Gauss-Newton Deformable Part Models for Face Alignment in-the-Wild

Georgios Tzimiropoulos
1. School of Computer Science
University of Lincoln, U.K.
2. Department of Computing
Imperial College London, U.K.
gtzimiropoulos@lincoln.ac.uk

Maja Pantic
1. Department of Computing
Imperial College London, U.K.
2. University of Twente
The Netherlands
m.pantic@imperial.ac.uk

Abstract

Arguably, Deformable Part Models (DPMs) are one of the most prominent approaches for face alignment with impressive results being recently reported for both controlled lab and unconstrained settings. Fitting in most DPM methods is typically formulated as a two-step process during which discriminatively trained part templates are first correlated with the image to yield a filter response for each landmark and then shape optimization is performed over these filter responses. This process, although computationally efficient, is based on fixed part templates which are assumed to be independent, and has been shown to result in imperfect filter responses and detection ambiguities. To address this limitation, in this paper, we propose to jointly optimize a part-based, trained in-the-wild, flexible appearance model along with a global shape model which results in a joint translational motion model for the model parts via Gauss-Newton (GN) optimization. We show how significant computational reductions can be achieved by building a full model during training but then efficiently optimizing the proposed cost function on a sparse grid using weighted least-squares during fitting. We coin the proposed formulation Gauss-Newton Deformable Part Model (GN-DPM). Finally, we compare its performance against the state-of-the-art and show that the proposed GN-DPM outperforms it, in some cases, by a large margin. Code for our method is available from `http://ibug.doc.ic.ac.uk/resources`

1. Introduction

Deformable models are extremely popular in computer vision for two reasons. The first reason is that they span a wide range of applications. For example, they have been extensively used for analyzing faces and medical images. The second reason is that learning and fitting deformable models is one of the most challenging problems in computer vision research. While some impressive developments have been reported over the last years, arguably, we are still far away from considering this problem solved. The focus of this work is on the difficult problem of fitting facial deformable models to unconstrained images, also known as face alignment *in-the-wild*.

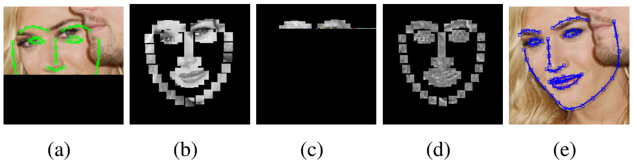


Figure 1. Overview of Gauss-Newton Deformable Part Models: Given a shape estimate (a), parts are extracted around the current estimate of the landmarks' location (b), and reconstructed by a part-based, trained in-the-wild, flexible appearance model (c). The reconstruction error (d) drives the joint optimization of shape and appearance which is performed by an efficient and robust Gauss-Newton algorithm. The fitted shape is shown in (e).

Perhaps the most well-known type of deformable models are Active Shape Models (ASMs) and Active Appearance Models (AAMs) [5, 4]. ASMs are generative models of global shape built by applying Principal Component Analysis (PCA) to a set of aligned training shapes. Appearance in ASMs is modelled locally by learning a patch expert for each point of the shape model. Fitting the shape model to a new image is an iterative process that entails (a) convolving the local experts with the image, (b) generating candidate locations for the landmarks by finding the locations of the maximum filter responses, and (c) refining these locations by a global shape optimization procedure. AAMs were proposed as a sophisticated extension of ASMs for modelling the process of generating instances of both shape and appearance of a specific object class. The shape model of an AAM is the same point distribution model of an ASM. An AAM additionally models global appearance using PCA, however, after removing texture variation due to shape deformation. As in ASMs, fitting an AAM to an image is an iterative process. At each iteration an update for the model parameters is estimated which is typically a function of the error between the model instance and the given image. AAM fitting approaches include learning this function via regression [4, 15, 16] or directly minimizing the error via non-linear optimization [13, 19].

In general, AAM fitting is considered a difficult problem, especially when the model is fitted to images of unseen variations. Recent research effort has concentrated on part-based deformable models which are considered easier to optimize, more robust and accurate due to the use of the local, part-based representation which is less sensitive to lighting and global appearance variations [17, 22]. A popular and very successful approach is the family of methods coined Constrained Local Models (CLMs) one example of which is the original ASM formulation [17]. CLMs differ from ASMs mainly in the way that filter responses are used in the optimization of the global shape model [6, 8, 20, 17, 12, 1]. For example in [6] a general purpose optimizer is used, while [8, 20, 17, 12] propose better tailored optimization strategies by assuming various parametric/non-parametric models for the filter responses. We refer the reader to [17] for a seminal framework which unifies various CLM approaches. The CLM of [1] along with the shape regression approach of [3] and the Supervised Descent Method (SDM) of [21] are considered the state-of-the-art in face alignment.

A common characteristic of the majority of the aforementioned works is that landmark detectors are learned discriminately during training and remain fixed during fitting. This process, although computationally efficient, has the following limitations: (a) it is based on a fixed appearance part model and (b) object parts are assumed to be independent, and each landmark detector is applied independently of the others. Because of (a) and (b), such an approach has been shown to result in imperfect filter responses and detection ambiguities which hinder the accurate localization of landmarks [17]. Hence, the focus of most works is how these inaccuracies and ambiguities can be remedied by the global shape optimization step.

