrary to the normalized gradients subspace where the baboon is isolated, most intensities eigentextures are corrupted with the baboon. *The figure is taken from [158].*

intensities and normalized gradients. Figure 6.2 shows that the PCA eigenvectors on intensities (top row) are corrupted with the baboon information. On the contrary, the employment of normalized gradients manages to separate the baboon information from the facial subspace and isolate it (second row). In our case, during the automatic training of the generative model, we expect to have both within-class and out-of-class outliers. Since the training images are captured in totally unconstrained conditions (*i.e.*, random images from the web), we expect many of them to have occluded objects, thus within-class outliers. Furthermore, in the cases where the fitted shape is either very inaccurate or even scrambled, the warped appearance consists an out-of-class outlier. However, the employment of the robust component analysis manages to remove such outliers from the appearance subspace.

For the automatic construction of the generative AAM, we formulate an iterative optimization problem that aims to automatically construct a generative appearance model that minimizes the mean AAM fitting ℓ_2^2 norm error over all given images. Specifically, given a set of N training images $\left\{\mathbf{t}^i\right\},\ i = 1, \ldots, N$ and a statistical shape model $\{\bar{\mathbf{s}}, \mathbf{U}_s\}$, we automatically train an AAM appearance model by iteratively solving

$$\begin{aligned} \underset{\bar{\mathbf{a}}, \mathbf{U}_a, \mathbf{p}^i, \mathbf{c}^i}{\operatorname{argmin}} \quad & \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{a}^i(\mathbf{p}^i) - \bar{\mathbf{a}} - \mathbf{U}_a \mathbf{c}^i \right\|^2 \\ \text{subject to} \quad & \mathbf{U}_a^{\mathsf{T}} \mathbf{U}_a = \mathbf{E} \end{aligned} \tag{6.3}$$

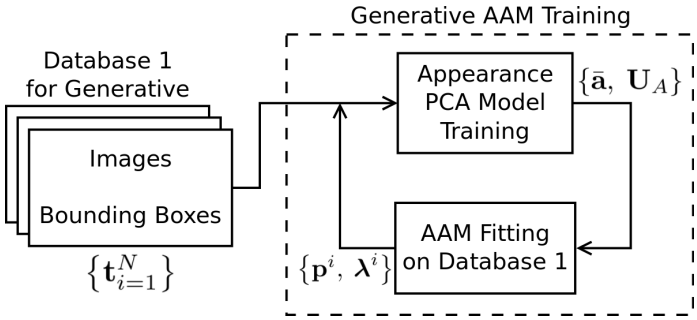


Figure 6.3: Automatic training of appearance model of Generative AAM. This diagram demonstrates the operation of Generative AAM Training step of Fig. 6.1. Given a set of images and the corresponding bounding boxes from the object detector, the method iteratively re-trains the appearance PCA model and re-performs AAM fitting on the images set to update the shapes.

in order to find the appearance subspace $\mathbf{U}_a$ and mean vector $\bar{\mathbf{a}}$ that minimize the mean ℓ_2^2 norm of the application of AAM fitting $(\mathbf{p}^i,\ \mathbf{c}^i)$ over all images. $\mathbf{a}^i(\mathbf{p}^i)$ is the warped feature representation of the training image $\mathbf{t}^i$ and $\mathbf{E}$ denotes the identity matrix. The explanation of this optimization procedure is visualized in Fig. 6.3. In brief, the algorithm iteratively trains a new PCA appearance model $\{\bar{\mathbf{a}}, \mathbf{U}_a\}$ based on the current estimate of the N shapes and then re-estimates the parameters $\{\mathbf{p}^i, \mathbf{c}^i\},\ i = 1, \ldots, N$ by minimizing the ℓ_2^2 norm between each warped image and the appearance model instance. Consequently, the optimization is solved in two steps:

(a) Fix $\{\mathbf{p}^i, \mathbf{c}^i\}$ and minimize with respect to $\{\bar{\mathbf{a}}, \mathbf{U}_a\}$ In this step we have a current estimate of $\{\mathbf{p}^i, \mathbf{c}^i\}$ for each image $i = 1, \ldots, N$. From the shape parameters estimate we extract the warped feature-based image vectors $\{\mathbf{a}^i(\mathbf{p}^i)\}$ on which we train a new PCA appearance model $\{\bar{\mathbf{a}}, \mathbf{U}_a\}$. The updated subspace is orthogonal, thus $\mathbf{U}_a^{\mathsf{T}}\mathbf{U}_a = \mathbf{E}$. In this work, we keep 150 eigenvectors per iteration.

(b) Fix $\{\bar{\mathbf{a}}, \mathbf{U}_a\}$ and minimize with respect to $\{\mathbf{p}^i, \mathbf{c}^i\}$ In this step we have a currently trained statistical appearance model $\{\bar{\mathbf{a}}, \mathbf{U}_a\}$ and aim to estimate the shape and appearance parameters $\{\mathbf{p}^i, \mathbf{c}^i\}$ for each image $i = 1, \ldots, N$ so that the ℓ_2^2 norm between each warped image and its reconstruction is minimized. Thus, we optimize

$$\underset{\mathbf{p}^i, \mathbf{c}^i}{\operatorname{argmin}} \left\| \mathbf{a}^i(\mathbf{p}^i) - \bar{\mathbf{a}} - \mathbf{U}_a \mathbf{c}^i \right\|^2,\ \forall i = 1, \ldots, N \tag{6.4}$$

This minimization can be solved with the efficient Gauss-Newton algorithm of Inverse Compositional Image Alignment (IC) [117, 124, 8, 9, 5], as presented in Chapter 4 (Sec. 4.4). Within the IC framework, Eq. 6.4 is written as

$$\underset{\mathbf{p}^i, \mathbf{c}^i}{\operatorname{argmin}} \left\| \mathbf{a}^i(\mathbf{p}^i) - \mathbf{a}_{\mathbf{c}^i}(\Delta\mathbf{p}^i) \right\|^2 \tag{6.5}$$

where $\mathbf{a}_{\mathbf{c}^i} = \bar{\mathbf{a}} + \mathbf{U}_a \mathbf{c}^i$ is the model instance and $\Delta\mathbf{p}^i$ is the increment used to inverse-compositionally update the shape parameters as

$$\mathcal{W}(\mathbf{p}^i) \leftarrow \mathcal{W}(\mathbf{p}^i) \circ \mathcal{W}(\Delta\mathbf{p}^i)^{-1} \tag{6.6}$$

As mentioned in Chapter 4, the two most commonly used IC optimization techniques are Project-Out IC (POIC) [117], where the shape and appearance parameters are decoupled, and the Simultaneous IC (SIC) [68] where the optimization is done simultaneously for the shape and appearance parameters.

We instead perform IC, by optimizing separately for shape and appearance parameters in an alternating mode, as proposed in Sec. 4.4. At each iteration, we have a fixed estimate of $\mathbf{p}^i$ and compute the appearance parameters as the least-squares solution

$$\mathbf{c}^i = \mathbf{U}_a^\mathsf{T} \left[\mathbf{a}^i(\mathbf{p}^i) - \bar{\mathbf{a}}\right] \tag{6.7}$$

Then, given the current estimate of $\mathbf{c}^i$ and taking the Taylor expansion around $\mathbf{p}^i = \mathbf{0}$, we solve for the shape increment

$$\Delta\mathbf{p}^i = -\left(\mathbf{J}^\mathsf{T}\mathbf{J}\right)^{-1} \mathbf{J}^\mathsf{T} \left[\mathbf{a}^i(\mathbf{p}^i) - \mathbf{a}_{\mathbf{c}^i}\right] \tag{6.8}$$

where

$$\mathbf{J} = \nabla\mathbf{a}_{\mathbf{c}^i} \frac{\partial\mathcal{W}}{\partial\mathbf{p}^i} \tag{6.9}$$

is the Jacobian matrix with the steepest descent images as its columns. The algorithm requires the computation of the inverse Hessian matrix $\mathbf{H} = \left(\mathbf{J}^\mathsf{T}\mathbf{J}\right)^{-1}$ and the current estimate of appearance parameters at each iteration which results in a total cost of $\mathcal{O}((n_a + n_s + 4)m + (4 + n_s)^2 m)$.

Even though the initial PCA model is expected to have many outliers and to be inaccurate, this optimization technique combined with the robust kernel of Eq. 6.1 iteratively results in an appearance model that eliminates the initial outliers. By keeping a small number of eigenvectors at each iteration, we ensure that the textures corresponding to inaccurate or scrambled shapes will not be included in our subspace. The convergence rate of this procedure is shown in Sec. 6.3.1.

A drawback of the optimization procedure is that it will stuck in a local minimum. In the following, in order to move the generative model from the local minimum, we will train a discriminative model using the already trained generative. We work under the assumption that the trained generative model is reliable enough to provide us with a sufficient number

of good fittings in a new disjoint set. It is obvious that we need a disjoint set to train the discriminative model, since training it in the same dataset as the generative would result in overfitting.

6.2.2 Robust Discriminative AAM

Motivated by the recent application of a cascade of regressors [50, 171, 36, 136] to discriminatively learn a model for face alignment, we propose a parametric discriminatively trained AAM. Even though discriminatively trained AAMs have appeared before, the difference between our method and, for example [136], is that we use simple cascaded linear regression, as in [171], and the robust component analysis [158]. Note that other feature descriptors can also be used, such as HOG [46] and SIFT [109]. Intuitively, the goal of the discriminative model is to move the generative model from the local minimum that it converged in the previous iteration and boost it towards a better minimum. We automatically select the appearance vectors on which it is trained so that as few outliers as possible are included. This selection is achieved by keeping the textures with the best ℓ_2^2 norm fitting error.

Fitting Discriminative AAM

During the training procedure, the method aims to learn a number of K regression steps so that the initial shape parameters of all the training images converge to their ground-truth values. Each of these cascade solutions consists of a generic descent direction term $\mathbf{R}_k$ and a bias term $\mathbf{b}_k$. Given an unseen image, the fitting process involves K additive steps to find an updated vector of shape and similarity parameters

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \mathbf{R}_{k-1}\mathbf{c}_{k-1} + \mathbf{b}_{k-1},\ k = 1, \ldots, K \tag{6.10}$$

where the appearance parameters are retrieved from the inverse projection of the image's warped feature-based texture to a given appearance subspace as in Eq. 6.7. In the first step, the update

$$\Delta\mathbf{p}_1 = \mathbf{R}_0\mathbf{c}_0 + \mathbf{b}_0 \tag{6.11}$$

is added to the initial parameters vectors as

$$\mathbf{p}_1 = \mathbf{p}_0 + \Delta\mathbf{p}_1 \tag{6.12}$$

The initial shape parameters vector $\mathbf{p}_0$ is computed from the image's bounding box, which practically initializes the rotation, translation and scaling values and leaves the rest equal to zero, thus

$$\mathbf{p}_0 = \left[p_0^1, \ldots, p_0^4, \mathbf{0}^{1:n_s}\right]^\mathsf{T} \tag{6.13}$$

The fitting algorithm has a real-time computational complexity of $\mathcal{O}((4+n_s)(n_a+2m))$ per iteration.

Training Discriminative AAM

Assume we have a set of N training images $\{\mathbf{t}^i\}$, $i=1,\ldots,N$ and their ground-truth shapes $\{\mathbf{s}_{tr}^i\}$ which correspond to a set of parameters $\{\mathbf{p}_{tr}^i\}$. For each image in the database, we generate M different parameters initializations $\{\mathbf{p}_0^{i,j}\}$, $j=1,\ldots,M$. This is done by sampling M different bounding boxes from a Normal distribution trained to describe the variance of various face detectors and retrieving the corresponding initialization shape parameters. To learn the sequence of generic descent directions and bias terms, we employ the Monte Carlo approximation of the ℓ_2^2-loss which results in solving the least-squares problem

$$\underset{\mathbf{R}_k,\mathbf{b}_k}{\operatorname{argmin}} \sum_{i=1}^{N}\sum_{j=1}^{M} \left\| \mathbf{p}_{tr}^i - \mathbf{p}_k^{i,j} - \mathbf{R}_k\mathbf{c}_k^{i,j} - \mathbf{b}_k \right\|^2 \tag{6.14}$$

for $k=1,\ldots,K$. At each iteration and for each image, we update the parameters vector $\mathbf{p}_k^{i,j}$ using the rule of Eq. 6.10 and compute the current appearance parameters from Eq. 6.7.

Shapes Selection

Due to the discriminative nature of this AAM, the ground-truth shapes $\{\mathbf{s}_{tr}^i\}$ need to include as few outliers as possible. This is achieved by applying least-squares based Subspace Clustering [110] on the final appearance model instances. Assume that we have estimated the appearance parameters $\{\mathbf{c}^i\}$ by fitting the generative AAM to the discriminative database's training images $\{\mathbf{t}^i\}$, $i=1,\ldots,N$. This set of parameters corresponds to a set of appearance model instances $\{\mathbf{a}_{\mathbf{c}^i}\}$. By concatenating these appearance vectors to a single matrix

$$\mathbf{A} = \left[\mathbf{a}_{\mathbf{c}^1}{}^{\mathsf{T}}, \mathbf{a}_{\mathbf{c}^2}{}^{\mathsf{T}}, \ldots, \mathbf{a}_{\mathbf{c}^N}{}^{\mathsf{T}}\right]^{\mathsf{T}} \tag{6.15}$$

we compute the block-diagonal affinity matrix (graph) by solving the least-squares regression problem

$$\min_{\mathbf{Z}} \|\mathbf{A} - \mathbf{A}\mathbf{Z}\|_F^2 + \mu \|\mathbf{Z}\|_F^2 \tag{6.16}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. This problem has a closed form solution

$$\mathbf{Z} = \left(\mathbf{A}^{\mathsf{T}}\mathbf{A} + \mu\mathbf{E}\right)^{-1}\mathbf{A}^{\mathsf{T}}\mathbf{A} \tag{6.17}$$

where $\mathbf{E}$ denotes the identity matrix. This affinity matrix provides a measure of the similarity between each pair of appearance vectors. Then we apply Normalized Spectral Clustering

(Normalized Cuts) [140] on $\mathbf{W} = \frac{1}{2}(\mathbf{Z} + \mathbf{Z}^{\mathsf{T}})$ to cluster our appearance vectors in two classes: those that include outliers and those that do not. Finally, we keep the shapes that correspond to the vectors without outliers, which ensures a discriminative model with better performance.

6.3 Experimental Results

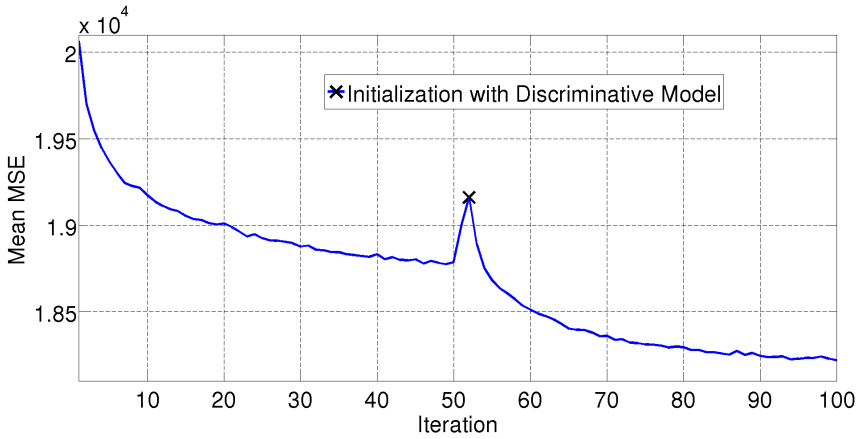
6.3.1 Convergence of AAM Automatic Construction

Firstly, in order to create a facial shape PDM, we use 50 annotated images of the LFPW database, appropriately selected to demonstrate various deformations and expressions, and apply PCA. Note that one could also project shape instances of a statistical 3D shape model [125, 32] to the 2D plane. Then, we automatically build a facial AAM with the proposed method (Fig. 6.1) using the images of LFPW and HELEN training sets (2810 images in total). In order to perform the iteration between generative and discriminative model, we split these images in two equal disjoint subsets, each consisting of half of the images of each database, thus 405 and 1000 from LFPW and HELEN, respectively. We retrieve the bounding boxes by using Google Picasa's face detection.

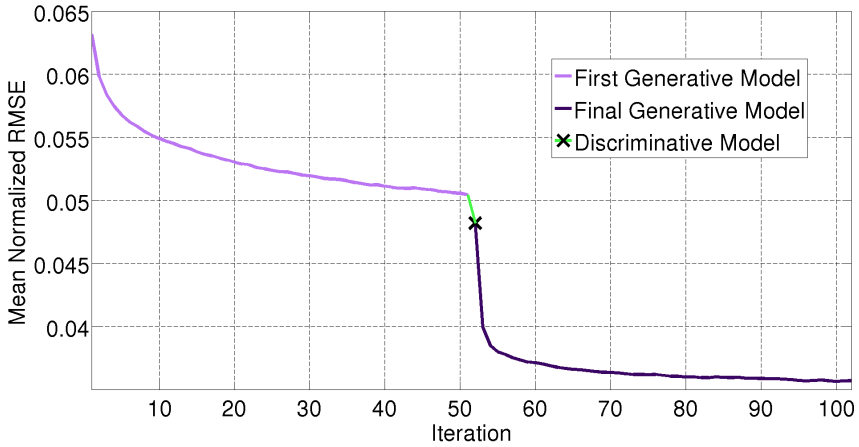
We execute the overall proposed methodology for 2 iterations in total, which involves an iterative generative model automatic construction followed by a discriminative model and then the final automatic generative model. Our experiments show that the method converges quickly and only a single application of the discriminative model is sufficient to move the generative model to a satisfactory minimum. Figure 6.4a plots the cost function vs. the number of iterations of the first generative model training on the generative database, the initialization with the first discriminative model (marked with an x) and the application of the final generative model. As can be seen the application of the discriminative step acts as a perturbation over the local optimum which in the end results to a better solution (similar to random perturbations in Simulated Annealing).

Figure 6.4b plots the normalized RMSE over the number of iterations for the generative database. The RMSE is the one defined in Eq. 3.15 with the face size as normalization constant (Eq. 3.16). As can be seen, it monotonically decreases. Furthermore, in Fig. 6.5 we demonstrate the evolution of the fitting curves of the generative database's shapes during this training procedure compared with the manually annotated shapes.

Figure 6.6 demonstrates the respective evolution of the mean appearance and the three most important eigenvectors. The last row demonstrates the subspace obtained from the PCA on



(a) Plot of the cost function per iteration. The marked point × denotes the beginning of the second iteration of the generative model.



(b) Plot of the respective point-to-point normalized RMSE.

Figure 6.4: Convergence of the automatic construction of AAM with a single application of the discriminative model. The convergence is shown with respect to the cost function minimization and the fitting accuracy.

the manual annotations of the generative database. The figure shows that the resulting facial appearance subspace gradually improves and isolates the outliers as expected, due to the employment of the robust component analysis. This is highlighted by the fact that the facial parts (eyes, nose, mouth etc.) can be distinguished more clearly in the final eigentextures, as opposed to the initial ones. The resulting appearance subspace is very similar to the annotations-based one, even though we performed only two iterations.

