Department of Computer Science

UNIVERSITY
of York

Submitted in part fulfilment for the degree of BSc.

Face fitting with robust image feature descriptors

Yuri Pieters

May 25, 2022

Supervisor: Nick Pears

Contents

List of Figures

List of Tables

Executive Summary

The aim of this computer vision project was to evaluate a technique for analysing images of things such as faces. The technique uses a statistical model of shape and appearance called an Active Appearance Model (AAM). The model was combined with densely sampled robust feature descriptors and trained and tested on sets of images collected in uncontrolled “in-the-wild” settings.

Computer vision is an important area of research in computer science. There are uses for it in many areas, from helping robots and self driving cars understand their surroundings, to human computer interfaces that can recognise their users face or react to their movement.

Many tasks in computer vision involve analysing images of an object in some way to infer information about it, like taking an image of a face and inferring where the person is looking. A common step in tasks such as this is getting the shape of the object in question by fitting a set of landmark points to it. The landmark points mark key parts of the object, which captures the shape and allows later steps to either use the shape as part of their algorithm, or to factor out the effect of the shape.

There are several approaches to fitting a set of landmarks like this. Many approaches learn from training images how to map image appearance to landmark locations directly. Others use a statistical model of shape called a deformable shape model, which they then fit to the image. An AAM is of this second type.

AAM is a relatively simple and straightforward method of deformable model fitting. It is a generative model, which means it generates instances of the thing it models. An AAM is fitted to an new image by minimising the difference between this generated image and the actual one.

Image features are points of interest in an image, from the perspective of an computer vision algorithm at least. A feature descriptor is a measurement extracted from the image at the location of the feature that summarises some of the properties of the image in the neighbourhood

at that location. A robust feature descriptor tries to not be affected by things like changes in illumination and scale. Features can be sampled densely, meaning that a descriptor is computed at every, or nearly every, pixel in the image.

A recent paper [1] introduced a fast method for fitting an AAM to an image, and tested the method by training an AAM on data collected in-the-wild. Their model was very simple, but showed very good results. In this work the goal is to extend their model by combining the model with robust image features.

The image features tested were Image Gradient Orientations (IGO), Histogram of Orientated Gradients (HOG), Scale Invariant Feature Transform (SIFT), and DAISY. These were compared against an AAM using the raw pixel intensity values.

Each model was evaluated on the task of fitting landmarks to images of human faces, with the accuracy of the models quantified by measuring the how closely the fitted shape reproduced the true landmark points.

The feature that produced the best results was HOG, followed by SIFT and DAISY. IGO showed a good step up in performance from the raw intensity values though, while being fast compared to the other methods.

This project uses faces as the subject for the models. This does come with some potential social and ethical concerns. The datasets used are popular ones in computer vision research and consist of publicly available images, which avoids legal issues.

However any face analysis software should be aware of diversity. Human faces have a huge variety, and everyone has a right to be able to use the technology that may be supported by such face analysis software. Therefore both the computer vision techniques and actual implementation of those techniques (e.g. the actual training images used) should be developed with the full diversity of humans in mind.

While the datasets used do not come with statistics of their diversity, manual inspection of images shows what appears to be a good cross section of skin colors, ages, and genders.

1 Introduction

This report details an investigation into using Active Appearance Models (AAM), a well established framework in computer vision for the task of fitting a set of landmarks to images of objects with variable shape and appearance (human faces being the classic example). State of the art techniques for image pre-processing and non-linear function optimisation, along with modern datasets of images collected in the wild, are used in the AAM framework, and the results are compared against other recent solutions to the same problem.

Computer vision is a field of computer science that deals with getting computers to “see”. The subject has a long history, dating back to the 1960s when it was thought that the problem would be essentially solved over a summer project [2]. It proved significantly more difficult of course, and it is only with the modern advances in imaging and computing technologies that progress has accelerated, both by making new things possible and increasing the demand for solutions to computer vision problems [3]. Fields where computer vision has been applied include include medical imaging, robotics, human-computer interaction, security, manufacturing, and more [4].

Fitting a set of landmarks to objects in an image is a way to describe the shape of objects. Such a set of landmarks can be seen in fig. 1.1. Landmarking like this is a good way to capture non-rigid variation in object shape. Non-rigid variation means that in different instances of the object the points may be in different positions relative to each other, as opposed to only being different by scaling, rotation, etc. This variation can be due to the non-rigid nature of the object themselves, but could also just represent variation between different objects of the same class. Human faces, one of the most common examples used, vary both non-rigidly themselves and between instances. Bones, on the other hand, also vary non-rigidly between instances, but each instance is of course rigid on its own. Bones have been used as examples for tasks involving deformable shape models, where they are useful for analysis of medical imagery [5]; however in this work we focus on human faces. This is due to the general popularity of faces for computer vision, which has resulted in many high-quality datasets being made available [6]–[9].