Main contributions. To alleviate (a) and (b) mentioned above, we propose Gauss-Newton Deformable Part Models (GN-DPMs). Unlike the majority of part-based face alignment methods (like CLMs), in the proposed GN-DPMs, the fitting procedure is totally different: there is no correlation-based independent local search followed by global shape optimization; instead we propose to jointly optimize a part-based, trained in-the-wild, flexible appearance model along with a global shape model via efficient and robust Gauss-Newton (GN) optimization [9, 13, 19]. We show that the proposed model/fitting strategy results in a joint translational motion model for the model parts the location of which along with their appearance are jointly updated at each iteration. Please see Fig. 1 for an overview of our approach. As in [21], we use SIFT features [11] to build the appearance model of GN-DPM. Although very robust such formulation results in a high dimensional appearance model which renders the fitting process slow. To alleviate this problem we show how significant computational reductions can be achieved by building a full model during training but then efficiently optimizing the proposed cost function on a sparse grid during fitting. Via a number of experiments, we show that the proposed GN-DPM outperforms the state-of-the-art SDM [21] in all three major in-the-wild facial databases, namely LFPW [2], Helen [10] and AFW [22].

2. Related work and motivation

The proposed GN-DPM entails fitting a part-based, trained in-the-wild, flexible appearance model to a new image using efficient and robust GN optimization. As such our method is primarily related to the generative GN formulation of [9, 13]. In [9], the authors proposed a GN formulation for fitting a rigid but flexible linear generative appearance model learned via PCA. In [13], the authors extend the work of [9] in a number of ways for the case of deformable models and AAMs. In general, fitting AAMs to unconstrained images is considered a difficult task. Perhaps, the most widely acknowledged reason for this is the limited representational power of the appearance model which is unable to generalize well to unseen variations. As it was recently shown in [19] though, when the appearance model of the AAM is trained in-the-wild and exact GN algorithms are used for model fitting, AAMs perform notably well for the case of unconstrained images even without having to resort to shape priors, robust features or robust norms for improving performance.

The proposed GN-DPM also employs a flexible, linear generative appearance model trained in-the-wild and fitted via GN, however, motivated by the recent success of part-based models [6, 20, 17, 21], it uses parts and a translational motion model as opposed to the holistic appearance model and the piecewise affine warp used in [19]. Among a large number of works in part-based deformable face alignment, our algorithm is more closely related to [20] and [21]. In particular, the shape optimization step employed in [20] is inspired by the problem of fitting a fixed part-based template to an image via GN. However, the authors in [20] advocated a standard CLM framework in which a set of fixed discriminatively trained part templates are first correlated with the image to yield a set of filter responses, each response is approximated by a quadratic, and then the aforementioned shape optimization step is performed to update the current shape estimate. Contrary to [20], we advocate a flexible part-based appearance model trained in-the-wild and propose to jointly optimize shape and appearance via an efficient and robust GN algorithm. A critical aspect in GN optimization is how to increase the basin of attraction. To this end, and similarly to [21], we also employed SIFT features to build the appearance model of the proposed GN-DPM.

3. Generative Deformable Part Models in-the-Wild

In our formulation, a generative DPM is described by generative models of global shape and local appearance both learned via PCA, as in the original CLM paper of [6] [1]. A key feature of the appearance model is that it is learned from all parts jointly, and, hence parts, although capture local appearance, are not assumed independent.

Learning the shape model of the generative DPM requires strong supervision, and can be summarized in 4 steps: (a) u landmarks $\mathbf{l}_i = [x_{i,1}, y_{i,1}, \dots x_{i,u}, y_{i,u}]$ are consistently annotated across D training face images $\mathbf{I}_i$, $i = 1, \dots, D$. (b) Procrustes Analysis is applied to remove similarity (scale, rotation and translation) transformations. (c) PCA is applied on the resulting shapes to obtain a shape model defined by the mean shape $\mathbf{s}_0$ and n shape eigenvectors $\mathbf{s}_i$ compactly represented as columns of $\mathbf{S} \in \mathcal{R}^{\{2u,n\}}$. (d) $\mathbf{S}$ is appended with 4 similarity eigenvectors [13] and re-orthonormalized. An instance of the shape model $\mathbf{s}(\mathbf{p})$ is given by

$$\mathbf{s}(\mathbf{p}) = \mathbf{s}_0 + \mathbf{S}\mathbf{p}, \tag{1}$$

where $\mathbf{p} \in \mathcal{R}^n$ is the vector of the shape parameters. We also denote by $\mathbf{s}_k = [\mathbf{x}_k \ ; \ \mathbf{y}_k]$ and and $\mathbf{s}_{i,k} = [\mathbf{x}_k^{\mathbf{s}_i} \ ; \ \mathbf{y}_k^{\mathbf{s}_i}]$ the $k-$th landmark point of $\mathbf{s}(\mathbf{p})$ and $\mathbf{s}_i$, respectively. These are related by

$$\mathbf{s}_k = [\mathbf{x}_k \ ; \ \mathbf{y}_k] = [\mathbf{x}_k^{\mathbf{s}_0} + \sum_{i=1}^{n} \mathbf{x}_k^{\mathbf{s}_i}\mathbf{p}_i \ ; \ \mathbf{y}_k^{\mathbf{s}_0} + \sum_{i=1}^{n} \mathbf{y}_k^{\mathbf{s}_i}\mathbf{p}_i]. \tag{2}$$