Furthermore, Fig. 6.8 shows the evolution of the fitted shapes for eight images during the automatic building procedure. Starting from the bounding boxes (first row), the final result of the last generative model (last row) is very accurate. This figure also highlights the importance

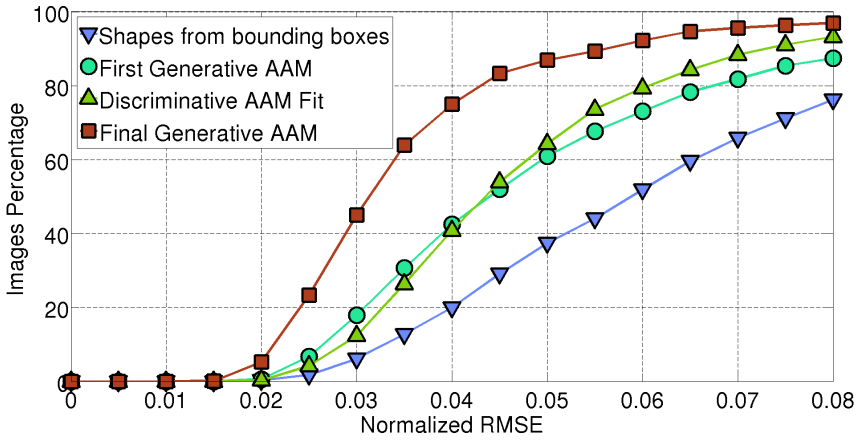


Figure 6.5: Automatic construction of AAM with a single application of the discriminative model. The plot shows the accuracy evolution of the generative database's shapes compared with their manual annotations.

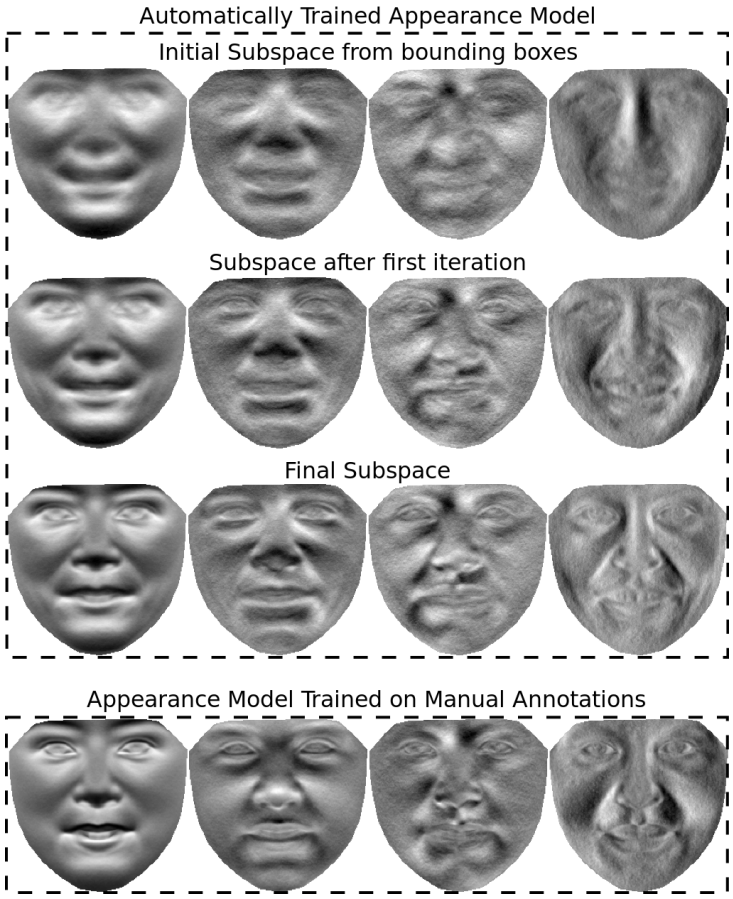


Figure 6.6: Automatic construction of AAM with a single application of the discriminative model.Visualization of the mean appearance and the three most important eigenvectors for the iterative automatically constructed AAM *(top)* and the AAM trained on manual annotations *(bottom)*.

of the discriminative model. Even though the fitted shapes that it provides are not accurate, because its discriminative nature requires carefully annotated data, however, it manages to move the generative model's shapes from the point where they stuck. We believe that the final fitted shapes shown at the last row of Fig. 6.8 are very impressive, given the automatic nature of the proposed method. Moreover, Fig. 6.7 shows the eight fitted shapes with the worst RMSE error, that were estimated automatically with the proposed procedure. As can be seen, even in the worst cases, the method provides decent shapes.

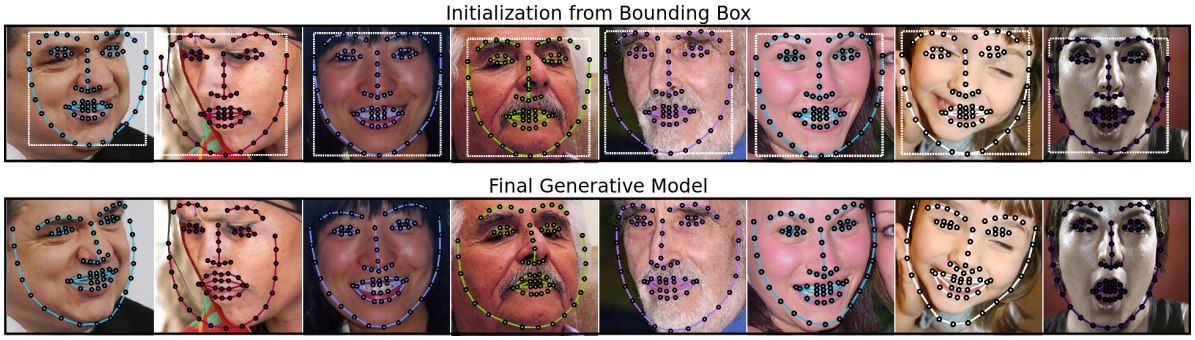


Figure 6.7: The 8 worst fitted shapes during the automatic construction of AAM with a single application of the discriminative model.

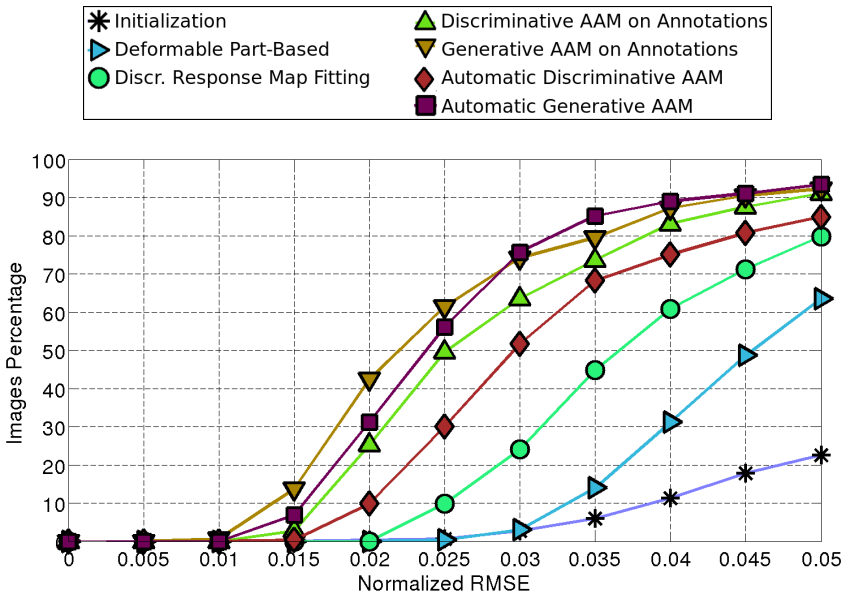
6.3.2 Comparison with Models Trained on Manual Annotations

After completing the iterations demonstrated in Figs. 6.4 and 6.6, we train a final generative and discriminative model on the 2810 images of the union of both datasets. We compare the performance of our model with the state-of-the-art method of Robust Discriminative Response Map Fitting (DRMF) for Constrained Local Models [13] and the Deformable Part-Based Models [185]. For both methods, we use the implementation provided by their authors, along with the pre-built models which are discriminatively trained on the manual annotations of much larger datasets than LFPW and HELEN datasets. Moreover, we compare with the generative and discriminative AAMs trained on the manual annotations of LFPW and HELEN trainsets. Figure 6.9 shows the normalized RMSE curves on AFW and the union of LFPW and HELEN testsets. Note that in both cases, we use Google Picasa's face detection to extract the bounding boxes that initialize the translation and scaling of the mean shape. The results show that our automatically trained models have a very good performance and greatly outperform the discriminative ones trained on manual annotations.

Finally, Figs. 6.10 and 6.11 show some indicative fitting results for the AFW dataset and the union of LFPW and HELEN databases, respectively. Again, we strongly believe that

Figure 6.8: Automatic construction of AAM with a single application of the discriminative model. The figures show the evolution of the fitted shapes for 8 images, starting from the bounding boxes. Each automatically trained generative model is performed for 50 iterations.

these results are very promising, especially considering the fact that our method's models were constructed by starting with just a bounding box per face.



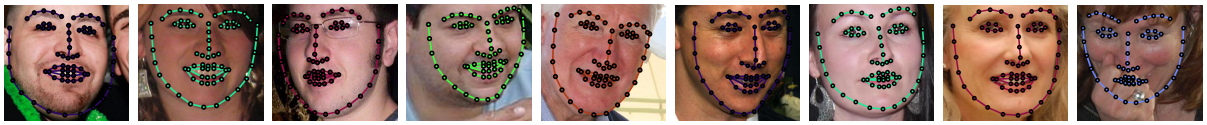
(a) AFW database

(b) LFPW and HELEN testing databases

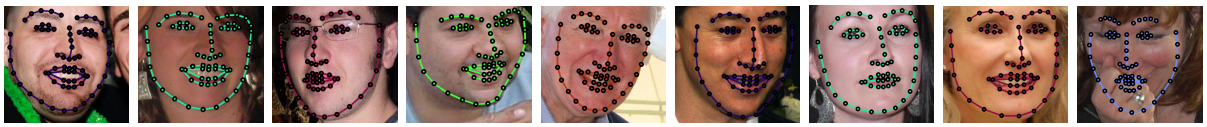
Figure 6.9: Comparison of automatically constructed deformable models (generative and discriminative) with other models trained on manual annotations.

6.4 Conclusions

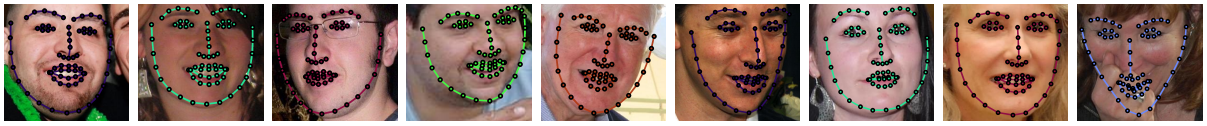
In this chapter, we proposed a method for automatic construction of Deformable Models. The method iteratively trains a generative and a discriminative AAM ending up with a powerful model. The only requirements of the method are a statistical shape model and a set of in-the-wild images with their bounding boxes, which means that it can be applied to any object. Our experiments on faces show that the method outperforms discriminative state-of-the-art methods trained on manual annotations. This is the first, to the best of our knowledge,



(a) Automatically trained generative model.



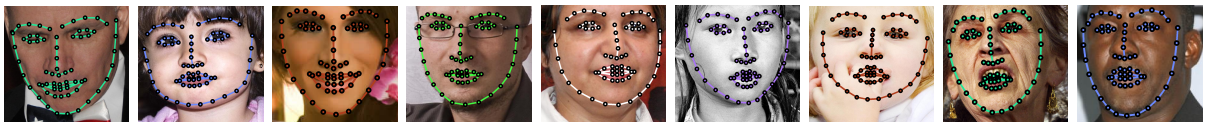
(b) Generative model trained on manual annotations.



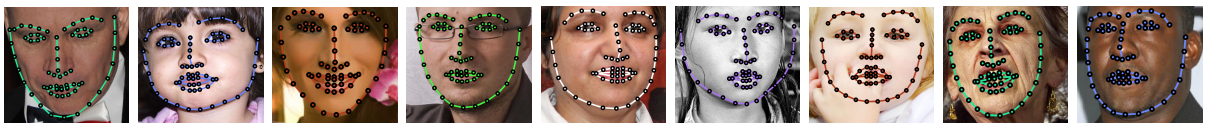
(c) Automatically trained discriminative model.

(d) Discriminative model trained on manual annotations.

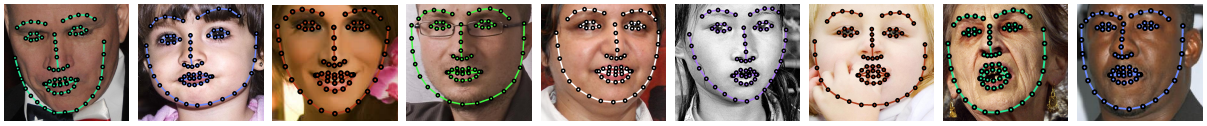
Figure 6.10: Fitting results on AFW database.



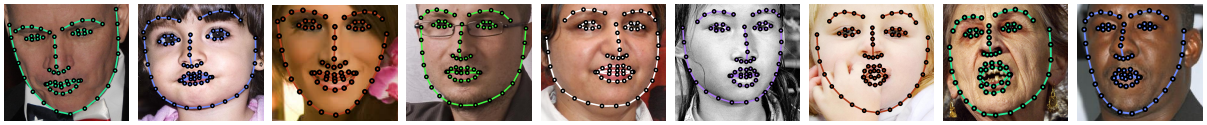
(a) Automatically trained generative model.



(b) Generative model trained on manual annotations.



(c) Automatically trained discriminative model.



(d) Discriminative model trained on manual annotations.

Figure 6.11: Fitting results on LFPW and HELEN testing databases.

methodology to automatically building a Deformable Model that demonstrates such promising results.

Chapter 7

Adaptive Cascaded Regression

Contents

7.1 Motivation

As explained in Chapter 1 (Sec. 1.1), the most commonly-used and well-studied face alignment methods can be separated in two major families:

- *Discriminative* models that employ regression in a cascaded manner.
- *Generative* models that are iteratively optimized using the Gauss-Newton algorithm.

Although both these families of techniques have been shown to achieve state-of-the-art performance, they both suffer from major weaknesses. Cascaded regression-based techniques [35, 171, 173, 50, 171, 36, 175, 176, 82, 128, 14, 152, 183] have the ability to return accurate results even with very challenging initializations, as they are coupled with a specific distribution of initializations during training. Hence, they seek to learn averaged descent directions with good generalization properties [172]. Furthermore, they are also ideal for real-time applications since a cascade of 4-5 steps has been shown to be adequate [171] and the calculation of the shape increment is usually efficient to compute. However, since the descent

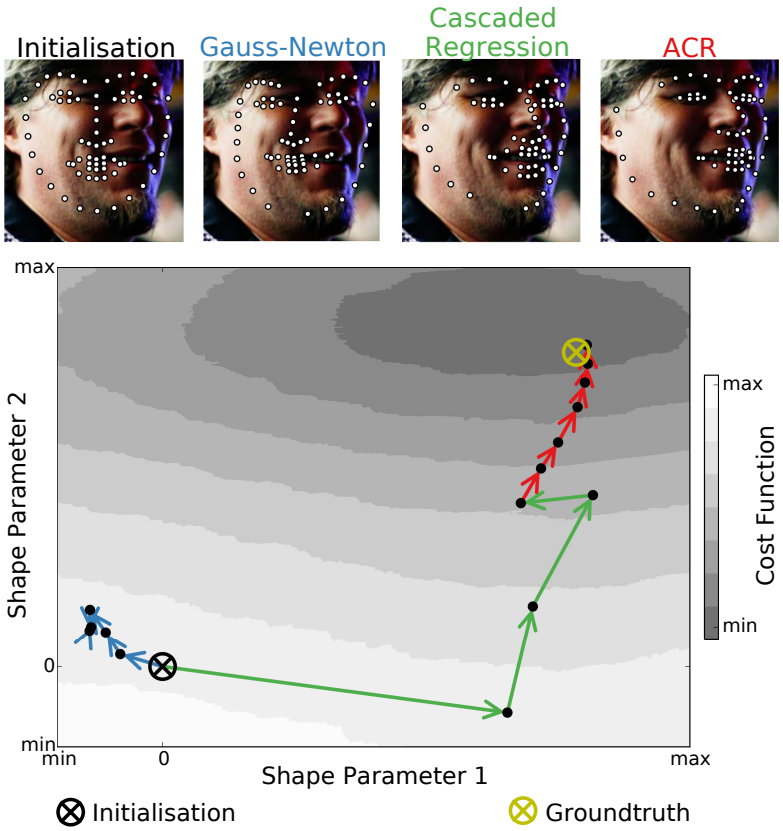


Figure 7.1: Example of descent directions obtained through optimization. The cost function, which is based on a parametric shape and appearance model, is plotted with respect to the two first shape parameters. Cascaded-regression (*green*) moves towards the correct direction but does not reach the optimum. Gauss-Newton (*blue*) diverges due to hard initialization. However, applying Gauss-Newton right after the final regression step (*red*) converges to the ground-truth optimum. Motivated by this behavior, we propose a unified model that combines the regression-based discriminative and Gauss-Newton generative formulations.

directions are not adaptive to the test image, they are not always able to recover the fine details of the object. They also have no theoretical guarantee of local convergence in test images. Theoretical guarantee for convergence exists only for the train set [172]. On the other hand, generative models [42, 38, 39, 117, 18, 124, 155, 3, 4, 156, 153, 8, 154, 9, 5] optimized with the Gauss-Newton algorithm have been shown to be much more accurate when initialized close to an optimum [156, 8, 9, 3] and it can be proved that their iterative procedure convergences to a local minimum with an expected quadratic rate. However, the linearization of the cost function required for Gauss-Newton optimization causes generative models to be highly sensitive to their initializations. In general, if a Gauss-Newton algorithm is not initialized within close proximity of an acceptable local minima, the resulting alignment will be poor.

In this chapter, we present a unified model that combines the generative and discriminative

formulation. Our motivation comes from the example of Figure 7.1. In this example, we plot the cost function that we aim to optimize based on a parametric shape model and a projected-out appearance subspace [117]. Note that the cost function is common for the discriminative and the generative models (more details will be given in Secs. 7.2.1, 7.2.2, Eq. 7.22). The cost function is plotted with respect to the first two shape parameters. We also draw the descent directions provided by cascaded regression, followed by a Gauss-Newton optimization. Note that even though the initialization is far from the ground-truth optimum, the cascaded regression manages to quickly converge towards the correct direction, but is not able to actually reach the optimum. By initializing the Gauss-Newton algorithm from the final result of the cascaded regression, we manage to reach the local optimum that corresponds to the ground-truth, which translates to a lower point-to-point error. On the other hand, the application of Gauss-Newton directly from the initial point completely diverges due to the large distance from the optimum.

Motivated by the experiment of Figure 7.1, we believe that the best result can be achieved by combining the discriminative cascaded regression with the iterative Gauss-Newton optimization within a unified model. Our proposed model employs a fully parametric cascade of regression-based descent directions, which are further adapted by the Gauss-Newton descent directions provided by the Hessian of the Gauss-Newton method. This adaptation allows the model to be robust to very challenging initializations and to converge to the local minimum which can recover accurate landmark localization for the fine details of an object. Inspired by our method's nature, we name it Adaptive Cascaded Regression (ACR).

