FAST NEWTON ACTIVE APPEARANCE MODELS

Jean Kossaifi[⋆] *Georgios Tzimiropoulos*[⋆†] *Maja Pantic*[⋆1]

[⋆] Imperial College London, UK, Department of Computing
[†]University of Lincoln, UK, Department of Computing
[1] University of Twente, The Netherlands

ABSTRACT

Active Appearance Models (AAMs) are statistical models of shape and appearance widely used in computer vision to detect landmarks on objects like faces. Fitting an AAM to a new image can be formulated as a non-linear least-squares problem which is typically solved using iterative methods. Owing to its efficiency, Gauss-Newton optimization has been the standard choice over more sophisticated approaches like Newton. In this paper, we show that the AAM problem has structure which can be used to solve efficiently the original Newton problem without any approximations. We then make connections to the original Gauss-Newton algorithm and study experimentally the effect of the additional terms introduced by the Newton formulation on both fitting accuracy and convergence. Based on our derivations, we also propose a combined Newton and Gauss-Newton method which achieves promising fitting and convergence performance. Our findings are validated on two challenging in-the-wild data sets.

Index Terms— Active Appearance Models, Newton method, LevenbergMarquardt, inverse compositional image alignment.

1. INTRODUCTION

Introduced in [1], Active Appearance Models (AAMs) are generative models of shape and appearance widely used in face and medical image modelling and landmark detection. As such, they have been extensively studied in computer vision research. Fitting an AAM to a new image can be formulated as a non-linear least-squares problem which is typically solved using iterative methods. There are mainly two lines of research for solving this problem: approximate methods like regression [2] or analytic gradient descent [2]. In this paper, we focus on the latter approach and the different ways of solving it.

Following the seminal work of [2], Gauss-Newton optimization has been the standard choice for optimizing AAMs. In [2], the authors proposed the so-called Project-Out Inverse Compositional algorithm (POIC). POIC decouples shape from appearance by projecting out appearance variation and computes a warp update in the model coordinate frame which is then composed to the current warp estimate. This results in a very fast algorithm which is the standard choice for fitting person specific AAMs. Its main disadvantage, though, is its limited generalization capability. In contrast to POIC, the Simultaneous Inverse Compositional (SIC) algorithm, proposed in [3], has been shown to perform robustly for the case of unseen variations [4]. However, the computational cost of the algorithm is almost prohibitive for most applications.

Fig. 1. Fitting examples taken from the LFPW dataset. Red: Gauss-Newton. Green: Pure Newton. Blue: Modified Levenberg-Marquardt.

Because of the increased computational complexity, to the best of our knowledge, no further attempts to study the performance of more sophisticated optimization techniques like Newton within AAMs have been made. However, as recently shown in [5], the cost of SIC can be significantly reduced without resorting to any approximations at all. Motivated by [5], we show that the Newton problem for the case of AAMs has structure and can be efficiently solved via block elimination which results in significant computational savings. Based on this observation and for the first time (to the best of our knowledge) in AAM literature, we derive the necessary equations for solving it. Additionally, we compare the derived equations to the ones derived from Gauss-Newton and illustrate which new terms are introduced by the Newton formulation. Then, we study their effect on fitting accuracy and speed of convergence. Finally, based on our findings, and

inspired by the Levenberg-Marquardt algorithm [6], we propose a combined Newton and Gauss-Newton method, which achieves promising fitting and convergence performance. Our findings are validated on two challenging in-the-wild data sets, namely LFPW [7] and Helen [8]. Illustrative examples for the methods presented in this paper are shown in Fig.1.

2. ACTIVE APPEARANCE MODELS

AAMs are characterized by shape, appearance and motion models. The shape model is obtained by firstly annotating the location of u landmarks across a training set of objects belonging to the same class (e.g. faces in our case). The annotated shapes are then normalized using Procrustes Analysis. This step removes variations due to translation, scaling and rotation. PCA is then applied to these normalized shapes and the first n shape eigenvectors $\{s_i, \cdots, s_n\}$ are kept to define the shape model along with the mean shape s_0. This model can be used to generate a shape $s \in \mathbb{R}^{2u}$ using $s = s_0 + \sum_{i=1}^{n} s_i q_i$, where $q \in \mathbb{R}^n$ is the vector of the shape parameters.

The appearance model is obtained from the texture of the training images, after appearance variation due to shape deformation is removed. This is achieved by warping each texture from its original shape into the mean shape s_0 using motion model W, which in this work is assumed to be a piecewise affine warp. Each shape-free texture is represented as a column vector of $\mathbb{R}^N$. Finally PCA is applied to all training shape-free textures to obtain the appearance model. This model can be used to generate a texture $a \in \mathbb{R}^N$ using $a = A_0 + \sum_{i=1}^{m} c_i A_i$, where $c \in \mathbb{R}^m$ is the vector of texture parameters. Finally, a model instance is synthesized to represent a test object by warping the texture instance a from the mean shape s_0 to the shape instance s using the piecewise affine warp W defined by s_0 and s. Please see [2] for more details on AAMs.