The appearance model of the generative DPM is obtained by (a) warping each training image $\mathbf{I}_i$ to a reference frame so that similarity transformations are removed, (b) extracting a $N_p = N_s \times N_s$ pixel-based part (i.e. patch) around each landmark, (c) obtaining a part-based texture for the whole image by concatenating all parts in a $N = uN_p$ vector, and (d) applying PCA on the part-based textures of all training images. In this way, we obtain the mean appearance $\mathbf{A}_0$, and m appearance eigenvectors $\mathbf{A}_i$ compactly represented as columns of $\mathbf{A} \in \mathcal{R}^{\{N,m\}}$. An instance of the appearance model $\mathbf{A}(\mathbf{c})$ is given by

$$\mathbf{A}(\mathbf{c}) = \mathbf{A}_0 + \mathbf{A}\mathbf{c}, \tag{3}$$

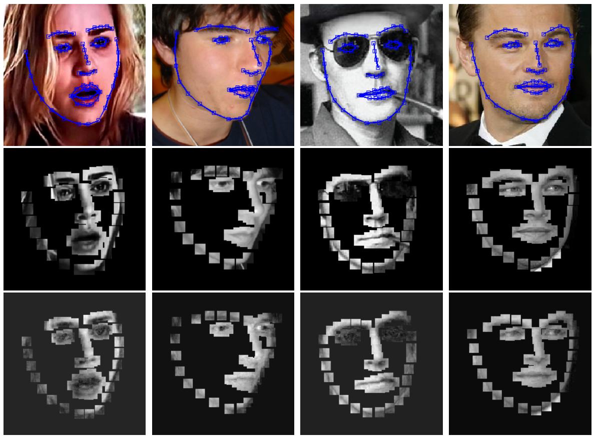


Figure 2. First row: Images taken from the test set of LFPW along with their ground truth landmarks. The images were not seen during training. Second row: parts extracted around landmarks. Third row: Reconstruction of the parts from the part-based appearance subspace. The appearance subspace is powerful because it was built in the wild.

where $\mathbf{c} \in \mathcal{R}^m$ is the vector of the appearance parameters. It is worth noting that each $\mathbf{A}_i$ (this also applies to the part-based texture representation of each training image $\mathbf{I}_i$) can be re-arranged as a $u \times N_p$ representation $[\mathbf{A}^{i,1} \ \mathbf{A}^{i,2} \ \dots \ \mathbf{A}^{i,N_p}]$. Each column $\mathbf{A}^{i,j} \in \mathcal{R}^u$ contains u pixels all belonging to a different part but all sharing the same index location j within their part. This representation allows us to interpret each patch as a N_p-dimensional descriptor for the corresponding landmark. Finally, we define $\mathbf{A}^j = [\mathbf{A}^{1,j} \ \mathbf{A}^{2,j} \ \dots \ \mathbf{A}^{m,j}] \in \mathcal{R}^{u \times m}$.

A notable deviation from prior work is that we leverage recently annotated in-the-wild face databases [14, 18] to train the generative DPM. In this way, the learned appearance model is powerful enough to faithfully reconstruct unseen unconstrained face images. Consider for example the images shown in the first row of Fig. 2. These are test images from the LFPW data set. The images were not seen during training, but similar images of unconstrained nature were used to train the shape and appearance model of the DPM. The second row of Fig. 2 shows the parts extracted around the ground truth landmarks and the third row the reconstruction of the parts from the appearance subspace. As we may see the part-based appearance model is powerful enough to reconstruct the parts almost perfectly.

4. Fitting Generative Deformable Part Models with Gauss-Newton

The proposed Gauss-Newton DPM is based on fitting the generative DPM of Section 3 to a test image using non-linear least squares optimization [9, 13, 19].

[1] Unlike [6], both models are kept independent [13] i.e. we do not apply a third PCA on the embeddings of the shape and texture.

4.1. 1-pixel GN-DPM

We start by describing the fitting process of a simplified version of the generative DPM by assuming that the patch for each landmark $\mathbf{s}_k$ is reduced to 1×1 ($N_s = 1$), that is 1 pixel is used to represent the appearance of each landmark and similarly the appearance model in (3) has a total of u pixels. In this case, the construction of the appearance model in Section 3 implicitly assumes a translational motion model in which each training image is sampled at $N = u$ locations $\mathbf{I}_i(\mathbf{l}_i)$ and then u pixels are shifted to a common reference frame which is defined as the frame of the mean shape $\mathbf{s}_0$. In this model, a model instance $\mathbf{M}_y$ is created by first generating u pixels using (3) for some $\mathbf{c} = \mathbf{c}_y$ and then shifting these pixels to u pixel locations obtained from (1) for some $\mathbf{p} = \mathbf{p}_y$. Hence, we can write

$$M_y(\mathbf{s}(\mathbf{p}_y)) = \mathbf{A}(\mathbf{c}_y). \tag{4}$$

Optimization of GN-DPM. The above model can be readily used to locate the landmarks in an unseen image $\mathbf{I}$ using non-linear least-squares. In particular, we wish to find $\{\mathbf{p}, \mathbf{c}\}$ such that

$$\arg\min_{\mathbf{p},\mathbf{c}} ||\mathbf{I}(\mathbf{s}(\mathbf{p})) - \mathbf{A}(\mathbf{c})||^2. \tag{5}$$

The difference term in the above cost function is linear in $\mathbf{c}$ but non-linear in $\mathbf{p}$. We therefore proceed by applying a first-order Taylor approximation. As mentioned in [13], we can linearize either the image or the model. The former case results in *forward* algorithms whereas the latter case in *inverse* algorithms. In this paper, we follow the inverse case which can result in significant pre-computations. Therefore, we proceed by linearizing the model. To do so we first write $\mathbf{I} = \mathbf{I}(\mathbf{s}(\mathbf{p}))$, and $\mathbf{A}_i = \mathbf{A}_i(\mathbf{s}(\mathbf{p} = \mathbf{0})) = \mathbf{A}_i(\mathbf{s}_0)$. Then, we have

$$\arg\min_{\Delta\mathbf{p},\Delta\mathbf{c}} ||\mathbf{I} - \mathbf{A}_0 - \mathbf{J}_0\Delta\mathbf{p} - \sum_{i=1}^{m}(c_i + \Delta c_i)(\mathbf{A}_i + \mathbf{J}_i\Delta\mathbf{p})||^2, \tag{6}$$