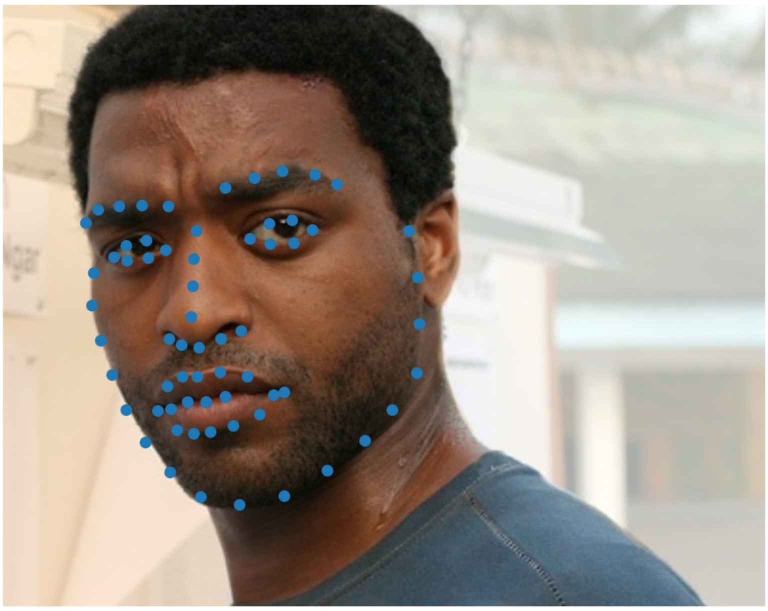

Figure 1.1: An image from the LFPW [6] training set annotated with a set of landmarks.

There have been many approaches to deformable model fitting. Examples include Constrained Local Models [8], [10], [11] and various regression methods, both cascaded [12]–[15], and not [16], [17]. These will be further discussed in sec. 2.1 in this report. One method that stands out for it's relative simplicity, while still achieving good results, is Active Appearance Models.

AAMs are generative statistical models of appearance and shape. They parametrically describe the variations in appearance and shape seen in objects, and can be fit to an unseen image by minimising the difference between it and the image generated by the model. The resulting model instance is a relatively compact description of the object that can be used in further analysis. For example, an instance of an AAM trained on human faces could be used to estimate head pose from the shape, or gender from the appearance.

1.1 Motivation, aims, and objectives

This work is in large part inspired by the work in [1], which introduced a new AAM fitting algorithm they named Fast-SIC, but has since been termed alternating inverse-compositional (AIC). They showed that, using this algorithm to fit an AAM trained on in-the-wild data, state-of-the art performance could be achieved on generic face fitting problems, even without the use of robust feature descriptors.

The aim of this work is to evaluate the performance of AAMs in a similar situation that do use robust features. A set of such features will be selected along with a sets of images collected in-the-wild, and the results will be evaluated to give insight into the power of these descriptors and AAM itself.

1.2 Report overview

The structure of the rest of this report is as follows. In sec. 2 a review of related works and relevant background is provided. In sec. 3 the experimental methodology and implementation details are given; this is followed by sec. 4 where the results of the experiments are presented. Finally, in sec. 5 the report is concluded.

2 Literature review

In this section we present a review of the literature on deformable model fitting and prior work on AAMs.

We start with an overview of solutions to the deformable model fitting problem, placing AAM into context. We then present a description of the basic formulation of an AAM, setting the stage for a review of various enhancements that have been proposed. Finally, because we compare AAMs with different image feature descriptors we review these.

2.1 Other landmarking methods

There have been many approaches taken to the problem of landmark fitting, but they can largely be divided into three categories [18]: *holistic methods*, *Constrained Local Model (CLM) methods*, and *regression based methods*. The categories are based on how the facial appearance and facial shape patterns are modelled and related. For holistic methods, the main example is AAM; the category is named because the holistic appearance is used to fit the landmarks. CLM methods train a set of independent models for each of the facial landmarks, but constrain the locations of the landmarks based on a global model of the face shape. Lastly, the regression based methods do not explicitly model the global face shape at all, instead directly relating image data (either local or global) to landmark locations.

Holistic methods

This category is largely defined by AAMs. AAMs were first explored in the work of Edwards et al. [19] in the late 90s. There was much interest at the time in interpretation by synthesis, a by which images are interpreted by synthesising a parametric version of the image.

Briefly, AAMs work by combining a model of shape and a model of appearance. The model of appearance is for the whole face, which is what makes it holistic. They generate (or synthesise) a face by generating the appearance in a reference shape and then warping it to the shape generated by the shape model.

More details of AAM are given in sec. 2.2.

Constrained Local Models

The central idea of CLM is to model, for each landmark, the likelihood that it should be placed on a certain part of the image, but to then constrain the final landmark locations to fit a model of the face shape as a whole.

CLM was first named as such in [10], but the definition used here comes from [11], where they identify the earliest paper that uses the method of CLM as [20]. In CLM, a set of simple detectors is trained for each landmark. On their own these detectors are not powerful enough to correctly place a landmark, as they do not take in enough context. Therefore their output is combined with a shape model which helps to disambiguate the possible locations for each landmark by constraining their positions to ones that make sense, e.g. are anatomically correct in the case of fitting faces. Fig. 2.1 illustrates the model.

There have been many variations on CLM in the literature. They typically vary in local appearance model, face shape model, and optimisation method [18].

In [11] they introduce a method for optimising a point distribution model for the shape and the local detector outputs jointly; they identify a number of previous works which this directly improves on [20]–[24].

Regression based methods

Regression based methods learn a mapping from face image appearance to landmark locations directly, without a parametric model for shape. They may use local image patch detectors as in CLM, but the models are powerful enough not to need the shape constraint.