In summary, the contributions of this chapter are:

- We propose a Deformable Model that takes advantage of the best of both worlds: cascaded discriminative and generative models. Our model combines these two approaches under a natural unified formulation. To the best of our knowledge, this is the first attempt of combining these two optimization worlds under a single cost function.
- We show that our method overcomes the disadvantages of both cascaded regression and Gauss-Newton optimization and exploits their strengths in terms of accuracy and convergence.
- We report state-of-the-art performance on the task of face alignment, using the most recent benchmark challenge 300-W [134, 133, 132].

The content of this chapter is based on the following publication:

- **E. Antonakos**[5], P. Snape[5], G. Trigeorgis, and S. Zafeiriou. "Adaptive Cascaded Regression", *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA, *Oral*, September 2016.

The rest of the chapter is structured as follows: Section 7.2 first presents the discriminative approach (Sec. 7.2.1) and then the generative one, in order to formulate the proposed model (Sec. 7.2.3). Section 7.3 shows extended experimental results and proves the state-of-the-art performance of the proposed Deformable Model. Finally, Section 7.4 concludes the chapter.

7.2 Method

In the following sections, we follow the notation of Secs. 3.1 and 3.2 for the shape and appearance models, respectively. Specifically, we employ the same shape representation

$$\mathbf{s} = \left[\boldsymbol{\ell}_1^\mathsf{T}, \boldsymbol{\ell}_2^\mathsf{T}, \ldots, \boldsymbol{\ell}_n^\mathsf{T}\right]^\mathsf{T} = [x_1, y_1, x_2, y_2, \ldots, x_n, y_n]^\mathsf{T} \tag{7.1}$$

as well as a shape model of the form of Eq. 3.3. With some abuse of notation, let us redefine the shape generation formulation of Eq. 3.4 as a function, *i.e.*,

$$\mathbf{s}(\mathbf{p}) = \bar{\mathbf{s}} + \mathbf{U}_s\mathbf{p} \tag{7.2}$$

where $\mathbf{p} = [p_1, p_2, \ldots, p_{n_s}]^\mathsf{T}$ is the $n_s \times 1$ vector of *shape parameters* that control the linear combination of the eigenvectors.

Moreover, we employ a part-based appearance representation as explained in Sec. 3.2.3. With some abuse of notation, we redefine Eq. 3.11 as a function, *i.e.*,

$$\mathbf{f}(\mathbf{s}) = \left[\mathcal{F}(\mathbf{t}_{\boldsymbol{\ell}_1})^\mathsf{T}, \mathcal{F}(\mathbf{t}_{\boldsymbol{\ell}_2})^\mathsf{T}, \ldots, \mathcal{F}(\mathbf{t}_{\boldsymbol{\ell}_n})^\mathsf{T}\right]^\mathsf{T} \tag{7.3}$$

We also create an appearance model following the description of Sec. 3.2.4, which can be used to generate new appearance vectors with the function

$$\mathbf{a}(\mathbf{c}) = \bar{\mathbf{a}} + \mathbf{U}_a\mathbf{c} \tag{7.4}$$

where $\mathbf{c} = [c_1, c_2, \ldots, c_{n_a}]^\mathsf{T}$ is the $n_a \times 1$ vector of *appearance parameters*. Finally, let us define

$$\mathbf{P} = \mathbf{E} - \mathbf{U}_a\mathbf{U}_a^\mathsf{T} \tag{7.5}$$

which is the orthogonal complement of the appearance subspace $\mathbf{U}_a$, where $\mathbf{E}$ denotes the $mn \times mn$ identity matrix. This projection operator is used in order to project-out the appearance variance in the following methods.

In the following sections, we present details of the discriminative (Sec. 7.2.1) and generative (Sec. 7.2.2) models in order to formulate our unified model (Sec. 7.2.3).

7.2.1 Cascaded Regression Discriminative Model

Herein, we present a fully parametric cascaded regression model. We employ an appearance model and learn a regression function that regresses from the object's projected-out appearance to the parameters of a linear shape model. Let us assume that we have a set of N training images $\{\mathbf{I}^1, \ldots, \mathbf{I}^N\}$ and their corresponding annotated shapes $\{\mathbf{s}^1, \ldots, \mathbf{s}^N\}$. By projecting each ground-truth shape to the shape basis $\mathbf{U}_s$, we get the set of ground-truth shape parameters $\{\mathbf{p}_1^*, \ldots, \mathbf{p}_N^*\}$. Moreover, we aim to learn a cascade of K levels, *i.e.*, $k = 1, \ldots, K$. During the training process of each level, we generate a set of P perturbed shape parameters $\mathbf{p}_{i,j}^k,\ j = 1, \ldots, P,\ i = 1, \ldots, N$, which are sampled from a distribution that models the statistics of the detector employed for initialization. By defining

$$\Delta\mathbf{p}_{i,j}^k = \mathbf{p}_i^* - \mathbf{p}_{i,j}^k,\ j = 1, \ldots, P,\ i = 1, \ldots, N \tag{7.6}$$

to be a set of shape parameters increments, the least-squares problem that we aim to solve during training at each cascade level k is

$$\underset{\mathbf{W}^k}{\operatorname{argmin}} \sum_{i=1}^{N} \sum_{j=1}^{P} \left\| \Delta\mathbf{p}_{i,j}^k - \mathbf{W}^k \mathbf{P} \left(\mathbf{f}_i(\mathbf{s}(\mathbf{p}_{i,j}^k)) - \bar{\mathbf{a}} \right) \right\|_2^2 \tag{7.7}$$

where $\mathbf{P}$ is the projection operator defined in Eq. 7.5 and $\mathbf{f}_i(\cdot)$ denotes the vector of concatenated feature-based patches extracted from the training image $\mathbf{I}^i$, as defined in Eq. 7.3. Note that the bias term of the above objective function is substituted by the mean appearance vector $\bar{\mathbf{a}}$. By denoting

$$\hat{\mathbf{f}}_{i,j,k} = \mathbf{P} \left(\mathbf{f}_i(\mathbf{s}(\mathbf{p}_{i,j}^k)) - \bar{\mathbf{a}} \right) \tag{7.8}$$

to be the projected-out residual, then the closed-form solution to the above least-squares problem is given by

$$\mathbf{W}^k = \left(\sum_{i=1}^{N} \sum_{j=1}^{P} \Delta\mathbf{p}_{i,j}^k \hat{\mathbf{f}}_{i,j,k}^{\mathsf{T}} \right) \left(\sum_{i=1}^{N} \sum_{j=1}^{P} \hat{\mathbf{f}}_{i,j,k} \hat{\mathbf{f}}_{i,j,k}^{\mathsf{T}} \right)^{-1} \tag{7.9}$$

for each level of the cascade $k = 1, \ldots, K$.

During testing, given the current estimate of the shape parameters $\mathbf{p}_k$ that was computed at cascade level k, we create the feature-based image vector $\mathbf{f}(\mathbf{s}(\mathbf{p}_k))$, subtract the mean

appearance vector $\bar{\mathbf{a}}$, project-out the appearance variation and estimate the shape parameters increment as

$$\Delta \mathbf{p}_k = \mathbf{W}^k \mathbf{P} \left(\mathbf{f}(\mathbf{s}(\mathbf{p}_k)) - \bar{\mathbf{a}} \right) \tag{7.10}$$

Then, the shape parameters vector is updated as

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \Delta \mathbf{p}_{k-1} \tag{7.11}$$

where we set $\mathbf{p}_0 = \mathbf{0}$ at the first iteration. The computational complexity of Eq. 7.10 per cascade level is $\mathcal{O}(n_s mn)$, thus the complexity per test image is $\mathcal{O}(K n_s mn)$.

7.2.2 Gauss-Newton Generative Model

The optimization of an AAM aims to minimize the reconstruction error of the input image with respect to the shape and appearance parameters, *i.e.*,

$$\underset{\mathbf{p},\mathbf{c}}{\operatorname{argmin}} \left\| \mathbf{f}(\mathbf{s}(\mathbf{p})) - \bar{\mathbf{a}} - \mathbf{U}_a \mathbf{c} \right\|_2^2 \tag{7.12}$$

where we employ the appearance model of Eq. 7.4 and $\mathbf{f}(\cdot)$ denotes the vectorized form of the input image as defined in Eq. 3.7. This cost function is commonly optimized in an iterative manner using the Gauss-Newton algorithm. This algorithm introduces an incremental update for the shape and appearance parameters, *i.e.*,, $\Delta \mathbf{p}$ and $\Delta \mathbf{c}$ respectively, and solves the problem with respect to $\Delta \mathbf{p}$ by first linearizing using first-order Taylor expansion around $\Delta \mathbf{p} = \mathbf{0}$. The Gauss-Newton optimization can be performed either in a forward or in an inverse manner, depending on whether the incremental update of the shape parameters is applied on the image or the model, respectively. In this work, we focus on the *inverse* algorithm, however the forward case can be derived in a similar way.

We follow the derivation of Chapter 4 that was first presented in [124] and later was readily employed in [155, 156]. By applying the incremental shape parameters on the part of the model, the cost function of Eq. 7.12 becomes

$$\underset{\Delta\mathbf{p},\Delta\mathbf{c}}{\operatorname{argmin}} \left\| \mathbf{f}(\mathbf{s}(\mathbf{p})) - \bar{\mathbf{a}}(\Delta \mathbf{p}) - \mathbf{U}_a(\Delta \mathbf{p})(\mathbf{c} + \Delta \mathbf{c}) \right\|_2^2 \tag{7.13}$$

where $\bar{\mathbf{a}}(\Delta \mathbf{p}) = \bar{\mathbf{a}}(\mathbf{s}(\Delta \mathbf{p}))$ and $\mathbf{U}_a(\Delta \mathbf{p}) = \mathbf{U}_a(\mathbf{s}(\Delta \mathbf{p}))$. Given the part-based nature of our model, the compositional update of the parameters at each iteration is reduced to a simple subtraction [156], as

$$\mathbf{p} \leftarrow \mathbf{p} - \Delta \mathbf{p} \tag{7.14}$$

By taking the first order Taylor expansion around $\Delta\mathbf{p} = \mathbf{0}$, we arrive at

$$\underset{\Delta\mathbf{p},\Delta\mathbf{c}}{\operatorname{argmin}} \|\mathbf{f}(\mathbf{s}(\mathbf{p})) - \bar{\mathbf{a}} - \mathbf{U}_a(\mathbf{c} + \Delta\mathbf{c}) - \mathbf{J}_a\Delta\mathbf{p}\|_2^2 \tag{7.15}$$

where

$$\mathbf{J}_a = \mathbf{J}_{\bar{\mathbf{a}}} + \sum_{i=1}^{m} c_i\mathbf{J}_i \tag{7.16}$$

is the model Jacobian. This Jacobian consists of the mean appearance Jacobian $\mathbf{J}_{\bar{\mathbf{a}}} = \frac{\partial\bar{\mathbf{a}}}{\partial\mathbf{p}}$ and the Jacobian of each appearance eigenvector denoted as $\mathbf{J}_i,\ i = 1,\ldots,m$.

By employing the projection operator of Eq. 7.5 in order to work on the orthogonal complement of the appearance subspace $\mathbf{U}_a$ and using the fact that $\mathbf{P}\mathbf{U}_a = \mathbf{P}^\mathsf{T}\mathbf{U}_a = \mathbf{0}$, the above cost function can be expressed as

$$\underset{\Delta\mathbf{p}}{\operatorname{argmin}} \|\mathbf{f}(\mathbf{s}(\mathbf{p})) - \bar{\mathbf{a}} - \mathbf{J}_a\Delta\mathbf{p}\|_{\mathbf{P}}^2 \tag{7.17}$$

The solution to this least-squares problem is

$$\Delta\mathbf{p} = \hat{\mathbf{H}}_a^{-1}\hat{\mathbf{J}}_a^\mathsf{T}(\mathbf{f}(\mathbf{s}(\mathbf{p})) - \bar{\mathbf{a}}) \tag{7.18}$$

where

$$\hat{\mathbf{J}}_a = \mathbf{P}\mathbf{J}_a \text{ and } \hat{\mathbf{H}}_a = \hat{\mathbf{J}}_a^\mathsf{T}\hat{\mathbf{J}}_a \tag{7.19}$$

are the projected-out Jacobian and Hessian matrices respectively. Note that even though $\mathbf{J}_{\bar{\mathbf{a}}}$ and $\mathbf{J}_i$ can be precomputed, the complete model Jacobian $\mathbf{J}_a$ depends on the appearance parameters $\mathbf{c}$ and has to be recomputed at each iteration. Given the current estimate of $\Delta\mathbf{p}$, the solution of $\mathbf{c}$ with respect to the current estimate $\mathbf{c}_c$ can be retrieved as

$$\mathbf{c} = \mathbf{c}_c + \mathbf{U}_a^\mathsf{T}\left(\mathbf{f}(\mathbf{s}(\mathbf{p})) - \bar{\mathbf{a}} - \mathbf{U}_\mathbf{a}\mathbf{c}_c - \mathbf{J}_a\Delta\mathbf{p}\right) \tag{7.20}$$

Thus, the computational complexity of computing Eq. 7.18 per iteration is $\mathcal{O}(n_s n_a mn + n_s^2 mn)$. The authors in [156] suggest that by approximating the projected-out Hessian matrix as $\hat{\mathbf{H}}_a \approx \mathbf{J}_a^\mathsf{T}\mathbf{J}_a$, reduces the complexity to $\mathcal{O}(n_a mn + n_s^2 mn)$ without any significant loss in performance.

The inverse approach that we followed, which was first proposed in [124], is different from the well-known project-out inverse compositional method of [117]. Specifically, in our case, the linearization of the cost function is performed *before* projecting-out. On the contrary, the authors in [117] followed the approximation of *projecting-out first and then linearising*, which eliminates the need to recompute the appearance subspace Jacobian. However, the project-out method proposed by [117] does not generalize well and is not suitable for generic facial alignment.

Given the fact that $\mathbf{P}^\mathsf{T} = \mathbf{P}$ and $\mathbf{P}^\mathsf{T}\mathbf{P} = \mathbf{P}$, then the solution of Eq. 7.18 can be expanded as

$$\Delta\mathbf{p} = (\mathbf{J}_a^\mathsf{T}\mathbf{P}\mathbf{J}_a)^{-1}\mathbf{J}_a^\mathsf{T}\mathbf{P}(\mathbf{f}(\mathbf{s}(\mathbf{p})) - \bar{\mathbf{a}}) \tag{7.21}$$

Thus, it is worth mentioning that the solution of the regression-based model in Eq. 7.10 is equivalent to the Gauss-Newton solution of Eq. 7.18 if the regression matrix has the form

$$\mathbf{W}^k = (\mathbf{J}_a^\mathsf{T}\mathbf{P}\mathbf{J}_a)^{-1}\mathbf{J}_a^\mathsf{T} \tag{7.22}$$

which further reveals the equivalency of the two cost functions of Eqs. 7.7 and 7.17.

7.2.3 Adaptive Cascaded Regression

As previously explained, both the AAMs of Section 7.2.2 and traditional SDMs as in 7.2.1 suffer from a number of disadvantages. To address these disadvantages, we propose ACR which combines the two previously described discriminative and generative optimization problems into a single unified cost function. Specifically, by employing the regression-based objective function of Eq. 7.7 along with the Gauss-Newton analytical solution of Eq. 7.18, the training procedure of ACR aims to minimize

$$\sum_{i=1}^{N}\sum_{j=1}^{P}\left\|\Delta\mathbf{p}_{i,j}^k - \left(\lambda^k\mathbf{W}^k - (1-\lambda^k)\mathbf{H}_{i,j}^{-1}\mathbf{J}_{i,j}^\mathsf{T}\right)\hat{\mathbf{f}}_{i,j,k}\right\|_2^2 \tag{7.23}$$

with respect to $\mathbf{W}^k$, where

$$\hat{\mathbf{f}}_i(\mathbf{s}(\mathbf{p}_{i,j}^k)) = \mathbf{P}\left(\mathbf{f}_i(\mathbf{s}(\mathbf{p}_{i,j}^k)) - \bar{\mathbf{a}}\right) \tag{7.24}$$

is the projected-out residual and $\mathbf{H}_{i,j}$ and $\mathbf{J}_{i,j}$ denote the Hessian and Jacobian matrices, respectively, of the Gauss-Newton optimization algorithm per image $i = 1, \ldots, N$ and per perturbation $j = 1, \ldots, P$. λ_k is a hyperparameter that controls the weighting between the regression-based descent directions and the Gauss-Newton descent directions at each level of the cascade $k = 1, \ldots, K$. The negative sign in front of the Gauss-Newton descent directions is due to the fact that the shape parameters update within the inverse Gauss-Newton optimization is performed with subtraction, as shown in Eq. 7.14.

Training

During training, ACR aims to learn a cascade of K linear regressors given the Gauss-Newton descent directions of each training image at each level. Let us assume that we have a set of N training images $\{\mathbf{I}_1, \ldots, \mathbf{I}_N\}$ along with the corresponding ground truth shapes $\{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$. We also assume that we have recovered the ground truth shape parameters for each training image $\{\mathbf{p}_1^*, \ldots, \mathbf{p}_N^*\}$ by projecting the ground truth shapes against the shape model.

Perturbations Before performing the training procedure, we generate a set of initializations per training image, so that the regression function of each cascade level learns how to estimate the descent directions that optimize from these initializations to the ground truth shape parameters. Consequently, for each training image, we first align the mean shape $\bar{\mathbf{s}}$ with the ground truth shape $\mathbf{s}^i$, project it against the shape basis $\mathbf{U}_s$ and then generate a set of P random perturbations for the first four shape parameters that correspond to the global similarity transform. Thus, we have a set of shape parameter vectors $\mathbf{p}_{i,j}^k,\ \forall i = 1,\ldots,N,\ \forall j = 1,\ldots,P$. Since the random perturbations are applied on the first four parameters, the rest of them remain zero, i.e., $\mathbf{p}_{i,j}^k = [p1_{i,j}^k, p2_{i,j}^k, p3_{i,j}^k, p4_{i,j}^k, \mathbf{0}_{n_s-4\times 1}^\mathsf{T}]^\mathsf{T}$. Moreover, the perturbations are sampled from a distribution that models the statistics of the detector that will be used for automatic initialization at testing time. This procedure is necessary only because we have a limited number of training images and can be perceived as training data augmentation. It could be avoided if we had more annotated images and a single initialization per image using the detector would be adequate. The perturbations are performed once at the beginning of the training procedure of ACR. The steps that are applied at each cascade level $k = 1,\ldots,K$, in order to estimate $\mathbf{W}^k$, are the following:

Step 1: Shape Parameters Increments Given the set of vectors $\mathbf{p}_{i,j}^k$, we formulate the set of shape parameters increments vectors $\Delta\mathbf{p}_{i,j}^k = \mathbf{p}_i^* - \mathbf{p}_{i,j}^k,\ \forall i = 1,\ldots,N,\ \forall j = 1,\ldots,P$ and concatenate them in a $n_s \times NP$ matrix

$$\Delta\mathbf{P}_k = \left[\Delta\mathbf{p}_{1,1}^k\ \cdots\ \Delta\mathbf{p}_{N,P}^k\right] \tag{7.25}$$

Step 2: Projected-Out Residuals The next step is to compute the part-based appearance vectors from the perturbed shape locations $\mathbf{f}_i(\mathbf{s}(\mathbf{p}_{i,j}^k))$ and then the projected-out residuals of Eq. 7.24 $\forall i = 1,\ldots,N,\ \forall j = 1,\ldots,P$. These vectors are then concatenated in a single $mn \times NP$ matrix as

$$\hat{\mathbf{F}}_k = \left[\hat{\mathbf{f}}_1(\mathbf{s}(\mathbf{p}_{1,1}^k))\ \cdots\ \hat{\mathbf{f}}_N(\mathbf{s}(\mathbf{p}_{N,P}^k))\right] \tag{7.26}$$

Step 3: Gauss-Newton Descent Directions Compute the Gauss-Newton solutions for all the images and their perturbed shapes and concatenate them in a $n_s \times NP$ matrix as

$$\mathbf{G}_k = (1-\lambda^k)\begin{bmatrix} [\mathbf{H}_{1,1}^{-1}\mathbf{J}_{1,1}^\mathsf{T}\hat{\mathbf{f}}_1(\mathbf{s}(\mathbf{p}_{1,1}^k))]^\mathsf{T} \\ \vdots \\ [\mathbf{H}_{i,j}^{-1}\mathbf{J}_{i,j}^\mathsf{T}\hat{\mathbf{f}}_i(\mathbf{s}(\mathbf{p}_{i,j}^k))]^\mathsf{T} \\ \vdots \\ [\mathbf{H}_{N,P}^{-1}\mathbf{J}_{N,P}^\mathsf{T}\hat{\mathbf{f}}_N(\mathbf{s}(\mathbf{p}_{N,P}^k))]^\mathsf{T} \end{bmatrix}^\mathsf{T} \tag{7.27}$$

Based on the expanded solution of Eq. 7.21, the calculation of the Jacobian and Hessian per image involves the estimation of the appearance parameters using Eq. 7.20 and then

$$\begin{aligned} \mathbf{J}_{i,j} &= \mathbf{J}_a \\ \mathbf{H}_{i,j} &= \mathbf{J}_{i,j}^{\mathsf{T}} \mathbf{P} \mathbf{J}_{i,j} \end{aligned} \tag{7.28}$$

where $\mathbf{J}_a$ is computed based on Eq. 7.16 for each image.