where $\mathbf{J}_i \in \mathcal{R}^{N \times n}$ is the Jacobian of $\mathbf{A}_i$ (notice that $N = u$). We construct $\mathbf{J}_i$ as follows: The $k-$th row of $\mathbf{J}_i$ contains the $1 \times n$ vector $[\mathbf{A}_{i,x}(\mathbf{s}_{0,k}) \;\; \mathbf{A}_{i,y}(\mathbf{s}_{0,k})]\frac{\partial \mathbf{s}_k(\mathbf{p})}{\partial \mathbf{p}} \mid_{\mathbf{p}=\mathbf{0}}$. $\mathbf{A}_{i,x}$ and $\mathbf{A}_{i,y}$ are the x and y gradients of $\mathbf{A}_i$ [2]. Finally differentiation of (2) yields $\frac{\partial \mathbf{s}_k(\mathbf{p})}{\partial \mathbf{p}} \mid_{\mathbf{p}=\mathbf{0}} = [\mathbf{x}_k^{\mathbf{s}_1} \ldots \mathbf{x}_k^{\mathbf{s}_n} \; ; \; \mathbf{y}_k^{\mathbf{s}_1} \ldots \mathbf{y}_k^{\mathbf{s}_n}] \in \mathcal{R}^{2 \times n}$.

An update for $\Delta\mathbf{c}$ and $\Delta\mathbf{p}$ can be obtained only after second order terms are omitted as follows

$$\arg\min_{\Delta\mathbf{p},\Delta\mathbf{c}} ||\mathbf{I} - \mathbf{A}(\mathbf{c}) - \mathbf{A}\Delta\mathbf{c} - \mathbf{J}\Delta\mathbf{p})||^2, \tag{7}$$

where $\mathbf{J} = \mathbf{J}_0 + \sum_{i=1}^{m} c_i\mathbf{J}_i$. To optimize (7) we follow the same strategy as the one used for the Fast-SIC algorithm described in [19]. More specifically, we optimize (7) with respect to $\Delta\mathbf{c}$, and then plug in the solution back to (7). Then, we can optimize (7), with respect to $\Delta\mathbf{p}$. Overall, we can update the appearance and shape parameters in an alternating fashion from

$$\Delta\mathbf{c} = \mathbf{A}^T(\mathbf{I} - \mathbf{A}(\mathbf{c}) - \mathbf{J}\Delta\mathbf{p}) \tag{8}$$

$$\Delta\mathbf{p} = \mathbf{H}_P^{-1}\mathbf{J}_P^T(\mathbf{I} - \mathbf{A}_0), \tag{9}$$

where $\mathbf{J}_P = \mathbf{P}\mathbf{J}$ and $\mathbf{H}_P = \mathbf{J}_P^T\mathbf{J}_P$ respectively, $\mathbf{P} = \mathbf{E} - \mathbf{A}\mathbf{A}^T$ is the projection operator that projects out appearance variation, and $\mathbf{E}$ is the identity matrix. The complexity per iteration is $O(nmN)$ for computing $\mathbf{J}_P$, $O(n^2N)$ for computing $\mathbf{H}_P$ and $O(n^3)$ for inverting $\mathbf{H}_P$.

Reducing the cost from $O(nmN + n^2N)$ to $O(mN + n^2N)$. We describe an approximation which results in significant reduction in the computational complexity and is applicable to all versions of GN-DPMs introduced in this paper. The main computational bottleneck in the above algorithm is the computation of the projected-out Jacobian $\mathbf{J}_P$. However, when computing (9), we can write $\mathbf{J}_P^T(\mathbf{I} - \mathbf{A}_0) = \mathbf{J}^T\mathbf{P}^T(\mathbf{I} - \mathbf{A}_0)$. Now $\mathbf{P}^T(\mathbf{I} - \mathbf{A}_0)$ takes $O(mN)$ and one can compute $\mathbf{J}$ as the Jacobian of $\mathbf{A}(\mathbf{c})$ also in $O(mN)$. Hence, if we approximate $\mathbf{H}_P$ with $\mathbf{H} = \mathbf{J}^T\mathbf{J}$, the overall cost of the algorithm is reduced to $O(mN + n^2N)$ where typically $m \approx n^2$. We observed no deterioration in performance when this approximation was used.

Inverse Composition Vs. Addition. A key feature of the inverse framework of [13] is that the update for the shape parameters is estimated in the model coordinate frame and then composed to the current shape estimate. For the piecewise affine warp used in [13], a first order approximation to inverse composition is used. On the contrary, because of the translational motion model employed in GN-DPMs, inverse composition is reduced to addition. To readily see this, let us first write $\mathbf{s}_y = f(\mathbf{s}_x; \mathbf{p}_a) = \mathbf{s}_x + \mathbf{S}\mathbf{p}_a$. Then, $\mathbf{s}_z = f(\mathbf{s}_y; \mathbf{p}_b) = \mathbf{s}_y + \mathbf{S}\mathbf{p}_b = \mathbf{s}_x + \mathbf{S}\mathbf{p}_a + \mathbf{S}\mathbf{p}_b = \mathbf{s}_x + \mathbf{S}(\mathbf{p}_a + \mathbf{p}_b)$, hence composition is reduced to addition. Similarly, we have $f(\mathbf{s}_x; \mathbf{p}_a)^{-1} = f(\mathbf{s}_x; -\mathbf{p}_a)$. Overall inverse composition is reduced to addition, and hence $\mathbf{p}$ can be readily updated in an additive fashion from $\mathbf{p} \leftarrow \mathbf{p} - \Delta\mathbf{p}$.