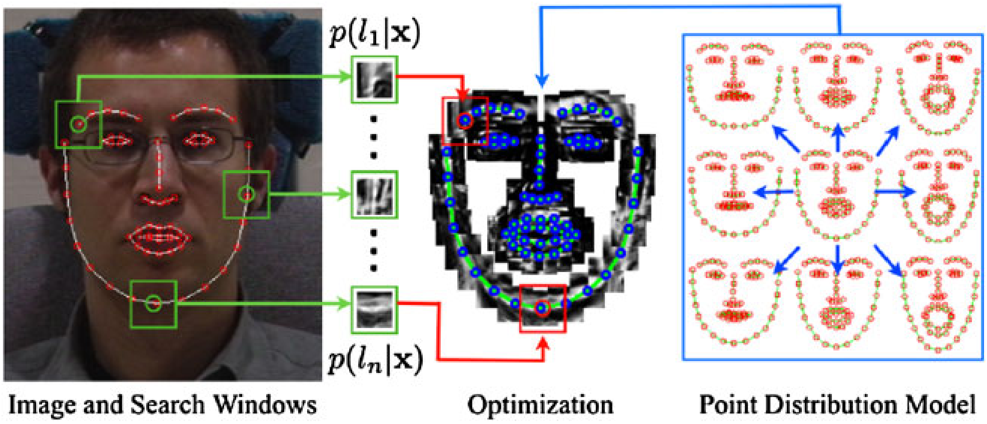

Figure 2.1: An illustration of CLM and its components. The left shows how each local model is run on an image patch taken from around a landmark point, producing a response map. The right illustrates the face shape model. They are combined during optimisation (centre) by picking landmark locations that are likely in the local models and fit the face shape model. Source: Adapted from [11]

2.2 Active Appearance Model design

An AAM is defined by the shape model, the appearance model, an image warping algorithm, and the fitting algorithm. Closely related but not strictly part of the AAM is the choice of image feature. This is typically applied as a pre-processing step, and while it can have a great effect on performance, it doesn't typically affect the structure of the model. Image features are examined in sec. 2.4.

The shape and appearance model both have a similar structure:

$$\boldsymbol{s} = \bar{\boldsymbol{s}} + \boldsymbol{\Phi}_{\boldsymbol{s}}\boldsymbol{p}_{\boldsymbol{s}}\boldsymbol{a} = \bar{\boldsymbol{a}} + \boldsymbol{\Phi}_{\boldsymbol{a}}\boldsymbol{p}_{\boldsymbol{a}} \tag{2.1}$$

They are both modeled as a mean vector ($\bar{\boldsymbol{s}}$, $\bar{\boldsymbol{a}}$) added to a linear combination of basis vectors ($\boldsymbol{\Phi}_{\boldsymbol{s}}$, $\boldsymbol{\Phi}_{\boldsymbol{s}}$) which are weighted by a parameter ($\boldsymbol{p}_{\boldsymbol{s}}$, $\boldsymbol{p}_{\boldsymbol{s}}$). This structure comes from both of them being constructed with principle component analysis (PCA), a technique for expressing high dimensional data in fewer dimensions. This throws away some of the variability in the original data, but this is considered to be variability due to noise. The basis vectors $\boldsymbol{\Phi}$ are a subset of the eigenvectors of the co-variance matrix of the data.

The shape model is known as a point distribution model (PDM) [25]. It is learned from a set of training images annotated with a set of landmark points. The landmark points have both global position variation, due to objects being in different parts of the images, and shape variation due to non-rigid variation of the objects. This is too complex to learn directly, so first the global variation is removed. For this an algorithm called generalised Procrustes analysis is used [26], which iteratively works out how to rotate, translate, and scale each shape so as to be nearly on top of each other. With this the mean shape $\bar{\boldsymbol{s}}$ and the covariance $\boldsymbol{\Sigma}$ of the points can be calculated, and finding the eigendecomposition of $\boldsymbol{\Sigma}$ gives the basis vectors. The subset of eigenvectors to keep is a trade-off between accuracy and speed, as more vectors mean more parameters to be found when fitting, but fewer means there may not be enough to accurately capture the variation in the object. In this work 15 eigenvectors are kept for each model.

To learn the appearance model the images must be warped to a reference shape, typically the mean shape $\bar{\boldsymbol{s}}$. This produces a shape-free image patch, which allows appearance to be learned in the absence of shape. PCA is then applied again to the shape-free patches to find $\boldsymbol{\Phi}_{\boldsymbol{a}}$.

The images are warped by a piecewise affine warp, which is the most common choice. It requires the points to be triangulated, as in fig. 2.2. Each triangular piece can then be warped to the new shape.

The final piece of AAM is the inference algorithm. This is typically formed as a least squares error minimisation problem for which Gauss-Newton optimisation is used. The Gauss-Newton algorithm is an iterative gradient descent algorithm, where the parameters are updated at each step by moving down the gradient of the error function.

2.3 AAM variations and advancements

The limiting factor for AAMs is largely speed. The appearance model is typically high dimensional, in the order of 10^1–10^2 principle components are typical, especially with multichannel images (or with multichannel image descriptors as in this work). Traditional Gauss-Newton optimisation requires computing and inverting large Hessian matrices to find the gradient and this is computationally expensive. Advances in AAMs then have mostly focused on this optimisation step.