Step 4: Regression Descent Directions By using the matrices definitions of Eqs. 7.25, 7.26 and 7.27, the cost function of ACR in Eq. 7.23 takes the form

$$\underset{\mathbf{W}^k}{\operatorname{argmin}} \left\| \Delta \mathbf{P}_k - \lambda^k \mathbf{W}^k \hat{\mathbf{F}}_k + \mathbf{G}_k \right\|_2^2 \tag{7.29}$$

The closed-form solution of the above least-squares problem is

$$\mathbf{W}^k = \frac{1}{\lambda^k} \left(\Delta \mathbf{P}_k + \mathbf{G}_k \right) \left(\hat{\mathbf{F}}_k^{\mathsf{T}} \hat{\mathbf{F}}_k \right)^{-1} \hat{\mathbf{F}}_k^{\mathsf{T}} \tag{7.30}$$

Note that the regression matrix of this step is estimated only in case $\lambda_k \geq 0$. If $\lambda_k = 0$, then we directly set $\mathbf{W}_k = \mathbf{0}_{n_s \times mn}$

Step 5: Shape Parameters Update The final step is to generate the new estimates of the shape parameters per training image. By employing Eqs. 7.30 and 7.28, this is achieved as

$$\mathbf{p}_{i,j}^{k+1} = \mathbf{p}_{i,j}^k + \left(\lambda_k \mathbf{W}^k - (1 - \lambda_k) \mathbf{H}_{i,j}^{-1} \mathbf{J}_{i,j}^{\mathsf{T}} \right) \mathbf{f}_i(\mathbf{s}(\mathbf{p}_{i,j}^k)) \tag{7.31}$$

$\forall i = 1, \ldots, N$ and $\forall j = 1, \ldots, P$. After obtaining $\mathbf{p}_{i,j}^{k+1}$, steps 1-5 are repeated for the next cascade level.

Weighting Hyperparameters $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_K]$ is a set of weights that control the linear combination between the regression-based descent directions and the Gauss-Newton descent directions. They are treated as a set of hyperparameters that are fine-tuned prior to fitting. Intuitively, given the properties of regression and Gauss-Newton descent directions explained above and shown in Fig. 7.1, we expect the regression-based descent directions to dominate the optimization on the first few iterations, as they are able to move towards the correct direction with steps of large magnitude. Then, the Gauss-Newton descent steps are necessary in order to converge to an accurate local minimum. The hyperparameters λ_k are fine-tuned by running extensive cross-validation experiments that perform grid search using the mean point-to-point error normalized with the interocular distance as evaluation criterion.

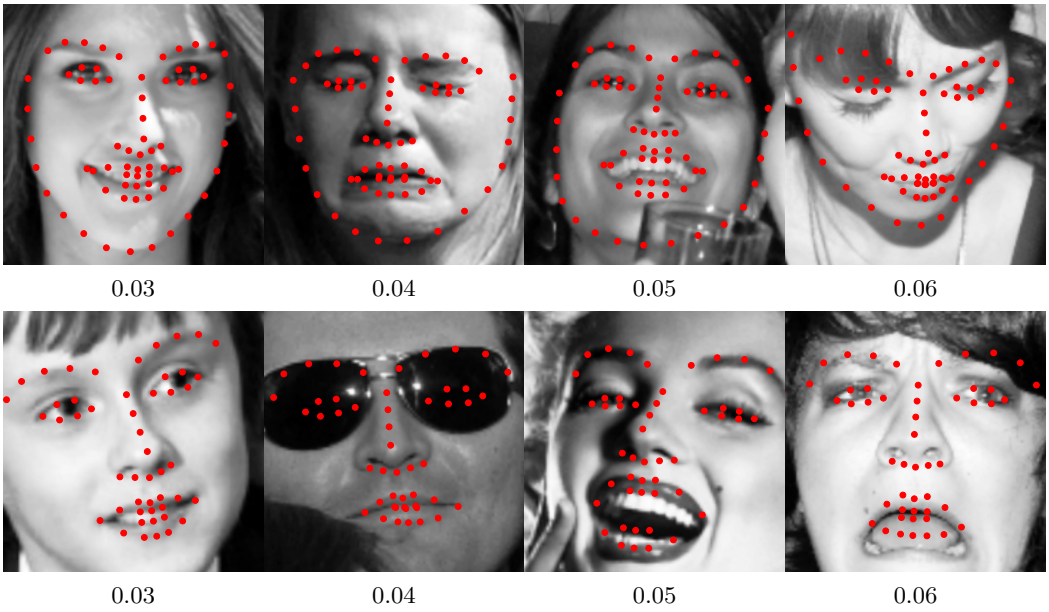


Figure 7.2: Representative examples of increasing normalised errors. (*top*) 68-points. (*bottom*) 49-points.

Fitting

In the fitting phase, given an unseen testing image $\mathbf{I}$ and its initial shape parameters $\mathbf{p}^0 = [p_1^0, p_2^0, p_3^0, p_4^0, \mathbf{0}]^\mathsf{T}$, we compute the parameters update at each cascade level $k = 1, \ldots, K$ as

$$\mathbf{p}^k = \mathbf{p}^{k-1} + \left(\lambda_k \mathbf{W}^k - (1 - \lambda_k)\mathbf{H}^{-1}\mathbf{J}^\mathsf{T}\right)\mathbf{f}(\mathbf{s}(\mathbf{p}^{k-1})) \tag{7.32}$$

where the Jacobian and Hessian are computed as described in Step 3 of the training procedure (Eq. 7.28). The computational complexity per iteration is $\mathcal{O}(n_s mn(n_a + n_s + 1))$.

7.3 Experimental Results

Evaluation Protocol To maintain consistency with the results of the original 300-W competition, we report Cumulative Error Distribution (CED) graphs using the point-to-point error normalized by the interocular distance defined by the outer eye corners. The mean error often reported in recent works [128, 183] is highly biased by alignments that completely fail. Therefore, we believe that the failure rate as shown in [35] is a much more informative error metric. To complement the failure rate, we propose the area under the curve (AUC), which enables simpler comparison of CED curves that are otherwise difficult to compare. We fix a maximum error that we believe represents a failed fitting, and thus the higher the AUC, the more fittings are concentrated within this acceptable fitting area. In all experiments, CED curves and AUC

errors are reported up to 0.06. Examples of different errors are given in Figure 7.2, which shows that 0.06 represents an alignment failure.

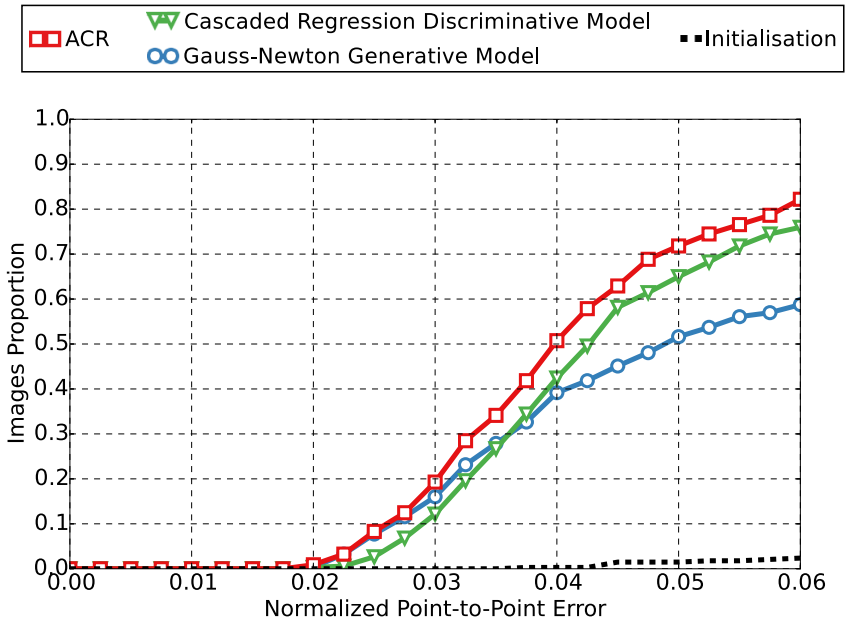
Implementation Details The following settings were used for training ACR. 20 components were kept for the shape model and 300 for the appearance model. After running extended cross-validation experiments, we found that the best performance is obtained by using a cascade of 14 levels and setting $\lambda = [1,\ 0.75,\ 0.5,\ 0.25]$ for the first four and $\lambda = 0$ for the rest. The first two were performed on the image at half scale, the rest at full scale. The patch sizes were $[(32 \times 32), (24 \times 24), (24 \times 24), (16 \times 16)]$ for the first four cascades and (24×24) for the rest. Dense SIFT [160, 109] features were used for all methods. When performing a regression, a ridge parameter of 100 was used. In order to increase the size of the training data, we augment it by perturbing the provided bounding boxes of the 300-W competition with uniform noise of 0.005 for scaling and 0.07 for translation (scaled by the bounding box size). The same options were used for training the generative model (AAM) and the discriminative cascaded-regression (SDM).

7.3.1 Self Evaluation

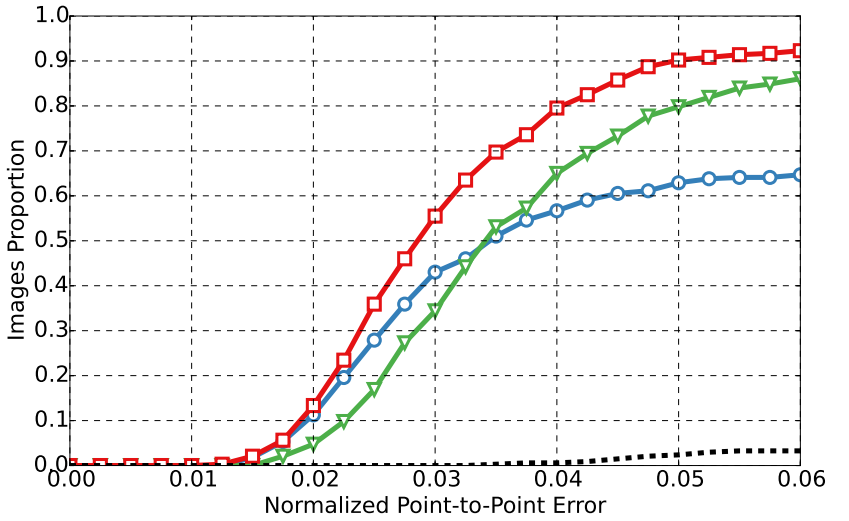
In the following experiments we performed self evaluations, comparing ACR to both the generative AAM and the discriminative SDM. In each case, we trained the SDM or AAM in the same manner as the corresponding part of ACR. We trained all 3 of the methods on LFPW (training, 811 images), HELEN (training, 2000 images) and IBUG (135 images). The testing database was chosen as AFW (337 images) as recent works (*e.g.*, [152]) have shown that AFW is still a challenging dataset. Figure 7.3 shows the CED curves for the SDM, AAM and ACR for both the 68-point and 49-point errors. Figure 7.3 clearly shows the improved performance of ACR over both SDM and AAM. To demonstrate the sensitivity of generative methods to initializations, we repeated the experiment on AFW by generating 10 initializations per image and then sorted the initialization errors (low-to-high). We then binned the initialization errors and plotted the final error of the SDM, AAM and ACR with respect to increasing initial errors. Figure 7.4 shows the results of this initialization experiment. Here we can clearly see that, as the initialization error increases, the AAM is incapable of converging towards an acceptable local-minima. It also shows that, although the SDM performs well, ACR outperforms it across all initialization errors.

7.3.2 Comparison with State-of-the-Art

In this section, we compare the performance of ACR against the state-of-the-art methods:



(a) 68-point error



(b) 49-point error

Figure 7.3: ACR, AAM (Gauss-Newton) and SDM (Discriminative), trained identically, tested on the images of AFW. Initialization given by the bounding boxes of [133, 132].

- Zhou *et al.* (300W 1) [182]
- Yan *et al.* (300W 2) [174]
- Coarse-to-fine Shape Searching (CFSS) [183]
- Project-Out Cascaded Regression (PO-CR) [152]

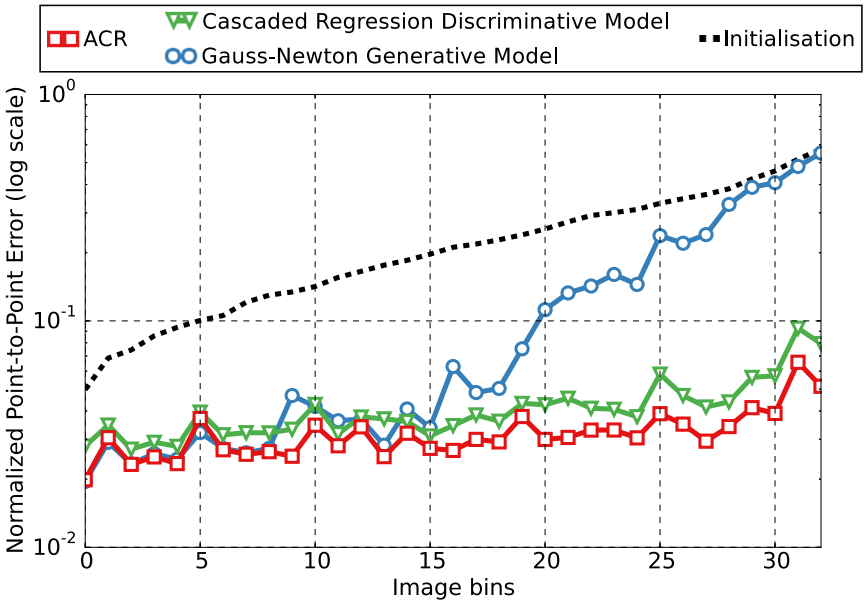


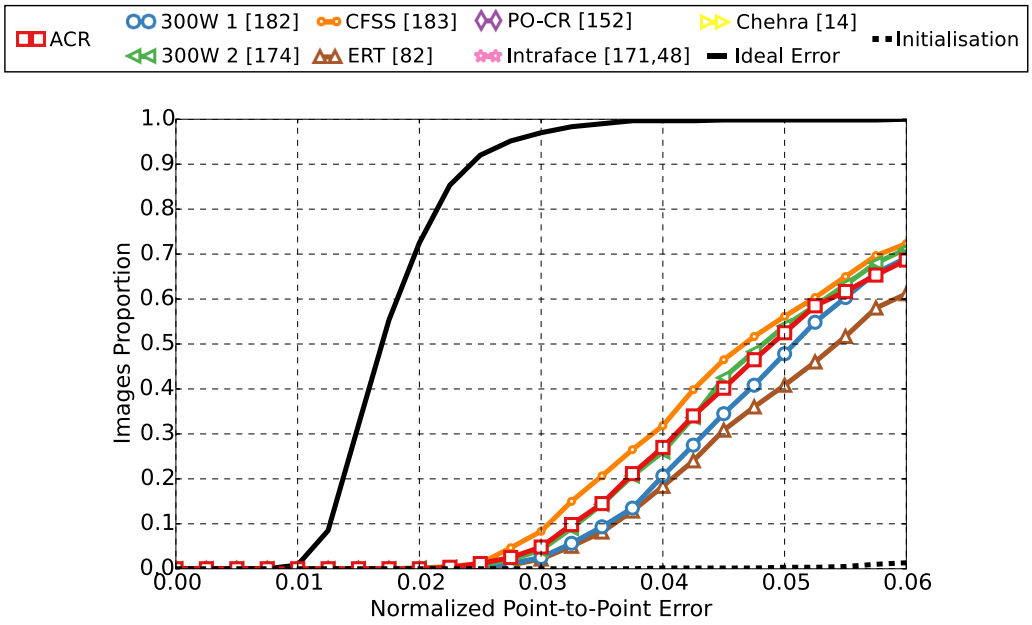
Figure 7.4: Sorted initial errors of 10 random initializations of each image in the AFW dataset. As the initial error increases, the AAM is unable to converge, whereas ACR is both robust to initializations and consistently accurate.

- Ensemble of Regression Trees (ERT) [82]
- Intraface [171, 48]
- Chehra [14]

ACR was trained using LFPW (training), HELEN (training), AFW and IBUG and both testing and training were initialized using the bounding boxes provided by 300-W [133, 133, 132]. The public implementations of some of these methods only return 49-points, and thus they are not included in the 68-point error results. We perform this experiment on the 300-W [133, 133, 132] (Sec. 7.3.2), LFPW testset [22] (Sec. 7.3.2) and HELEN testset [97] (Sec. 7.3.2) databases.

300-W Database

The 300-W face alignment challenge [133, 133, 132] utilizes a dataset of testing images to perform evaluations. The dataset includes 600 "in-the-wild" testing images and that are drawn from the same distribution as the IBUG dataset. In Figure 7.5, we see that the recently proposed CFSS method is currently the best performing method for 68-points. However, for the 49-points, ACR is the most accurate technique and slightly outperforms (300W 1), which is a much more complex deep learning method provided by industry. Table 7.1 reinforces the



(a) 68-point error

(b) 49-point error

Figure 7.5: Normalized error for the testing dataset of 300-W challenge [133, 132]. This database represents a fair benchmark for state-of-the-art face alignment methods.

results of Figure 7.5 by showing that ACR is highly accurate for the 49-points and slightly less robust than the method of [182] over all images.