4.2. GN-DPM

Having defined the 1-pixel version of our model, we can now readily move on to GN-DPM. The only difference is that the appearance of a landmark is now represented by an $N_p = N_s \times N_s$ patch (descriptor) each pixel (element) of which can be seen as a 1-pixel appearance model for the corresponding landmark. Using the $\mathbf{A}^j$ representation defined in Section 3, the cost function to optimize for GN-DPMs is

[2] In practice, we never use one pixel but a patch and hence we compute gradients from a 3×3 neighborhood.

given by

$$\arg\min_{\Delta\mathbf{p},\Delta\mathbf{c}} \sum_{j=1}^{N_p} ||\mathbf{I}^j - \mathbf{A}^j(\mathbf{c}) - \mathbf{A}^j\Delta\mathbf{c} - \mathbf{J}^j\Delta\mathbf{p})||^2. \quad (10)$$

By re-arranging the terms above appropriately, it is not difficult to re-write (10) as in (7) where now the error term $\mathbf{I} - \mathbf{A}_0$ has size $N = uN_p$, $\mathbf{J}$ has size $N \times n$, and the solutions for $\Delta\mathbf{c}$ and $\Delta\mathbf{p}$ take the form of (8) and (9). The complexity of the exact and approximate versions is $O(nmuN_p + n^2uN_p)$ and $O(muN_p + n^2uN_p)$ respectively.

As in most works on deformable registration, our best performing implementation is based on robust descriptors. Our formulation can be readily extended to accommodate such a case. Assume that each pixel is described by a N_h-dimensional descriptor, and therefore each patch has now $N_p \times N_h$ elements. The cost function to optimize is readily given by

$$\arg\min_{\Delta\mathbf{p},\Delta\mathbf{c}} \sum_{j=1}^{N_p \times N_h} ||\mathbf{I}^j - \mathbf{A}^j(\mathbf{c}) - \mathbf{A}^j\Delta\mathbf{c} - \mathbf{J}^j\Delta\mathbf{p})||^2. \quad (11)$$

In particular, we describe each pixel with a reduced SIFT representation with $N_h = 8$ features computed over an 8×8 cell using the implementation provided in [21]. Finally, the complexity of the exact and approximate versions is $O(nmuN_pN_h + n^2uN_pN_h)$ and $O(muN_pN_h + n^2uN_pN_h)$, respectively.

4.3. Efficient weighted least-squares optimization of SIFT features

Although robust, one disadvantage inherent to the descriptor-based formulation of (11) is the increased computational complexity. Our experiments have shown that in this case GN-DPM is very robust but also quite slow. The main reason for this increased computational burden is the fact that a descriptor of size N_h is computed for every pixel resulting in a very dense representation. Prior work on object and face detection though (please see for example [7, 22]) have shown that almost as good performance can be achieved by computing a single descriptor for a $N_w \times N_w$ neighborhood. For example, for the HOG descriptor $N_w = 8$ and hence the size of the descriptor is less than the total number of pixels in the neighborhood used to compute the descriptor. In this section, we propose an approach which results in similar computational reduction but is quite different from the one used in object detection algorithms.

In particular, rather than creating a model based on sparsely computed descriptors as in [7, 22], we create a dense model (i.e. we use a descriptor for each pixel) as described in Section 3 but then we evaluate the cost functions of (10) or (11) on a sparse grid. In our case, this sparse grid is defined by an indicator function for each patch $\mathbf{W}_p$ of size $N_s \times N_s$ with elements $w_j = 1$ corresponding to the points that we wish to evaluate our cost function and $w_j = 0$ otherwise. Hence, our cost function in (10) (or in (11)) becomes

$$\arg\min_{\Delta\mathbf{p},\Delta\mathbf{c}} \sum_{j=1}^{N_p} w_j||\mathbf{I}^j - \mathbf{A}^j(\mathbf{c}) - \mathbf{A}^j\Delta\mathbf{c} - \mathbf{J}^j\Delta\mathbf{p})||^2. \quad (12)$$

It is not difficult to re-formulate (12) as a weighted least-squares problem

$$\arg\min_{\Delta\mathbf{p},\Delta\mathbf{c}} ||\mathbf{I} - \mathbf{A}(\mathbf{c}) - \mathbf{A}\Delta\mathbf{c} - \mathbf{J}\Delta\mathbf{p})||^2_{\mathbf{W}}, \quad (13)$$

where we have used the notation $||\mathbf{z}||^2_W = \mathbf{z}^T\mathbf{W}\mathbf{z}$ to denote the weighted ℓ_2 norm and $\mathbf{W}$ is a $N \times N$ diagonal matrix the elements of which are equal to 1 corresponding to the locations that we wish to evaluate our cost function and 0 otherwise.

The question of interest now is whether one can come up with closed-form solutions for $\Delta\mathbf{c}$ and $\Delta\mathbf{p}$, as in (8) and (9). Fortunately, the answer is positive. Let us define matrices $\mathbf{A}_w = \mathbf{W}\mathbf{A}$, $\mathbf{J}_{i,w} = \mathbf{W}\mathbf{J}_i$, $\mathbf{J}_w = \mathbf{J}_{0,w} + \sum_{i=1}^m c_i\mathbf{J}_{i,w}$, $\mathbf{P}_w = \mathbf{W} - \mathbf{A}_w(\mathbf{A}_w^T\mathbf{A}_w)^{-1}\mathbf{A}_w^T$. Then we can update $\Delta\mathbf{c}$ and $\Delta\mathbf{p}$ in alternating fashion from

$$\Delta\mathbf{c} = (\mathbf{A}_w^T\mathbf{A}_w)^{-1}\mathbf{A}_w^T(\mathbf{W}(\mathbf{I} - \mathbf{A}(\mathbf{c})) - \mathbf{J}_w\Delta\mathbf{p}) \quad (14)$$

$$\Delta\mathbf{p} = \mathbf{H}_{P_w}^{-1}\mathbf{J}_{P_w}^T(\mathbf{W}(\mathbf{I} - \mathbf{A}(\mathbf{c}))), \quad (15)$$