In the original work on AAMs, computing the parameter update analyti-

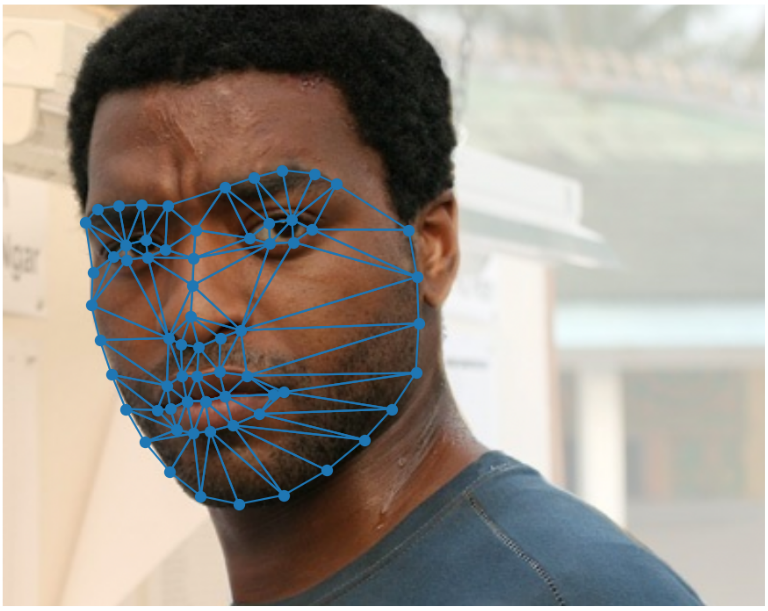

Figure 2.2: An image annotated with the traingles necessary for piecewise affine warping

cally was prohibitive on the hardware of the time. Therefore they used an additional step during learning to learn a linear approximation of the update step based on the image. The idea was that the gradient direction on the training images would generalise to unseen images. There are several iterations of this idea, improving the model for the parameter update. However, these methods trade away a fair bit of accuracy, robustness, and generalisability for their speed.

The other form of improvement for fitting was in the analytical methods for computing the update, along with increasing computing power making expensive algorithms more viable. An early breakthrough was the project-out inverse-compositional (POIC) algorithm of Matthews and Baker [27], which simplified the optimisation problem by decoupling the shape and appearance variation by "projecting-out" the appearance variation, working a subspace that is the orthogonal complement of the appearance variation as a result [28]. This algorithm is very fast, but not very robust, sacrificing accuracy for speed. It tends to break down when fitted to an image with high appearance variation or outliers. There is also simultaneous inverse composition (SIC), which is a slow but accurate algorithm [29].

More recent algorithm, the alternating inverse-composition (AIC), has been shown [1] to be equivalent to SIC (produces the same update step) but much faster. While not quite as fast as POIC, AIC is much more accurate.

Table 2.1: Algorithmic complexity for some of the main AAM inference algorithms [28].

Algorithm	Complexity
POIC	$\mathcal{O}(N_\mathrm{S} L_\mathrm{A} + N_\mathrm{S}^2)$
SIC	$\mathcal{O}((N_\mathrm{S} + N_\mathrm{A})^2 L_\mathrm{A} + (N_\mathrm{S} + N_\mathrm{A})^3)$
AIC	$\mathcal{O}(N_\mathrm{S}^2 N_\mathrm{A}^2 + (N_\mathrm{S} + N_\mathrm{A}) L_\mathrm{A} + N_\mathrm{S}^3)$

In tbl. 2.1, N_S is the number of shape components, N_A is the number of appearance components, and L_A is the length of the appearance vector.

2.4 Image feature descriptors

An image feature is a point of interest in an image, and a feature descriptor is a measurement extracted from an image that attempts to describe the contents at the feature. By this definition, the pixel intensity values themselves are feature descriptors, though typically weak ones. The term descriptor makes no guarantees of strength or usefulness. Looked at from another angle, the fitted AAM instance is itself an feature descriptor. It is an attempt to describe the image contents at a point of interest such as a face, after all. Indeed, higher level algorithms may treat AAM as a black box feature extraction algorithm.

For the purposes of AAM itself however, image features have a few requirements. Some feature extraction algorithms for example only produce a sparse set of features; for AAM this isn't useful as the shape of the object is lost. The feature set must preserve shape. Some feature algorithms summarise an image in a regular grid of sub-regions. The default scale-invariant feature transform (SIFT) and histogram of oriented gradients (HOG) features are like this. These preserve shape, but lose spacial accuracy, as points can only be localised to the area of the sub-region. Therefore the feature set must be dense, i.e. computed at every pixel.

All the feature extraction algorithms used are based on image gradients. The gradient is high at edges in an image, where pixel values change abruptly. By doing this they aim to introduce some invariance to things like illumination changes.

A dense feature image is computed by applying a feature extraction function to an image. The function produces a vector of values for each pixel, turning a greyscale image with size $W \times H \times 1$ into an feature image with size $W \times H \times C$, where W, H, and C are the width, height, and feature vector sizes respectively. The value of C varies between features and can be affected by parameters for the feature function.

In the following sub-sections the 4 image features selected for this work are examined. They are Image Gradient Orientation, Histogram of Gradients, Dense Scale Invariant Feature Transform, and DAISY.

Table 2.2: Feature extraction algorithm parameter defaults

Feature type	Parameter values	Channels
IGO		2

Feature type	Parameter values	Channels
HOG	$N_{\text{bins}} = 9$, $N_{\text{block}} = 2$, cell $= 8 \times 8$	36
DSIFT	$N_{\text{bins}} = 9$, $N_{\text{block}} = 2$, cell $= 8 \times 8$	36
DAISY	$Q = 2, T = 4, H = 4$	36

Image Gradient Orientation

IGO is a image feature introduced in [30] and applied to principle component analysis of faces. The feature image is computed in the following manner:

1. Compute $\mathbf{G}_x = \mathbf{F}_x * \mathbf{I}$, $\mathbf{G}_y = \mathbf{F}_y * \mathbf{I}$; that is, convolve the image with a horizontal and vertical first derivative filter to produce the horizontal and vertical gradient component images. Examples of such filters are the central difference operator, Prewitt, and Sobel filters.
2. Compute $\mathbf{\Phi} = \arctan \frac{\mathbf{G}_y}{\mathbf{G}_x}$
3. The final image is $\frac{1}{\sqrt{N}}[\cos \mathbf{\Phi}^{\mathsf{T}}, \sin \mathbf{\Phi}^{\mathsf{T}}]^{\mathsf{T}}$. Thus $C = 2$ for IGO.