Method	*AUC*	*Failure rate (%)*
ACR	**0.43**	11.0
300W 1 [182]	0.42	**9.3**
CFSS [183]	0.40	13.5
300W 2 [174]	0.38	14.2
PO-CR [152]	0.37	17.7
ERT [82]	0.28	23.7
Intraface [171, 48]	0.27	23.8
Chehra [14]	0.24	46.8
Initialisation	0.01	96.8

Table 7.1: The area under the curve (AUC) and percentage failure rate for the 49-point CED curve given in Figure 7.5. Failure rate is the % of images with error > 0.06.

Method	*mean* $\pm$ *std*	*median*	*mad*	*max*	*AUC*	*Failure rate (%)*
ACR	$\mathbf{0.0267 \pm 0.0092}$	**0.0248**	**0.0045**	0.0841	**0.60**	1.3
CFSS [183]	0.0283 ± 0.0079	0.0270	0.0046	**0.0688**	0.58	**0.4**
PO-CR [152]	0.0386 ± 0.0790	0.0279	0.0046	0.8041	0.56	2.2
ERT [82]	0.0353 ± 0.0147	0.0318	0.0060	0.1238	0.48	4.0
Intraface [171, 48]	0.0666 ± 0.1071	0.0314	0.0050	0.6062	0.46	13.4
Chehra [14]	0.0761 ± 0.1185	0.0284	0.0080	0.7344	0.44	23.7
Initialisation	0.1749 ± 0.1098	0.1449	0.0593	0.7273	0.01	94.2

Table 7.2: Various statistical measures, area under the curve (AUC) and percentage failure rate for the 49-point CED curve given in Figure 7.6 for LFPW testset. Failure rate is the % of images with error > 0.06.

LFPW Testset

Figure 7.6 shows the accuracy of each method on LFPW testset [23] in the form of a Cumulative Error Distribution (CED) curve. Table 7.2 reports some statistical measures (mean, standard deviation, median, median absolute deviation, max), the area under the curve (AUC) and the failure rate of all methods based on Fig. 7.6. Note that ACR is more accurate than all the other methods by a large margin. Especially in the band of low errors, it achieves an improvement of even about 10%. ACR is also slightly less robust than CFSS. Another interesting observation is the very high maximum errors for all the cascaded regression methods (PO-CR, Chehra, Intraface) that indicate that in case of a fitting failure, the final shape is completely scrambled.

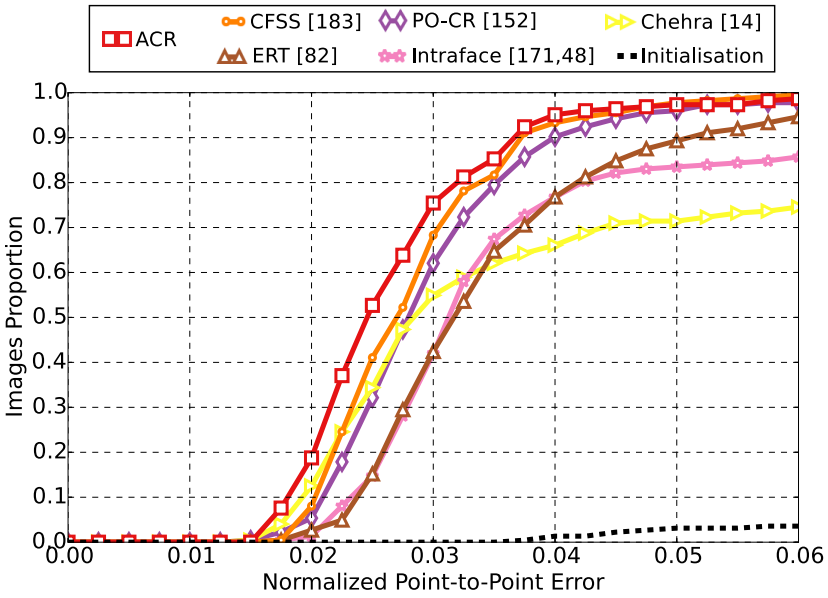


Figure 7.6: Normalized error for the testing LFPW dataset based on 49 points.

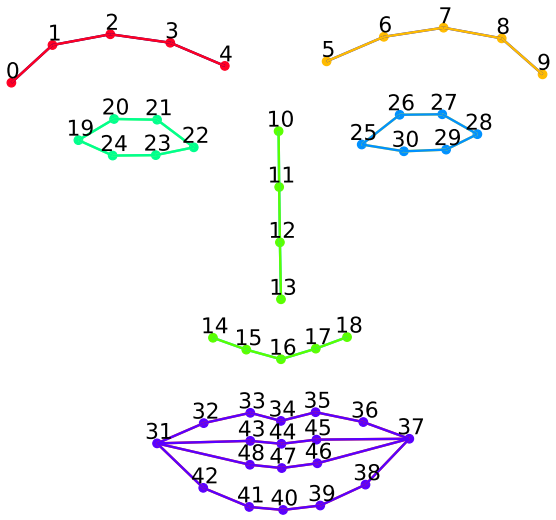


Figure 7.7: The numbering and grouping of the landmarks in the 49-points configuration. The coloring and numbering of this figure is to be linked with Figures 7.8 and 7.11.

Figure 7.8 reports the mean and standard deviation of the error per landmark point for all the methods. The numbering and coloring of each landmark point is linked with the mean shape of Figure 7.7. Once again, note that we only take into consideration the fittings with final error smaller than 0.06. ACR is very accurate on all facial parts. On the contrary, all the cascaded-regression based techniques (PO-CR, Intraface, Chehra) heavily fail on the internal mouth points and are not equally accurate on the eyebrows and eyes. Finally, Fig. 7.9 shows

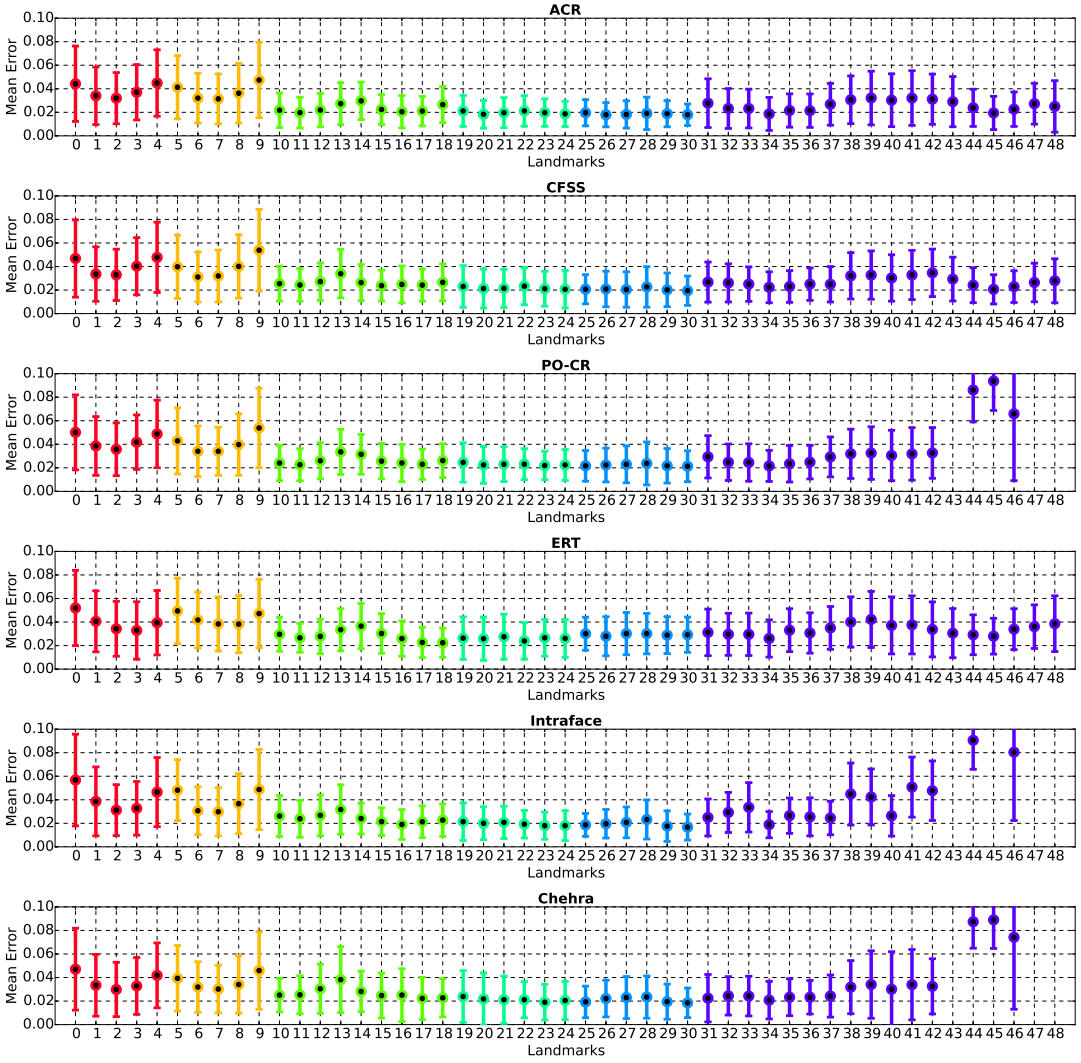


Figure 7.8: Mean and standard deviation of the normalized error per landmark point for all the methods on LFPW testset. The coloring and numbering of the landmarks is linked with the mean shape of Figure 7.7.

the 10 best and 10 worst fitting results achieved by ACR. As it can be observed, even the worst results have not heavily failed.

HELEN Testset

Figure 7.10 shows the accuracy of each method on the HELEN testset [97] in the form of a Cumulative Error Distribution (CED) curve. Table 7.3 reports some statistical measures (mean, standard deviation, median, median absolute deviation, max), the area under the curve (AUC) and the failure rate of all methods based on Fig. 7.6. In this case, ACR is more accurate and more robust than all the other methods, since it achieves the best AUC as well

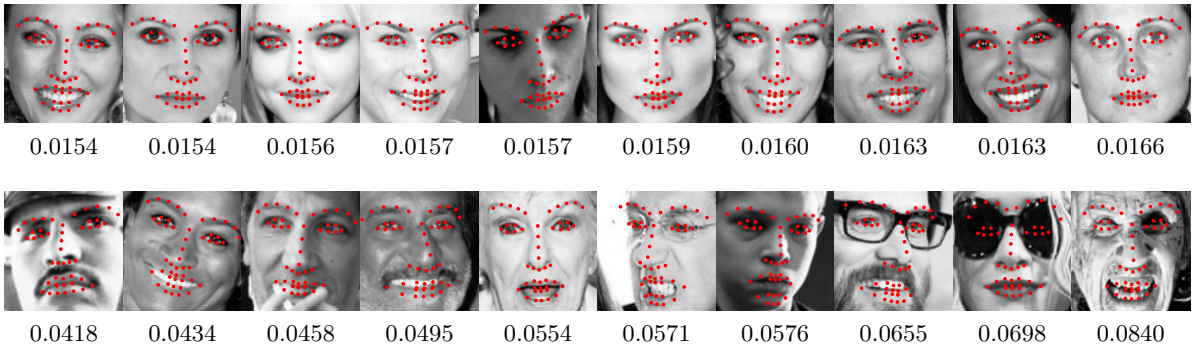


Figure 7.9: 10 *best* (top), and 10 *worst* (bottom) fitting results of ACR on LFPW testset.

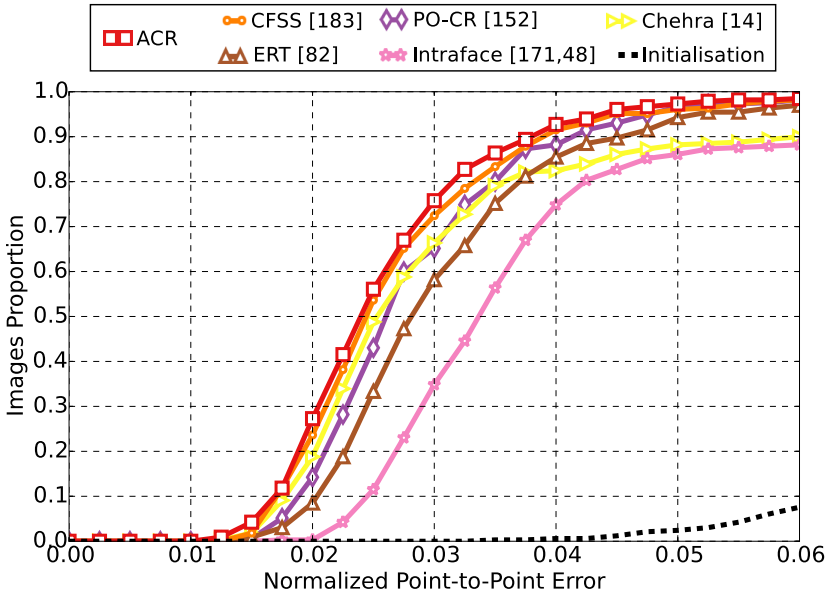


Figure 7.10: Normalized error for the testing HELEN dataset based on 49 points.

as the lowest failure rate.

Figure 7.11 reports the mean and standard deviation of the error per landmark point for all the methods. Similar to the LFPW case, the numbering and coloring of each landmark point is linked with the mean shape of Figure 7.7. The results are again similar and indicate that ACR is more accurate on all facial parts, especially on the mouth region. Finally, Fig. 7.12 shows the 10 best and 10 worst fitting results achieved by ACR.

Method	*mean ± std*	*median*	*mad*	*max*	*AUC*	*Failure rate (%)*
ACR	$\mathbf{0.0262 \pm 0.0104}$	**0.0240**	0.0050	**0.0968**	**0.61**	**1.2**
CFSS [183]	0.0288 ± 0.0318	0.0244	**0.0048**	0.5644	0.60	1.5
PO-CR [152]	0.0299 ± 0.0287	0.0260	0.0051	0.5199	0.58	0.6
ERT [82]	0.0323 ± 0.0236	0.0280	0.0055	0.3732	0.54	1.8
Intraface [171, 48]	0.0666 ± 0.1094	0.0336	0.0060	0.7718	0.45	11.5
Chehra [14]	0.0391 ± 0.0507	0.0251	0.0054	0.4853	0.55	9.4
Initialisation	0.1757 ± 0.1050	0.1475	0.0603	0.5656	0.02	90.9

Table 7.3: Various statistical measures, area under the curve (AUC) and percentage failure rate for the 49-point CED curve given in Figure 7.10 for HELEN testset. Failure rate is the % of images with error > 0.06.

7.4 Conclusions

In this chapter, we have shown that by combining the descent directions of cascaded regression and Gauss-Newton optimization, we can achieve both robustness to challenging initializations and accuracy with respect to fine details. We report state-of-the-art performance on the task of facial alignment, using the most recent benchmark challenge and have experimentally verified that ACR outperforms both AAM and SDM for a range of initializations.

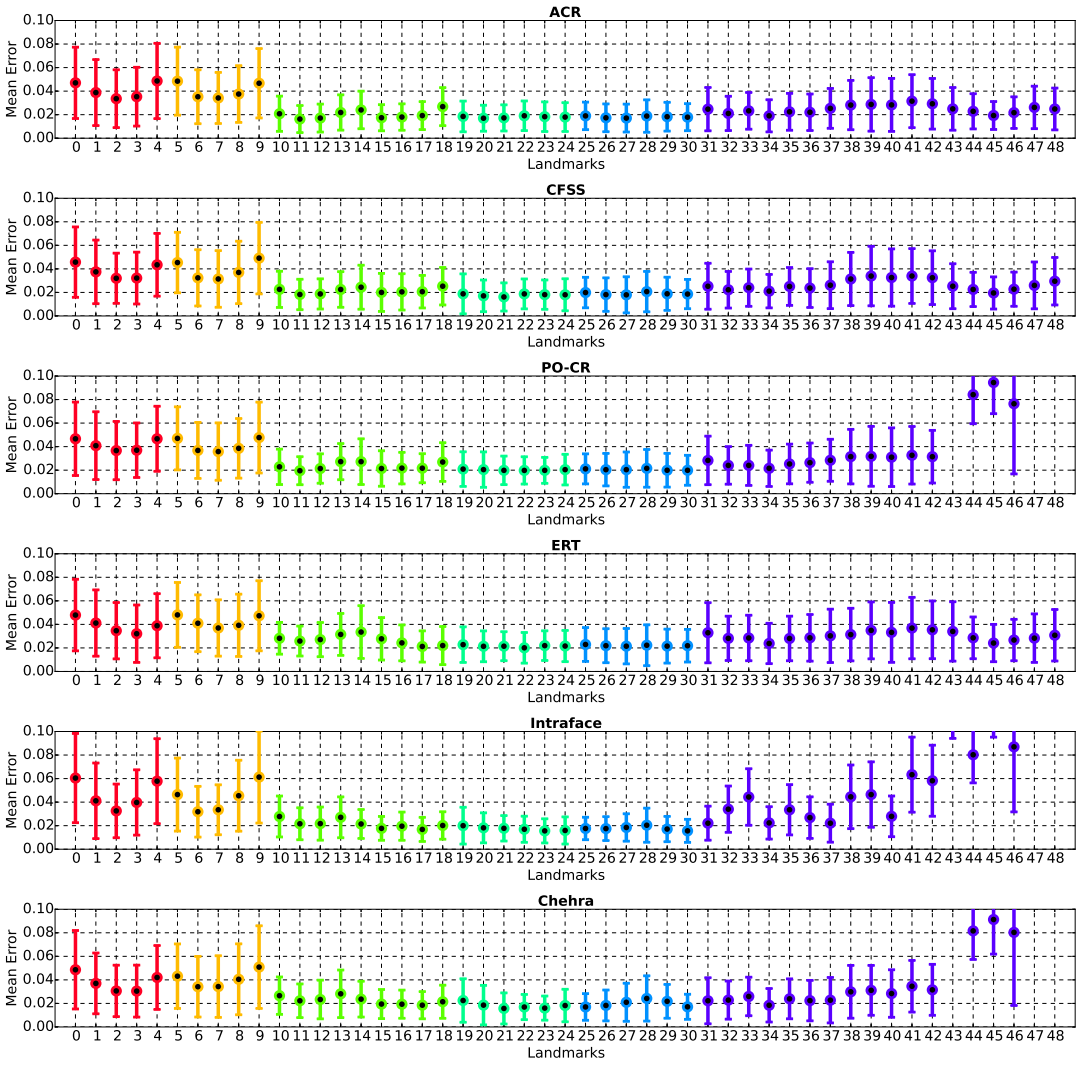
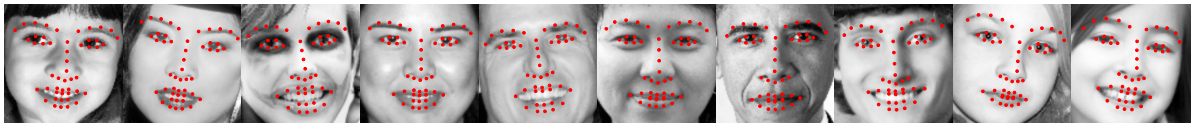
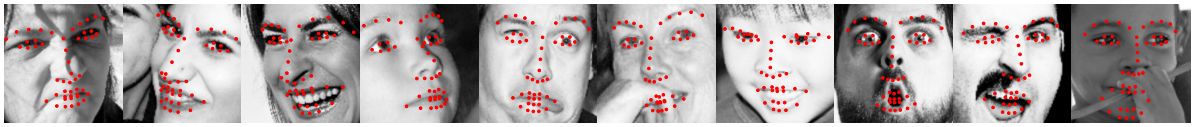


Figure 7.11: Mean and standard deviation of the normalized error per landmark point for all the methods on HELEN testset. The coloring and numbering of the landmarks is linked with the mean shape of Figure 7.7.