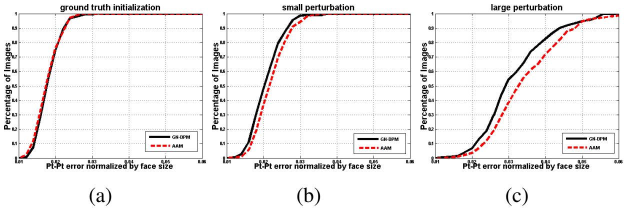
where $\mathbf{J}_{P_w} = \mathbf{P}_w\mathbf{J}_w$ and $\mathbf{H}_{P_w} = \mathbf{J}_{P_w}^T\mathbf{J}_{P_w}$, respectively. Finally, notice that in practice, we *never* calculate and store matrix multiplications of the form $\mathbf{W}\mathbf{X}$, for any matrix $\mathbf{X} \in \mathcal{R}^{N \times l}$. Essentially, the effect of this multiplication is a reduced size matrix of dimension $N_w \times l$, where N_w is the number of non-zero elements in $\mathbf{W}$. In our implementation we used a grid such that $N_w/N < 1/N_h$. Hence, in our SIFT-based GN-DPM, there are less features than the number of pixels in the original GN-DPM based on pixel-based parts. This version is very fast.

5. Comparison with AAMs

Two questions that naturally arise when comparing the part-based GN-DPMs over the holistic approach of AAMs [19] are: (a) do both models have the same representational power? and (b) which model is easier to optimize? Because it is difficult to meaningfully compare the representational power of the models directly, we provide in this section an attempt to shed some light on both questions by conducting an indirect comparison between the two models.

In particular, we trained both models on the same train set (the train set of LFPW), and then fitted both models on the same unseen test set (the test set of LFPW) [3]. For each

[3] We obtained very similar results by testing on Helen and AFW.



(a) (b) (c)

Figure 3. Comparison between GN-DPMs and AAMs [19]. Both algorithms were initialized using (a) the ground truth landmark locations, (b) the ground truth after small perturbation of the first shape parameter, and (c) the ground truth after large perturbation of the first shape parameter. The average (normalized) pt-pt Euclidean error Vs fraction of images is plotted.

method, we report the achieved fitting accuracy by plotting the familiar cumulative curve corresponding to the fraction of images for which the normalized error between the ground truth points and the fitted points was less than a specific value (please also see Section 6). To investigate question (a), we initialized both algorithms using the *ground truth* locations of the landmarks for each image. We assume that the more powerful the appearance model is, the better it will reconstruct the appearance of an unseen image, and hence the fitting process will not cause much drifting from the ground truth locations. Fig. 3 (a) shows the obtained cumulative curves for GN-DPMs and AAMs. We may see that both methods achieve literally the same fitting accuracy illustrating that the part-based and holistic approaches have the same representational power. An interesting observation is that the drift from ground truth is very small and the achieved fitting accuracy is at least as good as any state-of-the-art method in literature is able to produce. This shows that generative deformable models when trained in-the-wild are able to produce a very high degree of fitting accuracy.

To investigate question (b), we reconstructed the ground truth points from the shape model, perturbed the first shape parameter by some amount and then performed fitting using both algorithms. Fig. 3 (b) and (c) show the cumulative curves obtained by applying a small and a large amount of perturbation, respectively. Clearly, when the perturbation is large, GN-DPMs largely outperform AAMs. This shows that the part-based generative appearance model of GN-DPMs is easier to optimize.

6. Experiments

The main aim of this section is to present a comprehensive evaluation of the proposed GN-DPM formulation. We present results for four cases of interest, an overview of which follows below:

Case 1: GN-DPMs Vs AAMs. We further compare pixel-based GN-DPMs (GN-DPM-PI) and the Fast-SIC (also based on pixel intensities) AAM fitting approach of [19]. As we show below, the proposed GN-DPM-PI largely outperforms Fast-SIC, further validating the conclusions of Section 5 .

Case 2: Variants of GN-DPMs. We compare two variants of GN-DPMs based on SIFT features. The first is the full model which is built and fitted on a dense grid, using exact GN optimization. We call this variant GN-DPM-SIFT-Full. The second one is the model which is built on a dense grid but fitted on a sparse grid, using the approximate GN algorithm based on the Hessian approximation described in the last paragraphs of Section 4.1. We call this variant GN-DPM-SIFT. GN-DPM-SIFT is orders of magnitude faster than GN-DPM-SIFT-Full, nevertheless, as we show below, it performs as good as GN-DPM-SIFT-Full.

Case 3: GN-DPMs Vs SDM. SDM [21] is currently considered the state-of-the-art method in face alignment. As we show below, when trained on LFPW [2] and initialized in the same way, GN-DPMs outperform SDM (trained on thousands of images) sometimes by a large margin.

Case 4: GN-DPMs Vs Oracle. We compare GN-DPMs (as well all other methods considered in our experiments) against the best possible fitting result achieved by an Oracle who knows the location of the landmarks in the test images and simply reconstructs them using the trained shape model.

We trained all GN-DPMs on LFPW [2]. We used a patch of size of 27 × 27. To fit, we used a multi-resolution approach with two levels. At the highest level the shape model has 15 shape eigenvectors and 400 appearance eigenvectors. We tested on LFPW and additionally on Helen [10] and AFW [22] with the latter being two challenging out-of-database experiments. We created our models using the publicly available 68-point landmark configurations of [14, 18]. For initialization, we used the method of [22]. To measure performance, we used the point-to-point Euclidean distance (pt-pt error) normalized by the face size [22] and report the cumulative curve corresponding to the fraction of images for which the error was less than a specific value. As for the comparison with SDM, we note that we initialized SDM using the same face detector [22] (following the authors' instructions), and we report performance on the 49 interior points because these are the points that the publicly available implementation of SDM provides.