Histogram of Oriented Gradients

HOG is a feature introduced in [31]. It uses gradient orientations similarly to IGO, but instead of each descriptor only capturing the local gradient, a histogram of the orientations in the local area is produced. In the original use of HOG a descriptor was extracted in a grid with a lower resolution than the image. As discussed above however, this loses spacial accuracy for the AAM, and so here a descriptor is computed for every pixel in the input image, as in [28].

HOG has parameters affecting the dimension of the descriptor and the area over which orientations are aggregated. The orientations in a $N_{\text{cell}} \times N_{\text{cell}}$ area are aggregated into a histogram with N_{bins} bins. A block of $N_{\text{block}} \times N_{\text{block}}$ cells are then normalised, and the final descriptor vector for that block is the concatenation of the normalised histograms. Therefore the descriptor describes a $(N_{\text{block}} N_{\text{cell}})^2$ area of the image, and the descriptor vector has $C = N_{\text{bins}} N_{\text{blocks}}^2$ channels.

Dense Scale Invariant Feature Transform

The SIFT feature descriptor shares similarities to HOG in that it summarises gradients over patches of the image. In its original use [32] the goal was to extract a sparse set of key points of the image and then compute a descriptor vector at these key points. In this work however we essentially treat every pixel as a feature, producing a dense feature set. The main difference to HOG is that the area for which the histogram is computed is oriented relative to the gradient at that point [28].

SIFT has similar parameters to HOG. Orientations are summarised from an $N_{\text{cell}} \times N_{\text{cell}}$ region into an N_{bins} bin histogram, and then a block of $N_{\text{block}} \times N_{\text{block}}$ of cells are aggregated into the descriptor. The descriptor length is $C = N_{\text{bins}} N_{\text{blocks}}^2$ again.

In the original SIFT a Gaussian window is applied to the gradients before aggregation into the histogram, weighting them by their distance from the centre. In this work we use a “fast” variant of SIFT from the VLFeat library [33] which applies the Gaussian window to the histogram bins after aggregation. This is substantially faster, a benefit when computing the feature densely, but reportedly has little effect on the accuracy of results.

DAISY

The DAISY feature descriptor is introduced in [34], where it is used for the task of stereo depth estimation. It is designed for efficiency computed densely, as is done here.

DAISY is another method of summarising gradients over a region. Gradients are summarised from concentric circles around the feature centre, with more distant gradients contributing less. The radius R from the centre to the outermost ring controls the spatial dimension over which the orientations are aggregated. The number of concentric circles that samples are collected from is Q, the radius quantisation number. The number of samples from each ring is denoted T, and the number of bins in a single histogram is H. Finally, the total length of the descriptor vector is $C = H(Q \times T + 1)$.

3 Methodology and implementation

In this section we define the requirements we need for the experiments. After this we detail the design and implementation choices made to fulfill those requirements. This section provides the background details, and the actual experiments that were run are detailed in sec. 4.

To evaluate AAMs we will need suitable image datasets. In order to evaluate the use of AAM in real world scenarios we want our datasets to reflect real world images, and not be artificially controlled. In sec. 3.1 we give details of various options and select the ones used. We also need a way to evaluate fitting results for accuracy; it's helpful as well to make these criteria compatible with prior work, to make it possible to compare results across different papers. We cover this in sec. 3.2. Finally we will need a way to actually implement the algorithms used. In sec. 3.3 we cover how this was done.

3.1 Datasets

There are several popular datasets available that are suitable for deformable shape model fitting. Faces are a popular subject.

The 300 Faces In-The-Wild Challenge [9], [35] provides an excellent source for suitable datasets. As part of setting up the challenge several existing sets of images were annotated with the same 68 point landmark configuration. These datasets had existing annotations, but all had different configurations making cross-dataset comparison difficult. The sets for which they provided new annotations were:

Multi-PIE The CMU Multi-Pose Illumination, Illumination, and Expression dataset [36].

750,000 image dataset captured in controlled conditions. Available for a fee.

XM2VTS The Extended Multi Modal Verification for Teleservices and Security applications dataset [37].

2360 images captured in controlled conditions. Available for a fee.

FRGC-V2 The Face Recognition Grand Challange Version 2.0 dataset [38].

4950 images captured in controlled conditions. Available on a case-by-case basis after a license is signed by a research institution.

AR The AR Face Database [39].

4000 uncompressed images captured in controlled conditions. Available on request from a university affiliated email address.

LFPW The Labeled Face Parts in the Wild dataset [6].

1287 links to images on the internet. Only a subset of 811 training images and 224 testing images could be downloaded for [36]. Available openly for reasearch.

Helen The Helen dataset [7]

2330 images downloaded from the flickr.com web service. Available openly for research.

AFW The Annotated Faces in-the-Wild dataset [8].

250 images (468 faces) collected from the flickr.com web service. Available openly for research.

The 300 Faces In-The-Wild Challenge also provided a new dataset referred to as **IBUG** which was collected for the competition, with 135 images downloaded from the web showing wide variation in expression, illumination, and pose.

In addition to these datasets provided for training, they also collected a new dataset for testing the contestants' entries on. This dataset consists of 300 images taken indoors and 300 taken outdoors, hence the name of the challenge. All the images were found on the web. This dataset (600 images in total) is referred to as **300W**.