0.0119 0.0122 0.0124 0.0129 0.0133 0.0136 0.0144 0.0144 0.0145 0.0147



0.0493 0.0504 0.0518 0.0528 0.0592 0.0619 0.0666 0.0791 0.0878 0.0968

Figure 7.12: 10 *best* (top), and 10 *worst* (bottom) fitting results of ACR on HELEN testset.

Chapter 8

Conclusion

Contents

In this thesis, we proposed novel and robust Deformable Models that achieve state-of-the-art performance on the task of landmark localization and semi-automatic annotation of large databases. The presented work is focused on the deformable object of human face, due to the fact that there are numerous manually annotated facial databases with thousands of images. The thesis was split in two parts.

In Part I, we focused on developing powerful generative Deformable Models that employ both holistic and part-based appearance representations. Specifically, in Chapter 4 we showed that the combination of LK (Gauss-Newton) optimization with highly-descriptive dense features greatly improves the performance of holistic AAMs. We proved, both theoretically and experimentally, that by extracting the features from the input image once and then warping the features image has better performance and lower computational complexity than computing features from the warped image at each iteration. Additionally, we provided a deep and comprehensive comparison between 10 popular feature descriptions and shed some light on the reasons why some of them outperform the rest. Our formulation using alternating optimization was tested on the tasks of image alignment and landmark localization. Our results showed that holistic AAMs with dense HOG and SIFT features achieve robust and accurate performance and manage to outperform discriminative Deformable Models that are trained on much more visual data. Moreover, in Chapter 5, we proposed a powerful part-based generative Deformable Model, referred to as APS, that combines the main ideas behind PS and AAMs. We experimentally proved that modeling the part-based appearance of a deformable object with a GMRF structure is more beneficial than readily applying a PCA model. This is justified

by the fact that PCA assumes correlations between all variables, whereas the GMRF allows the selection of meaningful correlations between specific parts of an object. Moreover, APS utilize a spring-like deformation prior term that makes them robust to bad initializations. We also presented a variant of the Gauss-Newton optimization with fixed Jacobian and Hessian to fit the model, which is the fastest existing algorithm of its kind and its low computational complexity is independent of the employed graph structure for the GMRF. Our experimental results showed that the method is very robust to bad initializations. Finally, its part-based nature makes it suitable for various deformable object classes with complex articulations.

In Part II, we took advantage of the properties of the generative Deformable Models presented in Part I and combined them with powerful discriminative Deformable Models to achieve state-of-the-art results in two different tasks. In Chapter 6 we proposed a novel formulation for the task of semi-automatic annotation of large visual databases. Taking advantage of the qualities of feature-based holistic AAMs shown in Chapter 4, the proposed framework iteratively trains a generative and a discriminative holistic AAM ending up very accurate landmark annotations. The only requirements of the method are a statistical shape model of the deformable object and the true positive bounding boxes of the object within the images. Our extensive experimental results proved that the semi-automatically acquired annotations have comparable accuracy to manual annotations. The proposed technique is the first one that demonstrates such promising results on the task of automatic training of Deformable Models and can easily be applied on various deformable object classes. Additionally, in Chapter 7 we proposed ACR, a novel methodology that achieves state-of-the-art performance on the task of landmark localization. The method combines the descent directions of cascaded regression and Gauss-Newton optimization. This combination allows ACR to demonstrate robustness to challenging initializations and accuracy with respect to fine details. We report state-of-the-art performance using the most recent benchmark challenge, comparing against powerful methodologies some of which are provided by industrial companies and are trained on much larger training datasets.

8.1 Future Work

The work proposed in this thesis can be further extended in various manners. Specifically, one of the biggest limitations of Deformable Models is that they are mostly applied and test on the object of human face, due to the numerous annotated publicly available databases. However, the next step is to develop generative Deformable Models for both articulated and non-articulated objects that achieve state-of-the-art performance without requiring a huge

amount of training data. There exist very limited generative models that are suitable and have been extensively tested on articulated objects [156, 10] (please refer to Chapter 5). This is because:

- Articulated objects often have more complex texture space than non-articulated objects (e.g., the variations of the human body texture space are larger than the variations of the human face, due to clothes, severe self occlusions etc.). Hence, linear component analysis techniques may fail to properly describe these textures statistically.
- The majority of the employed generative component analysis techniques are based on holistic low-rank assumptions (such as PCA and its linear and non-linear variations). These methods are not able to capture the relationship between parts of articulated objects both in the appearance space, as well as in the deformable shape space.

These two challenges can be addressed in the following ways:

- Apply recently developed deep methodologies for feature extraction, which can be trained in an unsupervised manner [24] or off-the-shelf trained DCNNs [139].
- Investigate the development of statistical component analysis techniques that combine low-rank and hierarchical/structured principles (e.g., introduce a part constraint PCA in order to encapsulate the dependencies between the object parts in terms of both texture and shape).

Additionally, there is plenty of room to propose novel methodologies for training Deformable Models with limited or even no human supervision and explore solutions towards the online incremental update of these models with new training samples (lifelong learning). This refers to the task of constantly updating generic Deformable Models with images coming from the web and gradually turning them into instant specific models. Chapter 6 provides a very solid proof of concept that supports the research towards this direction. Everyday thousands of images are uploaded on the Internet. Hence, the methodologies should be able to constantly incorporate new knowledge in an incremental fashion. To this end, it should be investigated how various component analysis techniques (especially the ones focused on articulated objects) could be reformulated so as to allow incremental learning. Moreover, in order to learn Deformable Models of a specific object instance, for example a person-specific body Deformable Model, one can safely rely on the fact that these image samples are highly correlated. Hence, it

is reasonable to assume that the object's appearance will reside in a low-rank subspace and incorporate extra low-rank constraints to powerful generative frameworks.

Finally, it is really important for the research community to continue developing challenging benchmarks and high-quality open-source implementations of the various approaches. Given the strong and increasing impact of industrial research due to the unlimited resources, open source knowledge is the only way in which academic research can keep leading the constantly growing advances.

Appendix A

Appendices

A.1 Precision matrix form of GMRF

Herein we provide a proof for the precision matrix formulations of Eqs. 5.12 and 5.14. For this purpose, let us define an undirected graph $G = (V, E)$ of n vertexes, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertexes and there is an edge $(v_i, v_j) \in E$ for each pair of connected vertexes.

A.1.1 Properties

The following properties can be easily proved.

Property 1: If $\begin{cases} f(i,j) \neq 0, \forall i,j : (v_i, v_j) \in E \\ f(i,j) = 0, \forall i,j : (v_i, v_j) \notin E \end{cases}$ then $\sum_{\forall i,j:(v_i,v_j)\in E} f(i,j) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f(i,j).$

Property 2: $\sum_{\forall i,j:(v_i,v_j)\in E} f(i) + f(j) = \sum_{i=1}^{n} c_i f(i)$, where $c_i = \sum_{\forall j:(v_i,v_j)\in E} 1 + \sum_{\forall j:(v_j,v_i)\in E} 1$ denotes the number of neighbours of vertex v_i.

A.1.2 Proof 1

Herein we provide a proof for the precision matrix formulation of Eq. 5.12. Assume that we have a set of vectors of length k that correspond to each vertex, *i.e.*, $\mathbf{x}_i = [x_1^i, x_2^i, \ldots, x_k^i], \forall i : v_i \in V$. Moreover, let us assume a set of symmetrix pairwise precision matrices for each edge of the graph of size $2k \times 2k$, that have the form

$$\mathbf{Q}^{ij} = \begin{bmatrix} \mathbf{Q}_i & \mathbf{Q}_{ij} \\ \mathbf{Q}_{ij}^\mathsf{T} & \mathbf{Q}_j \end{bmatrix}, \forall i,j : (v_i, v_j) \in E \tag{A.1}$$

We aim to find the structure of $\mathbf{Q}$, so that

$$\sum_{\forall i,j:(v_i,v_j)\in E} \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix}^\mathsf{T} \mathbf{Q}^{ij} \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix} = \mathbf{x}^\mathsf{T}\mathbf{Q}\mathbf{x} \tag{A.2}$$

where $\mathbf{x} = \left[\mathbf{x}_1^\mathsf{T}, \mathbf{x}_2^\mathsf{T}, \dots, \mathbf{x}_n^\mathsf{T}\right]^\mathsf{T}$. By separating the $kn \times kn$ matrix $\mathbf{Q}$ in blocks of size $k \times k$ as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \cdots & \mathbf{K}_{1n} \\ \mathbf{K}_{21} & \mathbf{K}_{22} & \cdots & \mathbf{K}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{K}_{n1} & \mathbf{K}_{n2} & \cdots & \mathbf{K}_{nn} \end{bmatrix} \tag{A.3}$$

the second part of Eq. A.2 can be written as

$$\begin{aligned} \mathbf{x}^\mathsf{T}\mathbf{Q}\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}^\mathsf{T} \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \cdots & \mathbf{K}_{1n} \\ \mathbf{K}_{21} & \mathbf{K}_{22} & \cdots & \mathbf{K}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{K}_{n1} & \mathbf{K}_{n2} & \cdots & \mathbf{K}_{nn} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} = \\ = \sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T}\mathbf{K}_{ii}\mathbf{x}_i + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \left(\mathbf{x}_i^\mathsf{T}\mathbf{K}_{ij}\mathbf{x}_j + \mathbf{x}_j^\mathsf{T}\mathbf{K}_{ji}\mathbf{x}_i\right) \end{aligned} \tag{A.4}$$

Given the properties of Sec. A.1.1, the first part of Eq. A.2 can be written as

$$\begin{aligned} \sum_{\forall i,j:(v_i,v_j)\in E} \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix}^\mathsf{T} \mathbf{Q}^{ij} \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix} = \sum_{\forall i,j:(v_i,v_j)\in E} \mathbf{x}_i^\mathsf{T}\mathbf{Q}_i\mathbf{x}_i + \mathbf{x}_j^\mathsf{T}\mathbf{Q}_j\mathbf{x}_j + 2\mathbf{x}_i^\mathsf{T}\mathbf{Q}_{ij}\mathbf{x}_j = \\ = \sum_{i=1}^{n} c_i\mathbf{x}_i^\mathsf{T}\mathbf{Q}_i\mathbf{x}_i + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} 2\mathbf{x}_i^\mathsf{T}\mathbf{Q}_{ij}\mathbf{x}_j \end{aligned} \tag{A.5}$$

By equalizing Eqs. A.4 and A.5 we get

$$\begin{aligned} &\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T}\mathbf{K}_{ii}\mathbf{x}_i + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \left(\mathbf{x}_i^\mathsf{T}\mathbf{K}_{ij}\mathbf{x}_j + \mathbf{x}_j^\mathsf{T}\mathbf{K}_{ji}\mathbf{x}_i\right) = \sum_{i=1}^{n} c_i\mathbf{x}_i^\mathsf{T}\mathbf{Q}_i\mathbf{x}_i + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} 2\mathbf{x}_i^\mathsf{T}\mathbf{Q}_{ij}\mathbf{x}_j \Rightarrow \\ &\Rightarrow \begin{cases} \mathbf{x}_i^\mathsf{T}\mathbf{K}_{ii}\mathbf{x}_i = c_i\mathbf{x}_i^\mathsf{T}\mathbf{Q}_i\mathbf{x}_i \\ \mathbf{x}_i^\mathsf{T}\mathbf{K}_{ij}\mathbf{x}_j + \mathbf{x}_j^\mathsf{T}\mathbf{K}_{ji}\mathbf{x}_i = 2\mathbf{x}_i^\mathsf{T}\mathbf{Q}_{ij}\mathbf{x}_j \end{cases} \Rightarrow \begin{cases} \mathbf{x}_i^\mathsf{T}\mathbf{K}_{ii}\mathbf{x}_i = \mathbf{x}_i^\mathsf{T}(c_i\mathbf{Q}_i)\mathbf{x}_i \\ \mathbf{x}_i^\mathsf{T}\mathbf{K}_{ij}\mathbf{x}_j + \left(\mathbf{x}_i^\mathsf{T}\mathbf{K}_{ji}^\mathsf{T}\mathbf{x}_j\right)^\mathsf{T} = \mathbf{x}_i^\mathsf{T}(2\mathbf{Q}_{ij})\mathbf{x}_j \end{cases} \Rightarrow \\ &\Rightarrow \begin{cases} \mathbf{K}_{ii} = c_i\mathbf{Q}_i \\ \mathbf{K}_{ij} = \mathbf{K}_{ji}^\mathsf{T} = \mathbf{Q}_{ij} \end{cases} \end{aligned} \tag{A.6}$$

Consequently, by defining $\mathcal{G}_i = \{(i-1)k+1, (i-1)k+2, \dots, ik\}$ to be a set of sampling indices and given Eq. A.1, in order for Eq. A.2 to be true, the structure of $\mathbf{Q}$ is

$$\mathbf{Q} = \begin{cases} \sum\limits_{\forall j:(v_i,v_j)\in E} \mathbf{Q}_{ij}(\mathcal{G}_1,\mathcal{G}_1) + \sum\limits_{\forall j:(v_j,v_i)\in E} \mathbf{Q}_{ji}(\mathcal{G}_2,\mathcal{G}_2), \ \forall v_i \in V, & \text{at } (\mathcal{G}_i,\mathcal{G}_i) \\ \mathbf{Q}_{ij}(\mathcal{G}_1,\mathcal{G}_2), \ \forall i,j:(v_i,v_j)\in E, & \text{at } (\mathcal{G}_i,\mathcal{G}_j) \text{ and } (\mathcal{G}_j,\mathcal{G}_i) \\ 0, & \text{elsewhere} \end{cases} \tag{A.7}$$

A.1.3 Proof 2

Similar to the previous case, herein we provide a proof for the precision matrix formulation of Eq. 5.14. Again, assume that we have a set of vectors of length k that correspond to each vertex, *i.e.*, $\mathbf{x}_i = [x_1^i, x_2^i, \dots, x_k^i], \ \forall i : v_i \in V$. We aim to find the structure of $\mathbf{Q}$, so that

$$\sum_{\forall i,j:(v_i,v_j)\in E} [\mathbf{x}_i - \mathbf{x}_j]^\mathsf{T} \mathbf{Q}^{ij} [\mathbf{x}_i - \mathbf{x}_j] = \mathbf{x}^\mathsf{T}\mathbf{Q}\mathbf{x} \tag{A.8}$$

where $\mathbf{Q}^{ij}$ is the $k \times k$ precision matrix corresponding to $\mathbf{x}_i - \mathbf{x}_j$ and $\mathbf{x} = [\mathbf{x}_1^\mathsf{T}, \mathbf{x}_2^\mathsf{T}, \dots, \mathbf{x}_n^\mathsf{T}]^\mathsf{T}$.

By separating the $kn \times kn$ matrix $\mathbf{Q}$ in blocks of size $k \times k$ as shown in Eq. A.3, the second part of Eq. A.8 has the same form as shown in Eq. A.4. Given the properties of Sec. A.1.1, the first part of Eq. A.8 can be written as

$$\begin{aligned} &\sum_{\forall i,j:(v_i,v_j)\in E} [\mathbf{x}_i - \mathbf{x}_j]^\mathsf{T} \mathbf{Q}^{ij} [\mathbf{x}_i - \mathbf{x}_j] = \sum_{\forall i,j:(v_i,v_j)\in E} \left[\mathbf{x}_i^\mathsf{T}\mathbf{Q}^{ij} - \mathbf{x}_j^\mathsf{T}\mathbf{Q}^{ij}\right] [\mathbf{x}_i - \mathbf{x}_j] = \\ &= \sum_{\forall i,j:(v_i,v_j)\in E} \mathbf{x}_i^\mathsf{T}\mathbf{Q}^{ij}\mathbf{x}_i + \mathbf{x}_j^\mathsf{T}\mathbf{Q}^{ij}\mathbf{x}_j - \mathbf{x}_i^\mathsf{T}\mathbf{Q}^{ij}\mathbf{x}_j - (\mathbf{x}_i^\mathsf{T}(\mathbf{Q}^{ij})^\mathsf{T}\mathbf{x}_j)^\mathsf{T} = \\ &= \sum_{\forall i,j:(v_i,v_j)\in E} \mathbf{x}_i^\mathsf{T}\mathbf{Q}^{ij}\mathbf{x}_i + \mathbf{x}_j^\mathsf{T}\mathbf{Q}^{ij}\mathbf{x}_j - 2\mathbf{x}_i^\mathsf{T}\mathbf{Q}^{ij}\mathbf{x}_j = \sum_{i=1}^{n} c_i \mathbf{x}_i^\mathsf{T}\mathbf{Q}^{ij}\mathbf{x}_i - \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} 2\mathbf{x}_i^\mathsf{T}\mathbf{Q}^{ij}\mathbf{x}_j \end{aligned} \tag{A.9}$$

By equalizing Eqs. A.4 and A.9 we get

$$\begin{aligned} &\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T}\mathbf{K}_{ii}\mathbf{x}_i + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \left(\mathbf{x}_i^\mathsf{T}\mathbf{K}_{ij}\mathbf{x}_j + \mathbf{x}_j^\mathsf{T}\mathbf{K}_{ji}\mathbf{x}_i\right) = \sum_{i=1}^{n} c_i \mathbf{x}_i^\mathsf{T}\mathbf{Q}^{ij}\mathbf{x}_i - \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} 2\mathbf{x}_i^\mathsf{T}\mathbf{Q}^{ij}\mathbf{x}_j \Rightarrow \\ &\Rightarrow \begin{cases} \mathbf{x}_i^\mathsf{T}\mathbf{K}_{ii}\mathbf{x}_i = c_i\mathbf{x}_i^\mathsf{T}\mathbf{Q}^{ij}\mathbf{x}_i \\ \mathbf{x}_i^\mathsf{T}\mathbf{K}_{ij}\mathbf{x}_j + \mathbf{x}_j^\mathsf{T}\mathbf{K}_{ji}\mathbf{x}_i = -2\mathbf{x}_i^\mathsf{T}\mathbf{Q}^{ij}\mathbf{x}_j \end{cases} \Rightarrow \begin{cases} \mathbf{x}_i^\mathsf{T}\mathbf{K}_{ii}\mathbf{x}_i = \mathbf{x}_i^\mathsf{T}(c_i\mathbf{Q}^{ij})\mathbf{x}_i \\ \mathbf{x}_i^\mathsf{T}\mathbf{K}_{ij}\mathbf{x}_j + \left(\mathbf{x}_i^\mathsf{T}\mathbf{K}_{ji}^\mathsf{T}\mathbf{x}_j\right)^\mathsf{T} = \mathbf{x}_i^\mathsf{T}(-2\mathbf{Q}^{ij})\mathbf{x}_j \end{cases} \\ &\Rightarrow \begin{cases} \mathbf{K}_{ii} = c_i\mathbf{Q}^{ij} \\ \mathbf{K}_{ij} = \mathbf{K}_{ji}^\mathsf{T} = -\mathbf{Q}_{ij} \end{cases} \end{aligned} \tag{A.10}$$