Fig. 4 shows our results on LFPW, Helen and AFW. Evaluation is based on all 68 points. We may observe that: (a) For all methods, the best performance is achieved on LFPW. There is a drop in performance for all methods on Helen and AFW because the faces of these databases are much more difficult to detect and fit. Nevertheless the relative difference in performance is similar. (b) GN-DPM-PI largely outperforms the AAM of [19] almost across the whole range of pt-pt error, i.e. it is significantly more robust and accurate. (c) There is a significant boost in performance when SIFT features are used, as expected. (d) The difference in performance between GN-DPM-SIFT and

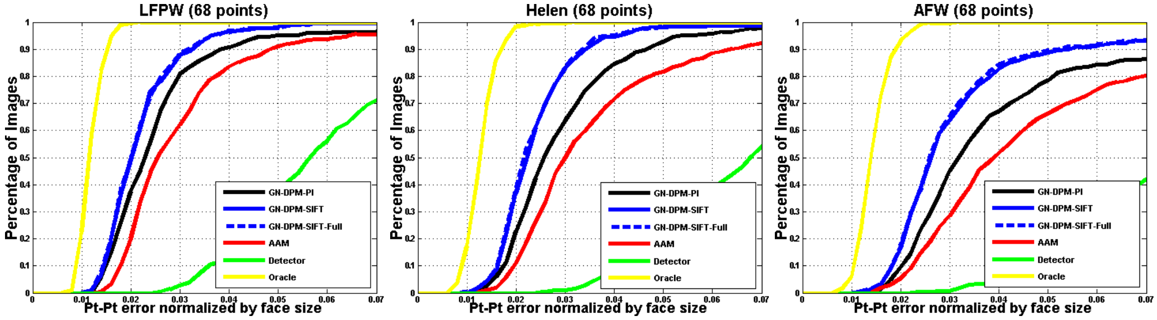


Figure 4. Average pt-pt Euclidean error (normalized by the face size) Vs fraction of images for LFPW, Helen and AFW. Evaluation is based on 68 points. The performance of different GN-DPMs variants and AAMs [19] is compared.

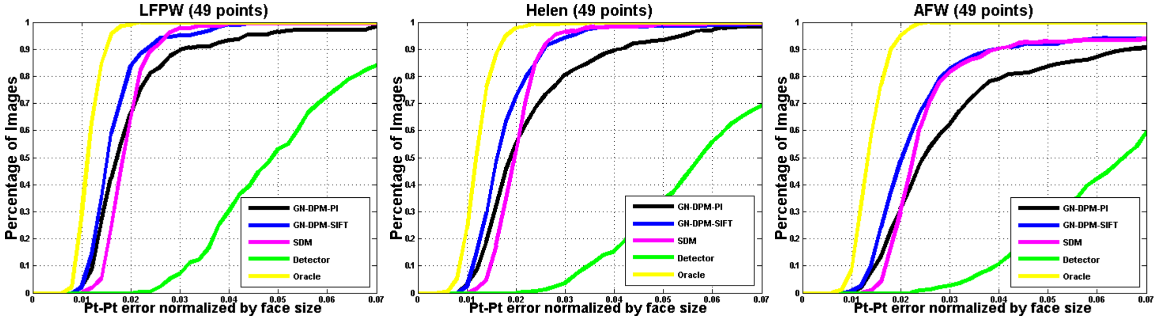


Figure 5. Average pt-pt Euclidean error (normalized by the face size) Vs fraction of images for LFPW, Helen and AFW. Evaluation is based on 49 points. The performance of GN-DPMs and SDM [21] is compared.

GN-DPM-SIFT-Full is negligible, although GN-DPM-SIFT is orders of magnitude faster. (d) There is a very large performance gap between GN-DPM-SIFT, which is the best performing method, and the best achievable result provided by the Oracle. Hence, we are still far away from considering face alignment in-the-wild a solved problem.

Fig. 5 shows our results for GN-DPM, GN-DPM-SIFT and SDM on LFPW, Helen and AFW. Evaluation is based on 49 points. We may observe that: (a) GN-DPM-SIFT outperforms SDM on all three databases and is significantly more accurate. (b) Interestingly, GN-DPM-PI (based on pixel intensities) performs better than SDM (based on SIFT features) for errors less than 0.02, that is it is more accurate, but worse than SDM for errors greater than 0.02, that is it is less robust.

Finally, representative fitting examples from LFPW and Helen can be seen in Fig. 6.

7. Conclusions

We introduced a DPM fitting strategy which jointly optimizes a global shape model and a part-based, trained in-the-wild, flexible appearance model, and thus by-passes a common limitation of most current DPM methods for face alignment. Our model results in a translational motion model which shifts parts so that a joint cost function of shape and appearance is minimized using efficient and robust Gauss-Newton optimization. Additionally, we showed that significant computational reductions can be achieved by building a full model during training but then evaluating the proposed cost function on a sparse grid using weighted least-squares during fitting. We coined the proposed formulation Gauss-Newton DPM. Finally, we conducted a number of experiments which showed that the proposed GN-DPM outperforms prior work sometimes by a large margin.

8. Acknowledgements

This work has been funded by the European Community 7th Framework Programme [FP7/2007-2013] under grant agreement no. 288235 (FROG).

References

[1] T. Baltrušaitis, P. Robinson, and L.-P. Morency. Constrained local neural fields for robust facial landmark detection in the wild. In *ICCV-W*, 2013.

[2] P. Belhumeur, D. Jacobs, D. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *CVPR*, 2011.

[3] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *CVPR*, 2012.

[4] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *TPAMI*, 23(6):681–685, 2001.

[5] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models-their training and application. *CVIU*, 61(1):38–59, 1995.