Any datasets that were not collected in-the-wild were not suitable for this project. In [1] it was shown that training an AAM with in-the-wild data greatly improves its generalisability. Because of this LFPW, Helen, AFW, IBUG, and 300W are suitable. The IBUG dataset and LFPW training set were used for training (946 images), with the 300W sets and the LFPW testing set (824 images) used for testing.

3.2 Result evaluation methodology

The accuracy of the algorithms was evaluated on the Euclidean distance between the fitted shape and the ground truth annotations on a point-to-point basis, normalised by the size of the ground truth annotation bounding box, as used in [8], [28]. Denoting $\boldsymbol{s}^{\mathrm{f}} = [x_1^{\mathrm{f}}, y_1^{\mathrm{f}}, \dots, x_{L_{\boldsymbol{s}}}^{\mathrm{f}}, y_{L_{\boldsymbol{s}}}^{\mathrm{f}}]$ and $\boldsymbol{s}^{\mathrm{g}} = [x_1^{\mathrm{g}}, y_1^{\mathrm{g}}, \dots, x_{L_{\boldsymbol{s}}}^{\mathrm{g}}, y_{L_{\boldsymbol{s}}}^{\mathrm{g}}]$ as the fitted and ground truth shape respectively, then the error between them is calculated as:

$$\text{Error} = \frac{1}{s_{\mathrm{bb}}^{\mathrm{g}} L_{\boldsymbol{s}}} \sum_{i=1}^{L_{\boldsymbol{s}}} \sqrt{(x_i^{\mathrm{f}} - x_i^{\mathrm{g}})^2 + (y_i^{\mathrm{f}} - y_i^{\mathrm{g}})^2} \tag{3.1}$$

Where $s_{\mathrm{bb}}^{\mathrm{g}}$ is the mean side length of the ground truth shape's bounding box:

$$s_{\mathrm{bb}}^{\mathrm{g}} = \frac{(\max x_i^{\mathrm{g}} - \min x_i^{\mathrm{g}}) + (\max y_i^{\mathrm{g}} - \min y_i^{\mathrm{g}})}{2} \tag{3.2}$$

3.3 Algorithm implementation

To implement the experiments Python and the `menpo`/`menpofit` packages were used. The AAMs were trained on the LFPW and IBUG datasets, and tested on the 300W challenge dataset.

The selection of Python was driven by the availability of the `menpo` packages. `menpo` and `menpofit` are Python packages that implement a framework for deformable object modelling, including extensible AAM classes.

4 Results and analysis

In this section the experimental results are presented. The power of each image feature for human face detection with an AAM was evaluated by building an AAM using each feature and running them against the testing datasets. The cumulative error was plotted for each dataset in fig. 4.1.

Table 4.1: Mean and median errors for each testset and image feature combination.

Testset	Image feature	Mean error	Median error
300W	DAISY	0.076	0.043
	DSIFT	0.096	0.047
	HOG	0.065	0.037
	IGO	0.110	0.100
	Intensity values	0.130	0.110
300W Indoor	DAISY	0.077	0.042
	DSIFT	0.095	0.047
	HOG	0.062	0.038
	IGO	0.110	0.100
	Intensity values	0.130	0.120
300W Outdoor	DAISY	0.076	0.043
	DSIFT	0.097	0.047
	HOG	0.069	0.037
	IGO	0.110	0.099
	Intensity values	0.130	0.110
LFPW	DAISY	0.035	0.023
	DSIFT	0.042	0.025
	HOG	0.029	0.023
	IGO	0.065	0.036
	Intensity values	0.075	0.052

On each dataset the HOG feature produces the best results. LFPW is the easiest dataset and HOG achieves 5% error or less on about 90% of the images. DSIFT and DAISY are not far behind however. IGO shows