Consequently, by defining $\mathcal{G}_i = \{(i-1)k+1, (i-1)k+2, \ldots, ik\}$ to be a set of sampling indices, in order for Eq. A.8 to be true, the structure of $\mathbf{Q}$ is

$$\mathbf{Q} = \begin{cases} \sum\limits_{\forall j:(v_i,v_j)\in E} \mathbf{Q}^{ij} + \sum\limits_{\forall j:(v_j,v_i)\in E} \mathbf{Q}^{ji}, \ \forall v_i \in V, & \text{at } (\mathcal{G}_i, \mathcal{G}_i) \\ -\mathbf{Q}^{ij}, \ \forall i,j : (v_i,v_j) \in E, & \text{at } (\mathcal{G}_i, \mathcal{G}_j) \text{ and } (\mathcal{G}_j, \mathcal{G}_i) \\ 0, & \text{elsewhere} \end{cases} \tag{A.11}$$

A.2 Forward-Additive Optimization of Active Pictorial Structures

Herein, we show the forward-additive Gauss-Newton optimization for Active Pictorial Structures (APS) of Chapter 5 and prove that it is much slower than the inverse one. The general cost function to be optimized is

$$\underset{\mathbf{p}}{\operatorname{argmin}} \|\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})) - \bar{\mathbf{a}}\|^2_{\mathbf{Q}^a} + \|\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}) - \bar{\mathbf{s}}\|^2_{\mathbf{Q}^d} \tag{A.12}$$

By using an additive iterative update of the parameters as

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p} \tag{A.13}$$

and having an initial estimate of $\mathbf{p}$, the cost function of Eq. 5.25 is expressed as minimizing

$$\underset{\Delta\mathbf{p}}{\operatorname{argmin}} \|\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}+\Delta\mathbf{p})) - \bar{\mathbf{a}}\|^2_{\mathbf{Q}^a} + \|\mathcal{S}(\mathbf{0}, \mathbf{p}+\Delta\mathbf{p})\|^2_{\mathbf{Q}^s} \tag{A.14}$$

with respect to $\Delta\mathbf{p}$. In order to find the solution we need to linearize around $\mathbf{p}$, thus using first order Taylor series expansion at $\mathbf{p}+\Delta\mathbf{p} = \mathbf{p} \Rightarrow \Delta\mathbf{p} = \mathbf{0}$ as

$$\begin{cases} \mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}+\Delta\mathbf{p})) \approx \mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})) + \mathbf{J}_{\mathcal{A}}|_{\mathbf{p}=\mathbf{p}} \Delta\mathbf{p} \\ \mathcal{S}(\mathbf{0}, \mathbf{p}+\Delta\mathbf{p}) \approx \mathcal{S}(\mathbf{0}, \mathbf{p}) + \mathbf{J}_{\mathcal{S}}|_{\mathbf{p}=\mathbf{p}} \Delta\mathbf{p} \end{cases} \tag{A.15}$$

where $\mathbf{J}_{\mathcal{S}}|_{\mathbf{p}=\mathbf{p}} = \mathbf{J}_{\mathcal{S}}$ is the $2n \times n_s$ shape Jacobian

$$\mathbf{J}_{\mathcal{S}} = \frac{\partial \mathcal{S}}{\partial \mathbf{p}} = \mathbf{U} \tag{A.16}$$

and $\mathbf{J}_{\mathcal{A}}|_{\mathbf{p}=\mathbf{p}} = \mathbf{J}_{\mathcal{A}}$ is the $mn \times n_s$ appearance Jacobian

$$\mathbf{J}_{\mathcal{A}} = \nabla_{\mathcal{A}} \frac{\partial \mathcal{S}}{\partial \mathbf{p}} = \nabla_{\mathcal{A}} \mathbf{U} = \begin{bmatrix} \nabla\mathcal{F}(\mathcal{S}_1(\bar{\mathbf{s}}, \mathbf{p}))\mathbf{U}_{1,2} \\ \nabla\mathcal{F}(\mathcal{S}_2(\bar{\mathbf{s}}, \mathbf{p}))\mathbf{U}_{3,4} \\ \vdots \\ \nabla\mathcal{F}(\mathcal{S}_n(\bar{\mathbf{s}}, \mathbf{p}))\mathbf{U}_{2i-1,2i} \end{bmatrix} \tag{A.17}$$

$\mathbf{H}$	$\mathbf{H}^{-1}$	$\mathbf{J}_{\mathcal{A}}$	$\mathbf{J}_{\mathcal{A}}{}^{\mathsf{T}}\mathbf{\Sigma}^{a}$	$\mathcal{S}(\bar{\mathbf{s}},\mathbf{p})$	$\mathbf{H}_{\mathcal{S}}\mathbf{p}$
$\mathcal{O}(m^2n^2n_s + mnn_s^2)$	$\mathcal{O}({n_s}^3)$	$\mathcal{O}(mnn_s)$	$\mathcal{O}(m^2n^2n_s)$	$\mathcal{O}(2nn_s)$	$\mathcal{O}({n_s}^2)$

Table A.1: The computational costs of all terms during the computation of the parameters increment. n is the number of landmark points, m is the length of the features' vector extracted from a patch and n_s is the number of shape parameters.

where $\mathbf{U}_{2i-1,2i}$ denotes the $2i-1$ and $2i$ row vectors of the basis $\mathbf{U}$. Note that we make an abuse of notation with $\nabla\mathcal{F}(\mathcal{S}_1(\bar{\mathbf{s}},\mathbf{p}))$ because $\mathcal{F}(\mathcal{S}_i(\bar{\mathbf{s}},\mathbf{p}))$ is a vector. However, it represents the gradient of a patch around landmark i and it has size $m\times 2$. By substituting we get

$$\begin{aligned}
&\underset{\Delta\mathbf{p}}{\operatorname{argmin}} \|\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}},\mathbf{p})) + \mathbf{J}_{\mathcal{A}}\Delta\mathbf{p} - \bar{\mathbf{a}}\|^2_{\mathbf{Q}^a} + \|\mathcal{S}(\mathbf{0},\mathbf{p}) + \mathbf{J}_{\mathcal{S}}\Delta\mathbf{p}\|^2_{\mathbf{Q}^s} = \\
&= \underset{\Delta\mathbf{p}}{\operatorname{argmin}} \Big([\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}},\mathbf{p})) + \mathbf{J}_{\mathcal{A}}\Delta\mathbf{p} - \bar{\mathbf{a}}]^{\mathsf{T}}\,\mathbf{Q}^a\,[\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}},\mathbf{p})) + \mathbf{J}_{\mathcal{A}}\Delta\mathbf{p} - \bar{\mathbf{a}}] + \\
&\qquad + [\mathcal{S}(\mathbf{0},\mathbf{p}) + \mathbf{J}_{\mathcal{S}}\Delta\mathbf{p}]^{\mathsf{T}}\,\mathbf{Q}^s\,[\mathcal{S}(\mathbf{0},\mathbf{p}) + \mathbf{J}_{\mathcal{S}}\Delta\mathbf{p}] \Big)
\end{aligned} \tag{A.18}$$

Taking the partial derivative with respect to $\Delta\mathbf{p}$ and solving for equality with $\mathbf{0}$ we get

$$\begin{aligned}
&2\mathbf{J}_{\mathcal{A}}{}^{\mathsf{T}}\mathbf{Q}^a\,(\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}},\mathbf{p})) + \mathbf{J}_{\mathcal{A}}\Delta\mathbf{p} - \bar{\mathbf{a}}) + 2\mathbf{J}_{\mathcal{S}}{}^{\mathsf{T}}\mathbf{Q}^s\,(\mathcal{S}(\mathbf{0},\mathbf{p}) + \mathbf{J}_{\mathcal{S}}\Delta\mathbf{p}) = \mathbf{0} \Rightarrow \\
&\Rightarrow 2\mathbf{J}_{\mathcal{A}}{}^{\mathsf{T}}\mathbf{Q}^a\,(\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}},\mathbf{p})) - \bar{\mathbf{a}}) + 2\mathbf{J}_{\mathcal{A}}{}^{\mathsf{T}}\mathbf{Q}^a\mathbf{J}_{\mathcal{A}}\Delta\mathbf{p} + 2\mathbf{J}_{\mathcal{S}}{}^{\mathsf{T}}\mathbf{Q}^s\mathcal{S}(\mathbf{0},\mathbf{p}) + 2\mathbf{J}_{\mathcal{S}}{}^{\mathsf{T}}\mathbf{Q}^s\mathbf{J}_{\mathcal{S}}\Delta\mathbf{p} = \mathbf{0} \Rightarrow \\
&\Rightarrow \Delta\mathbf{p} = -[\mathbf{J}_{\mathcal{A}}{}^{\mathsf{T}}\mathbf{Q}^a\mathbf{J}_{\mathcal{A}} + \mathbf{J}_{\mathcal{S}}{}^{\mathsf{T}}\mathbf{Q}^s\mathbf{J}_{\mathcal{S}}]^{-1}[\mathbf{J}_{\mathcal{A}}{}^{\mathsf{T}}\mathbf{Q}^a\,(\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}},\mathbf{p})) - \bar{\mathbf{a}}) + \mathbf{J}_{\mathcal{S}}{}^{\mathsf{T}}\mathbf{Q}^s\mathcal{S}(\mathbf{0},\mathbf{p})]
\end{aligned} \tag{A.19}$$

Thus by denoting as

$$\left.\begin{aligned}
\mathbf{H}_{\mathcal{A}} &= \mathbf{J}_{\mathcal{A}}{}^{\mathsf{T}}\mathbf{Q}^a\mathbf{J}_{\mathcal{A}} \\
\mathbf{H}_{\mathcal{S}} &= \mathbf{J}_{\mathcal{S}}{}^{\mathsf{T}}\mathbf{Q}^s\mathbf{J}_{\mathcal{S}} = \mathbf{U}^{\mathsf{T}}\mathbf{Q}^s\mathbf{U}
\end{aligned}\right\} \Rightarrow \mathbf{H} = \mathbf{H}_{\mathcal{A}} + \mathbf{H}_{\mathcal{S}} \tag{A.20}$$

the combined $n_s\times n_s$ Hessian matrix and getting into account that $\mathbf{J}_{\mathcal{S}}{}^{\mathsf{T}}\mathbf{Q}^s\mathcal{S}(\mathbf{0},\mathbf{p}) = \mathbf{U}^{\mathsf{T}}\mathbf{Q}^s\mathbf{U}\mathbf{p} = \mathbf{H}_{\mathcal{S}}\mathbf{p}$ then the parameters increment is given by

$$\Delta\mathbf{p} = -\mathbf{H}^{-1}[\mathbf{J}_{\mathcal{A}}{}^{\mathsf{T}}\mathbf{Q}^a\,(\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}},\mathbf{p})) - \bar{\mathbf{a}}) + \mathbf{H}_{\mathcal{S}}\mathbf{p}] \tag{A.21}$$

In Eq. A.21, $\mathbf{H}_{\mathcal{S}}$ can be precomputed but $\mathbf{J}_{\mathcal{A}}$ and $\mathbf{H}^{-1}$ need to be computed at each iteration. Consequently, based on the costs of Tab. A.1, the total computational cost is $\mathcal{O}(m^2n^2n_s + mnn_s + {n_s}^3)$, which is much slower than the cost of the weighted inverse compositional algorithm with fixed Jacobian and Hessian ($\mathcal{O}(mn)$).

List of Figures

List of Tables

Bibliography

[1] Joan Alabort-i-Medina, Epameinondas Antonakos, James Booth, Patrick Snape, and Stefanos Zafeiriou. Menpo: A comprehensive platform for parametric image alignment and visual deformable models. In *Proceedings of the ACM International Conference on Multimedia (ACM'MM)*, pages 679–682. ACM, 2014. 10, 11, 12, 67, 68, 90, 147, 148

[2] Joan Alabort-i-Medina, Epameinondas Antonakos, James Booth, Patrick Snape, and Stefanos Zafeiriou. The menpo project. In *ACM SIGMM Records*, volume 8. ACM, 2016. 10, 11, 12, 147

[3] Joan Alabort-i-Medina and Stefanos Zafeiriou. Bayesian active appearance models. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 3438–3445, 2014. 11, 12, 18, 21, 85, 89, 116

[4] Joan Alabort-i-Medina and Stefanos Zafeiriou. Unifying holistic and parts-based deformable model fitting. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2015. 18, 116

[5] Joan Alabort-i-Medina and Stefanos Zafeiriou. A unified framework for compositional fitting of active appearance models. *International Journal of Computer Vision (IJCV)*, pages 1–39, 2016. 2, 3, 9, 18, 103, 116

[6] Brian Amberg, Andrew Blake, and Thomas Vetter. On compositional image alignment, with an application to active appearance models. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1714–1721. IEEE, 2009. 18

[7] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1014–1021. IEEE, 2009. 8, 21, 74

[8] Epameinondas Antonakos, Joan Alabort-i-Medina, Georgios Tzimiropoulos, and Stefanos Zafeiriou. Hog active appearance models. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 224–228. IEEE, 2014. 2, 3, 6, 8, 9, 11, 12, 21, 90, 103, 116

[9] Epameinondas Antonakos, Joan Alabort-i-Medina, Georgios Tzimiropoulos, and Stefanos Zafeiriou. Feature-based lucas-kanade and active appearance models. *IEEE Transactions on Image Processing (TIP)*, 24(9):2617–2632, September 2015. 2, 3, 6, 8, 9, 11, 12, 13, 21, 90, 103, 116

[10] Epameinondas Antonakos, Joan Alabort-i-Medina, and Stefanos Zafeiriou. Active pictorial structures. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 5435–5444, Boston, MA, USA, June 2015. IEEE. 2, 3, 12, 139

[11] Epameinondas Antonakos, Patrick Snape, George Trigeorgis, and Stefanos Zafeiriou. Adaptive cascaded regression. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016. 2, 3

[12] Epameinondas Antonakos and Stefanos Zafeiriou. Automatic construction of deformable models in-the-wild. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1813–1820, Columbus, OH, USA, June 2014. IEEE. 19

[13] Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Robust discriminative response map fitting with constrained local models. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 3444–3451, 2013. 22, 38, 66, 67, 68, 69, 110, 148, 149

[14] Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Incremental face alignment in the wild. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1859–1866, 2014. 2, 3, 9, 11, 12, 20, 21, 115, 128, 130, 134

[15] Akshay Asthana, Stefanos Zafeiriou, Georgios Tzimiropoulos, Shiyang Cheng, and Maja Pantic. From pixels to response maps: Discriminative image filtering for face alignment in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(6):1312–1320, 2015. 11, 12

[16] Simon Baker, Ralph Gross, Iain Matthews, and Takahiro Ishikawa. Lucas-kanade 20 years on: A unifying framework: Part 2. Technical Report CMU-RI-TR-03-01, Robotics Institute, Pittsburgh, PA, February 2003. 8, 13, 18, 21, 38, 46

[17] Simon Baker and Iain Matthews. Equivalence and efficiency of image alignment algorithms. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2001. 18, 47

[18] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision (IJCV)*, 56(3):221–255, 2004. 2, 3, 8, 13, 18, 21, 26, 37, 38, 45, 46, 47, 55, 56, 82, 116

[19] Simon Baker, Iain Matthews, and Jeff Schneider. Automatic construction of active appearance models as an image coding problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(10):1380–1384, 2004. 6, 22, 23, 98, 99

[20] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. Continuous conditional neural fields for structured regression. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 593–608. Springer, 2014. 22

[21] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(8):1670–1687, 2015. 4

[22] Peter N Belhumeur, David W Jacobs, David J Kriegman, and Neeraj Kumar. Localizing parts of faces using a consensus of exemplars. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2011. 5, 17, 31, 38, 58, 86, 97, 128

[23] Peter N Belhumeur, David W Jacobs, David J Kriegman, and Neeraj Kumar. Localizing parts of faces using a consensus of exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(12):2930–2940, 2013. 22, 130

[24] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009. 139

[25] Michael J Black and Allan D Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision (IJCV)*, 26(1):63–84, 1998. 18

[26] 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*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999. 4, 5, 13

[27] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 25(9):1063–1074, 2003. 4

[28] Vishnu Naresh Boddeti, Takeo Kanade, and BVK Kumar. Correlation filters for object alignment. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2291–2298, 2013. 91

[29] Fred L Bookstein et al. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 11(6):567–585, 1989. 21

[30] James Booth, Epameinondas Antonakos, Stylianos Ploumpis, George Trigeorgis, Yannis Panagakis, and Stefanos Zafeiriou. 3d face morphable models "in-the-wild". In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. 4

[31] James Booth, Anastasios Roussos, Allan Ponniah, David Dunaway, and Stefanos Zafeiriou. Large scale 3d morphable models. *International Journal of Computer Vision (IJCV)*, 2017. 4, 5

[32] James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway. A 3d morphable model learnt from 10,000 faces. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 5543–5552, 2016. 4, 5, 107

[33] Gary Bradski et al. The opencv library. *Doctor Dobbs Journal*, 25(11):120–126, 2000. 13

[34] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(8):1872–1886, 2013. 70

[35] Xavier P Burgos-Artizzu, Pietro Perona, and Piotr Dollár. Robust face landmark estimation under occlusion. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 1513–1520. IEEE, 2013. 20, 21, 115, 125

[36] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision (IJCV)*, 107(2):177–190, 2014. 20, 21, 99, 105, 115

[37] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016. 5

[38] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 484–498, 1998. 18, 21, 116

[39] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(6):681–685, 2001. 3, 8, 18, 20, 21, 26, 37, 73, 116

[40] Timothy F Cootes, Stephen Marsland, Carole J Twining, Kate Smith, and Christopher J Taylor. Groupwise diffeomorphic non-rigid registration for automatic model building. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2004. 22

[41] Timothy F Cootes and Christopher J Taylor. On representing edge structure for model matching. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2001. 19, 38, 40

[42] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer Vision and Image Understanding (CVIU)*, 61(1):38–59, 1995. 12, 18, 22, 26, 116

[43] Timothy F Cootes, Cristopher J Taylor, et al. Statistical models of appearance for computer vision, 2004. 29, 45

[44] Mark Cox, Sridha Sridharan, Simon Lucey, and Jeffrey Cohn. Least squares congealing for unsupervised alignment of images. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008. 22

[45] David Cristinacce and Timothy F Cootes. Feature detection and tracking with constrained local models. In *British Machine Vision Computing (BMVC)*, page 3, 2006. 22

[46] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005. 3, 8, 19, 20, 21, 27, 38, 41, 105

[47] Sune Darkner, Rasmus Larsen, Mikkel B Stegmann, and BK Ersboll. Wedgelet enhanced appearance models. In *Proceedings of IEEE Computer Vision and Pattern Recognition Workshops (CVPR-W)*, 2004. 19

[48] F. De la Torre, W.-S. Chu, X. Xiong, F. Vicente, X. Ding, and J. F. Cohn. Intraface. In *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 2015. 128, 130, 134

[49] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 3, 5

[50] Piotr Dollár, Peter Welinder, and Pietro Perona. Cascaded pose regression. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1078–1085, 2010. 20, 21, 99, 105, 115

[51] Nicholas Dowson and Richard Bowden. Mutual information for lucas-kanade tracking (milk): An inverse compositional formulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(1):180–185, 2008. 18

[52] Benoit Duc, Stefan Fischer, and Josef Bigun. Face authentication with gabor information on deformable graphs. *IEEE Transactions on Image Processing (TIP)*, 8(4):504–516, 1999. 69, 70

[53] Chi Nhan Duong, Khoa Luu, Kha Gia Quach, and Tien D Bui. Beyond principal components: Deep boltzmann machines for face modeling. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 4786–4794. IEEE, 2015. 18, 19, 20