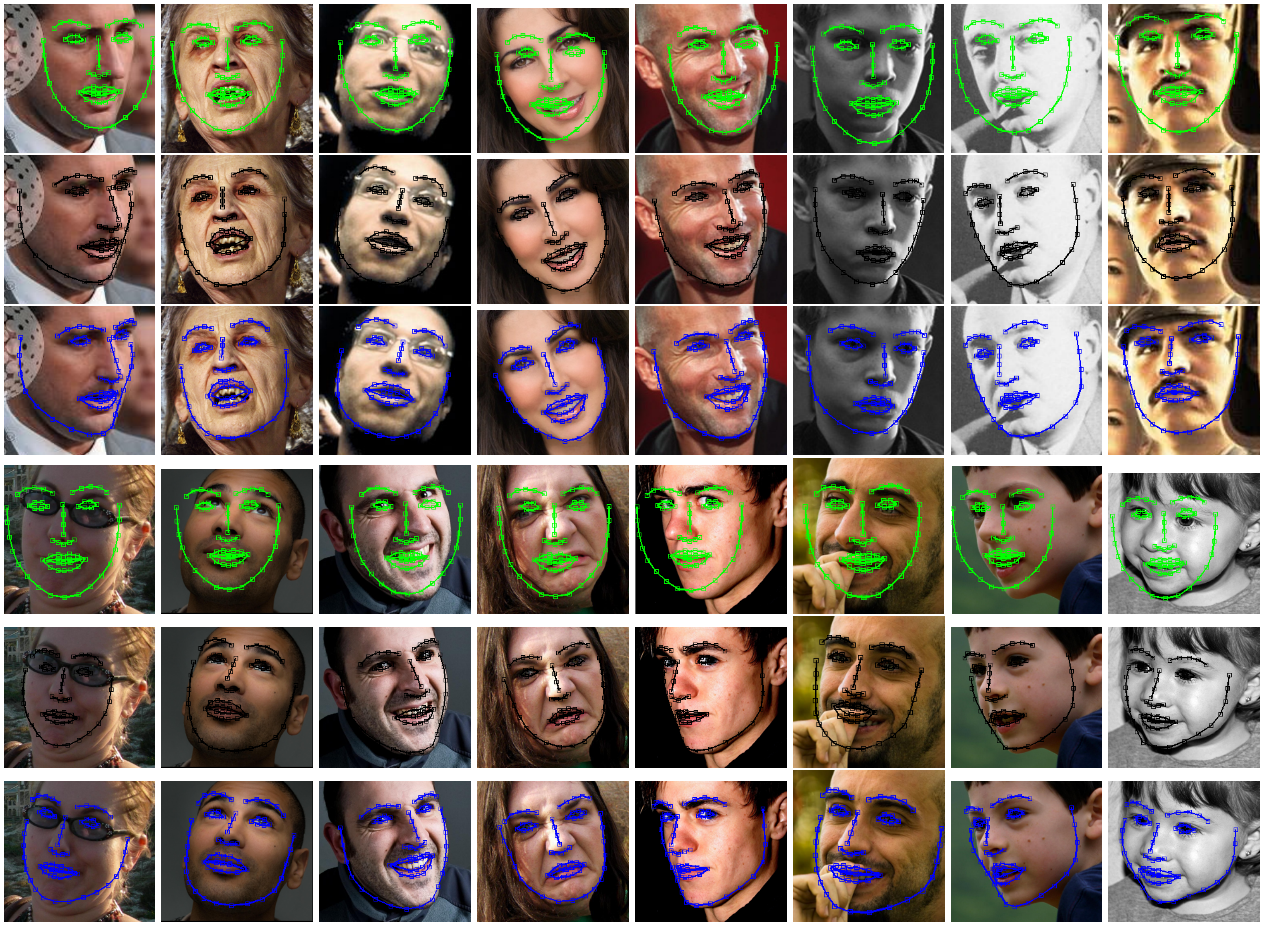


Figure 6. Fitting examples from LFPW and Helen. Green: Detector. Black: GN-DPM built from pixel intensities (GN-DPM-PI). Blue: GN-DPM built from SIFT features (GN-DPM-SIFT).

[6] D. Cristinacce and T. Cootes. Automatic feature localisation with constrained local models. *Pattern Recognition*, 41(10):3054–3067, 2008.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[8] L. Gu and T. Kanade. A generative shape regularization model for robust face alignment. In *ECCV*. 2008.

[9] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE TPAMI*, 20(10):1025–1039, 1998.

[10] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *ECCV*. 2012.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[12] P. Martins, R. Caseiro, J. F. Henriques, and J. Batista. Discriminative bayesian active shape models. In *ECCV*. 2012.

[13] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004.

[14] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *CVPR-W*, 2013.

[15] J. Saragih and R. Gocke. Learning aam fitting through simulation. *Pattern Recognition*, 42(11):2628–2636, 2009.

[16] J. Saragih and R. Goecke. A nonlinear discriminative approach to aam fitting. In *ICCV*, 2007.

[17] J. Saragih, S. Lucey, and J. Cohn. Deformable model fitting by regularized landmark mean-shift. *IJCV*, 91(2):200–215, 2011.

[18] G. Tzimiropoulos, J. Alabort-i Medina, S. Zafeiriou, and M. Pantic. Generic active appearance models revisited. In *ACCV 2012*. 2013.

[19] G. Tzimiropoulos and M. Pantic. Optimization problems for fast aam fitting in-the-wild. In *ICCV*, 2013.

[20] Y. Wang, S. Lucey, and J. Cohn. Enforcing convexity for improved alignment with constrained local models. In *CVPR*, 2008.

[21] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, 2013.

[22] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark estimation in the wild. In *CVPR*, 2012.