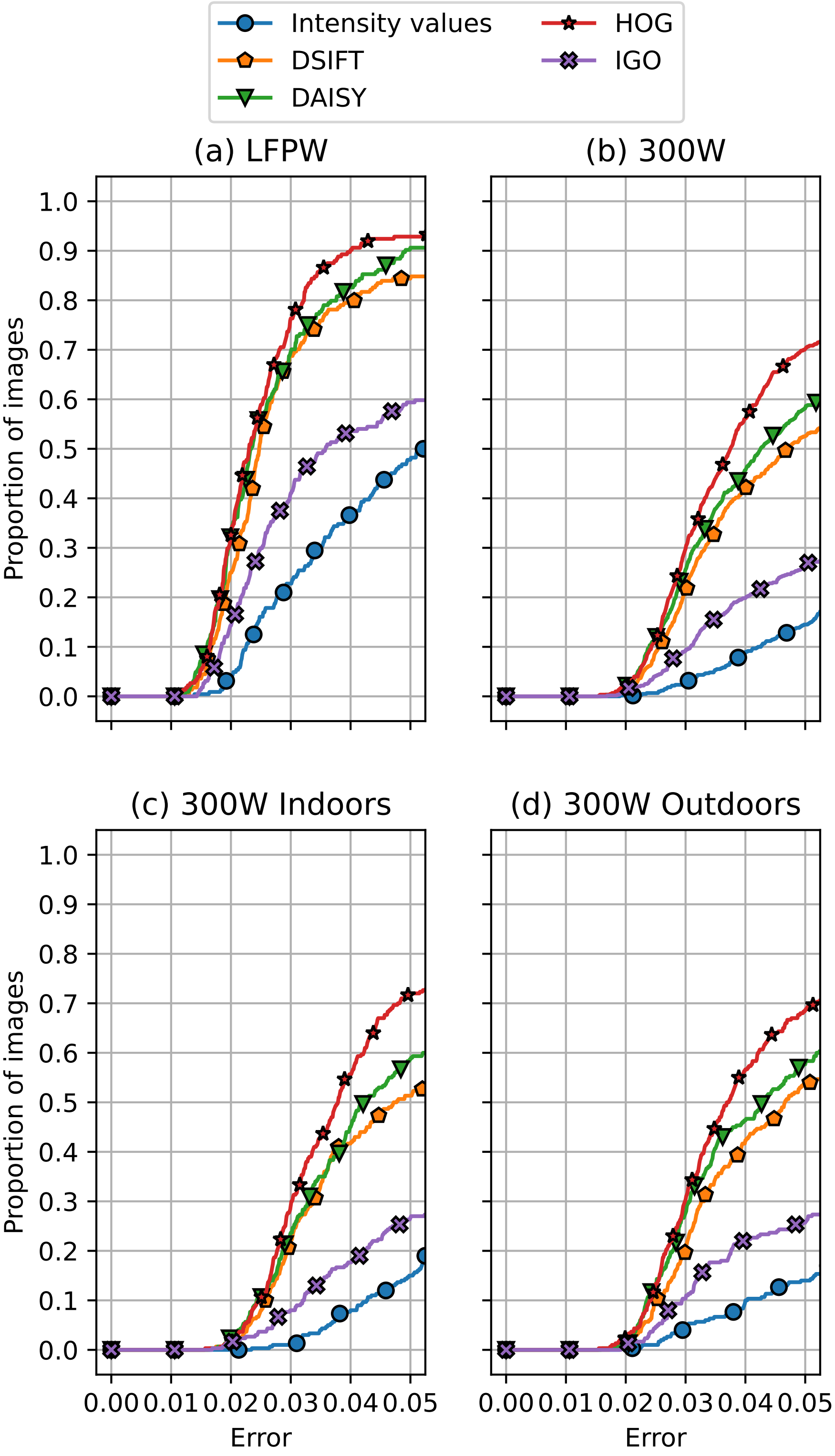

Figure 4.1: Cumulative error diagrams

improvement over pixel intensity values, but not in the significant way that the others do.

The similar performance from HOG, DAISY, and SIFT is not surprising considering the similarities between these descriptors; they all aggregate image gradients over a local area. It may be that adjusting the parameters for DAISY and SIFT would bring them up to the level of HOG, or or send HOG even higher.

5 Conclusion

This project investigated the use of robust gradient-based feature descriptors for the task of facial landmark fitting with Active Appearance Models. The results show that AAMs can be trained with just a few hundred images, generalise well to unseen images showing wide variations in appearance, illumination, pose, and expression. The results also gave insight into the relative power of different image features; HOG consistently had the best performance, closely followed by DAISY and SIFT.

Bibliography

[1] G. Tzimiropoulos and M. Pantic, “Optimization Problems for Fast AAM Fitting in-the-Wild,” in *2013 IEEE International Conference on Computer Vision*, Sydney, Australia, Dec. 2013, pp. 593–600. doi: 10.1109/ICCV.2013.79.

[2] S. A. Papert, “The Summer Vision Project,” *AI Memos (1959 - 2004)*, Jul. 1966, Accessed: May 14, 2022. [Online]. Available: https://dspace.mit.edu/handle/1721.1/6125

[3] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, 2nd ed. Boston: Pearson, 2012.

[4] R. Szeliski, *Computer Vision: Algorithms and Applications*. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-030-34372-9.

[5] M. B. Stegmann, “Active appearance models: Theory, extensions and cases,” Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2000. Available: http://www.imm.dtu.dk/~aam/main/

[6] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, “Localizing parts of faces using a consensus of exemplars,” in *CVPR 2011*, Jun. 2011, pp. 545–552. doi: 10.1109/CVPR.2011.5995602.

[7] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, “Interactive Facial Feature Localization,” in *Computer Vision – ECCV 2012*, Berlin, Heidelberg, 2012, pp. 679–692. doi: 10.1007/978-3-642-33712-3_49.

[8] X. Zhu and D. Ramanan, “Face detection, pose estimation, and landmark localization in the wild,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2012, pp. 2879–2886. doi: 10.1109/CVPR.2012.6248014.

[9] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, “300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge,” in *2013 IEEE International Conference on Computer Vision Workshops*, Dec. 2013, pp. 397–403. doi: 10.1109/ICCVW.2013.59.

[10] D. Cristinacce and T. F. Cootes, “Feature Detection and Tracking with Constrained Local Models,” in *Procedings of the British Machine Vision Conference 2006*, Edinburgh, 2006, pp. 95.1–95.10. doi: 10.5244/C.20.95.

[11] J. M. Saragih, S. Lucey, and J. F. Cohn, “Deformable Model Fitting by Regularized Landmark Mean-Shift,” *Int J Comput Vis*, vol. 91, no. 2, pp. 200–215, Jan. 2011, doi: 10.1007/s11263-010-0380-4.

[12] J. Yan, Z. Lei, D. Yi, and S. Z. Li, “Learn to Combine Multiple Hypotheses for Accurate Face Alignment,” in *2013 IEEE International Conference on Computer Vision Workshops*, Sydney, Australia, Dec. 2013, pp. 392–396. doi: 10.1109/ICCVW.2013.126.