[54] Jean-Denis Durou, Maurizio Falcone, and Manuela Sagona. Numerical methods for shape-from-shading: A new survey with benchmarks. *Computer Vision and Image Understanding (CVIU)*, 109(1):22–43, 2008. 4

[55] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010. 3

[56] Haoqiang Fan and Erjin Zhou. Approaching human level facial landmark localization by deep learning. *Image and Vision Computing (IMAVIS)*, 47:27–35, 2016. 3

[57] Pedro Felzenszwalb and Daniel Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell University, 2004. 21

[58] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(9):1627–1645, 2010. 3, 13, 21, 74, 75

[59] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient matching of pictorial structures. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 66–73. IEEE, 2000. 21, 74, 84

[60] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision (IJCV)*, 61(1):55–79, 2005. 8, 21, 74, 75, 81, 83, 87

[61] Martin A Fischler and Robert A Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on computers*, 22(1):67–92, 1973. 8, 21, 74

[62] Brendan J Frey, M Jojic, and Anitha Kannan. Learning appearance and transparency manifolds of occluded objects in layers. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2003. 22

[63] Xinbo Gao, Ya Su, Xuelong Li, and Dacheng Tao. Gabor texture in active appearance models. *Journal of Neurocomputing*, 72(13):3174–3181, 2009. 19, 69

[64] Yongxin Ge, Dan Yang, Jiwen Lu, Bo Li, and Xiaohong Zhang. Active appearance models using statistical characteristics of gabor based texture representation. In *Journal of Visual Communication and Image Representation*, 2013. 19

[65] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(6):643–660, 2001. 38, 55

[66] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014. 2, 5

[67] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. 26

[68] Ralph Gross, Iain Matthews, and Simon Baker. Generic vs. person specific active appearance models. *Image and Vision Computing (IMAVIS)*, 23(12):1080–1093, 2005. 6, 18, 21, 49, 73, 104

[69] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing (IMAVIS)*, 28(5):807–813, 2010. 17, 31, 66

[70] Rıza Alp Güler, George Trigeorgis, Epameinondas Antonakos, Patrick Snape, Stefanos Zafeiriou, and Iasonas Kokkinos. Densereg: Fully convolutional dense shape regression in-the-wild. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[71] Gregory D Hager and Peter N Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(10):1025–1039, 1998. 18, 82

[72] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. *arXiv preprint arXiv:1703.06870*, 2017. 5

[73] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 5

[74] Gary B Huang, Marwan Mattar, Honglak Lee, and Erik G Learned-Miller. Learning to align from scratch. In *Advances in Neural Information Processing Systems (NIPS)*, 2012. 22

[75] László A Jeni, Sergey Tulyakov, Lijun Yin, Nicu Sebe, and Jeffrey F Cohn. The first 3d face alignment in the wild (3dfaw) challenge. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 511–520. Springer, 2016. 5

[76] Huaizu Jiang and Erik Learned-Miller. Face detection with the faster r-cnn. *arXiv preprint arXiv:1606.03473*, 2016. 2

[77] Tingting Jiang, Frederic Jurie, and Cordelia Schmid. Learning shape prior models for object matching. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2009. 22, 98

[78] Nebojsa Jojic, John Winn, and Larry Zitnick. Escaping local minima through hierarchical model selection: Automatic object discovery, segmentation, and tracking in video. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2006. 22

[79] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002. 18, 26

[80] A Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems (NIPS)*, 14:841, 2002. 6

[81] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision (IJCV)*, 1(4):321–331, 1988. 18

[82] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1867–1874, 2014. 2, 3, 9, 12, 20, 21, 115, 128, 130, 134

[83] Ira Kemelmacher-Shlizerman. Internet based morphable model. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 3256–3263, 2013. 4

[84] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 4873–4882, 2016. 5

[85] Davis E King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10(Jul):1755–1758, 2009. 12, 13

[86] Davis E King. Max-margin object detection. *arXiv preprint arXiv:1502.00046*, 2015. 13

[87] Panachit Kittipanya-ngam and Timothy F Cootes. The effect of texture representations on aam performance. In *Proceedings of IEEE International Conference on Pattern Recognition (ICPR)*, 2006. 19

[88] Iasonas Kokkinos and Alan Yuille. Unsupervised learning of object deformation models. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2007. 6, 22, 98

[89] Martin Köstinger, Paul Wohlhart, Peter M Roth, and Horst Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *Proceedings of IEEE International Conference on Computer Vision Workshops (ICCV-W)*, pages 2144–2151. IEEE, 2011. 17

[90] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. Centre for Exploration Targeting, Uni. of Western Australia. http://www.csse.uwa.edu.au/$\sim$pk/research/matlabfns/. 44

[91] Peter Kovesi. Symmetry and asymmetry from local phase. In *10th Australian Joint Conference on Artificial Intelligence*, 1997. 38, 44

[92] Peter Kovesi. Image features from phase congruency. *VIDERE: Journal of computer vision research*, 1(3):1–26, 1999. 38, 44

[93] Marek Kowalski, Jacek Naruniec, and Tomasz Trzcinski. Deep alignment network: A convolutional neural network for robust face alignment. *arXiv preprint arXiv:1706.01789*, 2017. 5

[94] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012. 5

[95] Christoph H Lampert, Matthew B Blaschko, and Thomas Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(12):2129–2142, 2009. 98

[96] Jukka Lankinen and Joni-Kristian Kämäräinen. Local feature based unsupervised alignment of object class images. In *British Machine Vision Computing (BMVC)*, 2011. 22

[97] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir Bourdev, and Thomas S Huang. Interactive facial feature localization. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 679–692. Springer, 2012. 5, 17, 22, 31, 38, 58, 97, 128, 132

[98] Erik Learned-Miller, Gary B Huang, Aruni RoyChowdhury, Haoxiang Li, and Gang Hua. Labeled faces in the wild: A survey. In *Advances in Face Detection and Facial Image Analysis*, pages 189–248. Springer, 2016. 3

[99] Erik G Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(2):236–250, 2006. 22

[100] Der-Tsai Lee and Bruce J Schachter. Two algorithms for constructing a delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3):219–242, 1980. 29, 53

[101] Donghoon Lee, Hyunsin Park, and Chang D Yoo. Face alignment using cascade gaussian process regression trees. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 4204–4212. IEEE, 2015. 20, 21

[102] Tai Sing Lee. Image representation using 2d gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 18(10):959–971, 1996. 38, 44

[103] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua. A convolutional neural network cascade for face detection. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 5325–5334, 2015. 2

[104] Yan Li, Leon Gu, and Takeo Kanade. Robustly aligning a shape model and its application to car alignment of unknown pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(9):1860–1876, 2011. 91

[105] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects

in context. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 740–755, 2014. 5

[106] Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic, and William T Freeman. Sift flow: dense correspondence across different scenes. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2008. 18, 19

[107] Xiaoming Liu, Yan Tong, and Frederick W Wheeler. Simultaneous alignment and clustering for an image ensemble. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2009. 22

[108] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 5

[109] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 1150–1157, 1999. 3, 8, 19, 20, 21, 27, 38, 42, 85, 105, 126

[110] Can-Yi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. Robust and efficient subspace segmentation via least squares regression. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2012. 106

[111] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *Proceedings of International Joint Conference on Artificial Intelligence*, volume 81, pages 674–679, 1981. 37, 46

[112] Simon Lucey, Sridha Sridharan, Rajitha Navarathna, and Ahmed Bilal Ashraf. Fourier lucas-kanade algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(6):1383–1396, 2013. 13, 18, 19, 69, 70

[113] Nenad Markuš, Miroslav Frljak, Igor S Pandžić, Jörgen Ahlberg, and Robert Forchheimer. Object detection with pixel intensity comparisons organized in decision trees. *arXiv preprint arXiv:1305.4537*, 2013. 13

[114] Pedro Martins, Rui Caseiro, and Jorge Batista. Non-parametric bayesian constrained local models. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1797–1804. IEEE, 2014. 22

[115] Pedro Martins, Rui Caseiro, Joao F Henriques, and Jorge Batista. Likelihood-enhanced bayesian constrained local models. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 303–307. IEEE, 2014. 22

[116] Markus Mathias, Rodrigo Benenson, Marco Pedersoli, and Luc Van Gool. Face detection without bells and whistles. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 720–735. Springer, 2014. 13, 74

[117] Iain Matthews and Simon Baker. Active appearance models revisited. *International Journal of Computer Vision (IJCV)*, 60(2):135–164, 2004. 2, 3, 8, 9, 11, 12, 18, 21, 29, 38, 45, 48, 55, 73, 79, 82, 85, 89, 103, 104, 116, 117, 121

[118] Rémi Mégret, J Authesserre, and Yannick Berthoumieu. Bidirectional composition on lie groups for gradient-based image alignment. *IEEE Transactions on Image Processing (TIP)*, 19(9):2369–2381, 2010. 18

[119] Rajitha Navarathna, Sridha Sridharan, and Simon Lucey. Fourier active appearance models. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 1919–1926. IEEE, 2011. 18

[120] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996. 3, 38, 43

[121] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification. In *Proceedings of International Conference on Advances in Pattern Recognition (ICAPR)*, 2001. 3, 38, 43

[122] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(7):971–987, 2002. 3, 38, 43

[123] Javier Orozco, Brais Martinez, and Maja Pantic. Empirical analysis of cascade deformable models for multi-view face detection. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2013. 58, 66

[124] George Papandreou and Petros Maragos. Adaptive and constrained algorithms for inverse compositional active appearance model fitting. In *Proceedings of IEEE Computer*

Vision and Pattern Recognition (CVPR), 2008. 2, 3, 9, 18, 29, 50, 51, 55, 63, 73, 103, 116, 120, 121

[125] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 296–301. IEEE, 2009. 4, 5, 107

[126] Yigang Peng, Arvind Ganesh, John Wright, Wenli Xu, and Yi Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(11):2233–2246, 2012. 22, 23, 98, 99

[127] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1990–1998, 2015. 5

[128] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. Face alignment at 3000 fps via regressing local binary features. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1685–1692, 2014. 2, 3, 9, 20, 21, 31, 115, 125

[129] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, 2015. 2, 5

[130] Samuel Rivera and Aleix M Martinez. Learning deformable shape manifolds. *Pattern Recognition*, 45(4):1792–1801, 2012. 5, 21

[131] Havard Rue and Leonhard Held. *Gaussian Markov random fields: theory and applications*. CRC Press, 2005. 75, 77

[132] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: Database and results. *Image and Vision Computing (IMAVIS), Special Issue on Facial Landmark Localisation "In-The-Wild"*, 47:3–18, 2016. 4, 5, 7, 17, 31, 86, 97, 117, 127, 128, 129, 151

[133] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proceedings of IEEE International Conference on Computer Vision Workshops (ICCV-W), 300 Faces in-the-Wild Challenge (300-W)*, 2013. 7, 17, 31, 38, 86, 97, 117, 127, 128, 129, 151

[134] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. A semi-automatic methodology for facial landmark annotation. In *Proceedings of IEEE Computer Vision and Pattern Recognition Workshops (CVPR-W)*, 2013. 7, 17, 31, 58, 86, 97, 117

[135] Jason Saragih. Principal regression analysis. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2888. IEEE, 2011. 22

[136] Jason Saragih and Roland Goecke. Iterative error bound minimisation for aam alignment. In *Proceedings of IEEE International Conference on Pattern Recognition (ICPR)*, 2006. 99, 105

[137] Jason M Saragih, Simon Lucey, and Jeffrey F Cohn. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision (IJCV)*, 91(2):200–215, 2011. 11, 12, 22

[138] Ian M Scott, Timothy F Cootes, and Christopher J Taylor. Improving appearance model matching using local image structure. In *Information Processing in Medical Imaging*, 2003. 19

[139] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 139

[140] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(8):888–905, 2000. 107

[141] Patrick Snape and Stefanos Zafeiriou. Kernel-pca analysis of surface normals for shape-from-shading. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1059–1066, 2014. 4

[142] Mikkel B Stegmann and Rasmus Larsen. Multi-band modelling of appearance. *Image and Vision Computing (IMAVIS)*, 21(1):61–67, 2003. 19

[143] Ya Su, Dacheng Tao, Xuelong Li, and Xinbo Gao. Texture representation in aam using gabor wavelet and local binary patterns. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2009. 19

[144] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 3476–3483. IEEE, 2013. 3, 5, 21

[145] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. 5

[146] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708, 2014. 5

[147] J Rafael Tena, Fernando De la Torre, and Iain Matthews. Interactive region-based linear 3d face models. *ACM Transactions on Graphics (TOG)*, 30(4):76, 2011. 74

[148] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1799–1807, 2014. 5

[149] Yan Tong, Xiaoming Liu, Frederick W Wheeler, and Peter H Tu. Semi-supervised facial landmark annotation. *Computer Vision and Image Understanding (CVIU)*, 116(8):922–935, 2012. 22

[150] George Trigeorgis, Patrick Snape, Iasonas Kokkinos, and Stefanos Zafeiriou. Face normals "in-the-wild" using fully convolutional networks. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, July 2017. 4

[151] George Trigeorgis, Patrick Snape, Mihalis Nicolaou, Epameinondas Antonakos, and Stefanos Zafeiriou. Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016. IEEE. 2, 5, 20

[152] Georgios Tzimiropoulos. Project-out cascaded regression with an application to face alignment. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 3659–3667. IEEE, 2015. 2, 9, 20, 21, 115, 126, 127, 130, 134

[153] Georgios Tzimiropoulos, Joan Alabort-i Medina, Stefanos Zafeiriou, and Maja Pantic. Generic active appearance models revisited. In *Proceedings of Asian Conference on*

Computer Vision (ACCV), pages 650–663. Springer, 2012. 2, 3, 9, 11, 12, 18, 19, 21, 41, 73, 85, 99, 116

[154] Georgios Tzimiropoulos, Joan Alabort-i-Medina, Stefanos Zafeiriou, and Maja Pantic. Active orientation models for face alignment in-the-wild. *IEEE Transactions on Information Forensics and Security, Special Issue on Facial Biometrics in-the-wild*, 9:2024–2034, December 2014. 11, 12, 18, 21, 41, 116

[155] Georgios Tzimiropoulos and Maja Pantic. Optimization problems for fast aam fitting in-the-wild. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2013. 18, 50, 51, 58, 63, 74, 116, 120

[156] Georgios Tzimiropoulos and Maja Pantic. Gauss-newton deformable part models for face alignment in-the-wild. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2014. 2, 3, 6, 9, 11, 12, 18, 21, 74, 82, 85, 89, 90, 116, 120, 121, 139

[157] Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. Robust and efficient parametric face alignment. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2011. 18, 19, 38, 41, 55

[158] Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. Subspace learning from image gradient orientations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(12):2454–2466, 2012. 18, 22, 38, 41, 98, 99, 101, 102, 105, 150

[159] Laurens Van Der Maaten and Emile Hendriks. Capturing appearance variation in active appearance models. In *Proceedings of IEEE Computer Vision and Pattern Recognition Workshops (CVPR-W)*, pages 34–41. IEEE, 2010. 6

[160] Andrea Vedaldi and Brian Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008. 62, 126

[161] Thomas Vetter, Michael J Jones, and Tomaso Poggio. A bootstrapping algorithm for learning linear models of object classes. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, 1997. 22

[162] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–511. IEEE, 2001. 9, 13, 98

[163] Paul Viola and Michael J Jones. Robust real-time face detection. *International Journal of Computer Vision (IJCV)*, 57(2):137–154, 2004. 9, 13, 98

[164] Paul Viola, Michael J Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision (IJCV)*, 63(2):153–161, 2005. 9, 13, 98

[165] M Weber, M Welling, and P Perona. Unsupervised learning of models for recognition. *Proceedings of European Conference on Computer Vision (ECCV)*, 2000. 22

[166] John Winn and Nebojsa Jojic. Locus: Learning object classes with unsupervised segmentation. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2005. 6, 22

[167] Laurenz Wiskott, J-M Fellous, N Kuiger, and Christoph Von Der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 19(7):775–779, 1997. 69, 70

[168] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987. 18, 26

[169] Lior Wolf, Tal Hassner, and Yaniv Taigman. Descriptor based methods in the wild. In *European Conference on Computer Vision (ECCV) Workshop on Faces in Real-Life Images: Detection, Alignment, and Recognition*, 2008. 38, 43

[170] CBH Wolstenholme and Christopher J Taylor. Wavelet compression of active appearance models. In *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 1999. 19

[171] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 532–539, 2013. 2, 3, 9, 11, 12, 20, 21, 38, 66, 67, 68, 69, 83, 90, 99, 105, 115, 128, 130, 134, 148, 149

[172] Xuehan Xiong and Fernando De la Torre. Supervised descent method for solving nonlinear least squares problems in computer vision. *arXiv preprint arXiv:1405.0601*, 2014. 20, 21, 115, 116

[173] Xuehan Xiong and Fernando De la Torre. Global supervised descent method. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2664–2673, 2015. 20, 21, 115

[174] Junjie Yan, Zhen Lei, Dong Yi, and Stan Li. Learn to combine multiple hypotheses for accurate face alignment. In *Proceedings of IEEE International Conference on Computer Vision Workshops (ICCV-W)*, pages 392–396, 2013. 21, 127, 130

[175] Heng Yang and Ioannis Patras. Face parts localization using structured-output regression forests. In *Proceedings of Asian Conference on Computer Vision (ACCV)*, pages 667–679. Springer, 2013. 20, 115

[176] Heng Yang and Ioannis Patras. Sieving regression forest votes for facial feature detection in the wild. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 1936–1943, 2013. 20, 115

[177] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 5525–5533, 2016. 5

[178] Wei Yang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 3073–3082, 2016. 5

[179] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(12):2878–2890, 2013. 74

[180] Stefanos Zafeiriou, George Trigeorgis, Grigorios Chrysos, Jiankang Deng, and Jie Shen. The menpo facial landmark localisation challenge: A step towards the solution. In *Proceedings of IEEE Computer Vision and Pattern Recognition Workshops (CVPR-W)*, 2017. 5

[181] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Learning deep representation for face alignment with auxiliary attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(5):918–930, 2016. 5, 31

[182] Erjin Zhou, Haoqiang Fan, Zhimin Cao, Yuning Jiang, and Qi Yin. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *Proceedings of IEEE International Conference on Computer Vision Workshops (ICCV-W)*, pages 386–391. IEEE, 2013. 127, 129, 130

[183] Shizhan Zhu, Cheng Li, Chen Change Loy, and Xiaoou Tang. Face alignment by coarse-to-fine shape searching. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 4998–5006. IEEE, 2015. 2, 9, 20, 21, 31, 115, 125, 127, 130, 134

[184] Song Chun Zhu and Alan L Yuille. Forms: a flexible object recognition and modelling system. *International Journal of Computer Vision (IJCV)*, 20(3):187–212, 1996. 22

[185] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2879–2886, 2012. 17, 21, 31, 38, 58, 74, 75, 81, 83, 84, 86, 90, 97, 98, 110

[186] M Zeeshan Zia, Michael Stark, Bernt Schiele, and Konrad Schindler. Revisiting 3d geometric models for accurate object shape and pose. In *Proceedings of IEEE International Conference on Computer Vision Workshops (ICCV-W)*, 2011. 98

[187] Silvia Zuffi, Oren Freifeld, and Michael J Black. From pictorial structures to deformable structures. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 3546–3553. IEEE, 2012. 21, 74