[13] V. Kazemi and J. Sullivan, “One millisecond face alignment with an ensemble of regression trees,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 1867–1874. doi: 10.1109/CVPR.2014.241.

[14] X. Xiong and F. De la Torre, “Supervised Descent Method and Its Applications to Face Alignment,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2013, pp. 532–539. doi: 10.1109/CVPR.2013.75.

[15] J. Zhang, S. Shan, M. Kan, and X. Chen, “Coarse-to-Fine Auto-Encoder Networks (CFAN) for Real-Time Face Alignment,” in *Computer Vision – ECCV 2014*, Cham, 2014, pp. 1–16. doi: 10.1007/978-3-319-10605-2_1.

[16] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool, “Real-time facial feature detection using conditional regression forests,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2012, pp. 2578–2585. doi: 10.1109/CVPR.2012.6247976.

[17] H. Yang and I. Patras, “Privileged information-based conditional regression forest for facial feature detection,” in *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Apr. 2013, pp. 1–6. doi: 10.1109/FG.2013.6553766.

[18] Y. Wu and Q. Ji, “Facial Landmark Detection: A Literature Survey,” *Int J Comput Vis*, vol. 127, no. 2, pp. 115–142, Feb. 2019, doi: 10.1007/s11263-018-1097-z.

[19] G. J. Edwards, C. J. Taylor, and T. F. Cootes, “Interpreting face images using active appearance models,” in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, Apr. 1998, pp. 300–305. doi: 10.1109/AFGR.1998.670965.

[20] T. F. Cootes and C. J. Taylor, “Active Shape Models - ‘smart snakes’,” in *Procedings of the British Machine Vision Conference 1992*, Leeds, 1992, pp. 28.1–28.10. doi: 10.5244/C.6.28.

[21] K. Nickels and S. Hutchinson, “Estimating uncertainty in SSD-based feature tracking,” *Image and Vision Computing*, vol. 20, no. 1, pp. 47–58, Jan. 2002, doi: 10.1016/S0262-8856(01)00076-2.

[22] X. S. Zhou, A. Gupta, and D. Comaniciu, “An information fusion framework for robust shape tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 115–129, Jan. 2005, doi: 10.1109/TPAMI.2005.3.

[23] Y. Wang, S. Lucey, J. F. Cohn, and J. Saragih, “Non-rigid face tracking with local appearance consistency constraint,” in *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, Amsterdam, Netherlands, Sep. 2008, pp. 1–8. doi: 10.1109/AFGR.2008.4813409.

[24] L. Gu and T. Kanade, “A Generative Shape Regularization Model for Robust Face Alignment,” in *Computer Vision – ECCV 2008*, vol. 5302, D. Forsyth, P. Torr, and A. Zisserman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 413–426. doi: 10.1007/978-3-540-88682-2_32.

[25] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Training Models of Shape from Sets of Examples,” in *Procedings of the British Machine Vision Conference 1992*, Leeds, 1992, pp. 2.1–2.10. doi: 10.5244/C.6.2.

[26] S. J. D. Prince, *Computer Vision: Models, Learning, and Inference*, 1st edition. New York: Cambridge University Press, 2012.

[27] I. Matthews and S. Baker, “Active Appearance Models Revisited,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, Nov. 2004, doi: 10.1023/B:VISI.0000029666.37597.d3.

[28] E. Antonakos, J. Alabort-i-Medina, G. Tzimiropoulos, and S. P. Zafeiriou, “Feature-Based Lucas–Kanade and Active Appearance Models,” *IEEE Transactions on Image Processing*, vol. 24, no. 9, pp. 2617–2632, Sep. 2015, doi: 10.1109/TIP.2015.2431445.

[29] S. Baker, R. Gross, and I. Matthews, “Lucas-Kanade 20 Years On: A Unifying Framework: Part 3,” p. 51.

[30] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, “Subspace Learning from Image Gradient Orientations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 12, pp. 2454–2466, Dec. 2012, doi: 10.1109/TPAMI.2012.40.

[31] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, Jun. 2005, vol. 1, pp. 886–893 vol. 1. doi: 10.1109/CVPR.2005.177.

[32] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Sep. 1999, vol. 2, pp. 1150–1157 vol.2. doi: 10.1109/ICCV.1999.790410.

[33] A. Vedaldi and B. Fulkerson, “VLFeat: An open and portable library of computer vision algorithms.” 2008. Available: http://www.vlfeat.org/

[34] E. Tola, V. Lepetit, and P. Fua, “DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, May 2010, doi: 10.1109/TPAMI.2009.77.

[35] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, “300 Faces In-The-Wild Challenge: database and results,” *Image and Vision Computing*, vol. 47, pp. 3–18, Mar. 2016, doi: 10.1016/j.imavis.2016.01.002.

[36] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, “Multi-PIE,” *Proc Int Conf Autom Face Gesture Recognit*, vol. 28, no. 5, pp. 807–813, May 2010, doi: 10.1016/j.imavis.2009.08.002.

[37] K. Messer, J. Matas, J. Kittler, K. Jonsson, J. Luettin, and G. Maître, “Xm2vtsdb: The extended m2vts database,” *Proc. of Audio- and Video-Based Person Authentication*, Apr. 2000.

[38] P. J. Phillips *et al.*, “Overview of the face recognition grand challenge,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, Jun. 2005, vol. 1, pp. 947–954 vol. 1. doi: 10.1109/CVPR.2005.268.

[39] A. Martinez and R. Benavente, “The AR Face Database,” *Tech. Rep. 24 CVC Technical Report*, Jan. 1998.