Localizing the landmarks of a face in a new image can be formulated as finding the shape and appearance parameters such that a model instance is "close" to the given image usually in a least-squares sense. This is equivalent to iteratively solving the following non-linear least-squares problem over all the pixels inside the mean shape (denoted by $v \in s_0$):

$$\arg\min_{q,c} \frac{1}{2} \sum_{v \in s_0} f(v,q,c) = \arg\min_{q,c} \frac{1}{2} \sum_{v \in s_0} g(v,q,c)^2, \quad (1)$$

where

$$g(v,q,c) = [A_0(v) + \sum_{i=1}^{m} c_i A_i(v) - I(W(v,q))].$$

Prior work on AAM fitting has mainly focused on solving the above problem using Gauss-Newton optimization. In particular, one can linearize the above cost function with respect to c and q, and then seek for updates, Δq and Δc, using least-squares. Notably, within the inverse compositional framework, the linearization with respect to q is performed on the model. To do so, we firstly write $A_i(v) = A_i(W(v, q = 0), i \in \{0, \cdots, m\}$. Then, to find an update, one proceeds as follows:

1. Linearize with respect to c. Also linearize the model $\{A_0, A\}$ around $q = 0$.

2. Compute updates, Δq and Δc, using least-squares.

3. Update c in an additive fashion, $c \leftarrow c + \Delta c$, and q in a compositional fashion $q \leftarrow q \circ \Delta q^{-1}$, where $\circ$ denotes the composition of two warps. Please see [2] for a principled way of applying the inverse composition to AAMs.

The above algorithm is known as the Simultaneous Inverse Compositional (SIC)[3], and it is the most popular *exact* Gauss-Newton algorithm for solving problem (1). One can show that the cost per iteration for SIC is $O((n+m)^2 N)$, and hence this algorithm is very slow [3]. Recently, the optimization problem for a fast but exact version of SIC was derived in [5]. The complexity of this algorithm is $O(nmN + n^2 N)$, only. Motivated by [5], in the next section, we develop a fast Newton algorithm for the efficient fitting of AAMs.

3. FAST NEWTON AAMS

The Newton method is an iterative method that works by approximating the objective function f with a quadratic function obtained from Taylor expansion. An update for the parameters is analytically found by setting the derivative of this approximation to zero. Newton's method writes $H_f \Delta r = -J_f^t$, where H_f and J_f are the Hessian and Jacobian matrices of f respectively, and $\Delta r = \{\Delta q, \Delta c\}$ is the update of the parameters. Although the cost of calculating the Hessian usually renders the Newton's algorithm computationally heavy and results in slow algorithms [6], in many cases, the problem at hand has structure which in turn can be used to provide computationally efficient solutions [9]. Fortunately, this is the case for the problem of AAM fitting. We take advantage of this structure to propose a computationally efficient Newton algorithm for fitting AAMs. To do so, let us decompose the problem as follows:

$$\begin{bmatrix} H_{qq} & H_{qc} \\ H_{cq} & H_{cc} \end{bmatrix} \begin{pmatrix} \Delta q \\ \Delta c \end{pmatrix} = \begin{pmatrix} -J_q^t \\ -J_c^t \end{pmatrix}, \quad (2)$$

with $H_{cc} = \frac{d^2 f}{dc^2} \in \mathbb{R}^{m,m}$, $H_{cq} = \frac{d^2 f}{dcdq} \in \mathbb{R}^{m,n}$, $H_{qc} = H_{cq}^t \in \mathbb{R}^{n,m}$, $H_{qq} = \frac{d^2 f}{dq^2} \in \mathbb{R}^{m,m}$, $J_q = \frac{df}{dq} \in \mathbb{R}^{1,n}$ and $J_c = \frac{df}{dc} \in \mathbb{R}^{1,m}$.

As we show below H_{cc} is the identity matrix, which in turn allows to efficiently update Δq and Δc in an alternating fashion by applying Schur's complement. In particular, by writing $(A_1(v), \ldots, A_m(v)) = A \in \mathbb{R}^{1,m}$ with $A^t A$ = Identity of $\mathbb{R}^{m,m}$, and $T = A_0 + \sum_{i=1}^{m} A_i c_i$, we have:

$$
\begin{aligned}
J_q &= \sum_v \nabla T(W(v,q)) \frac{dW}{dq} g(v,q,c) \\
J_c &= \sum_v A g(v,q,c) \\
H_{cc} &= \sum_v A^t A = \text{Identity of } \mathbb{R}^{m,m} \\
H_{qq}^{Newton} &= \sum_v \left(\tfrac{dW}{dq}\right)^t \left(\nabla^2 T(W(v,q))\right) \left(\tfrac{dW}{dq}\right) g(v,q,c) \\
&\quad + \nabla T(W(v,q)) \left(\frac{d^2 W}{d^2 q}\right) g(v,q,c) \\
H_{qq}^{GN} &= \sum_v \left(\nabla T(W(v,q)) \frac{dW}{dq}\right)^t \left(\nabla T(W(v,q)) \frac{dW}{dq}\right) \\
H_{qq} &= H_{qq}^{Newton} + H_{qq}^{GN} \\
H_{qc}^{Newton} &= \sum_v \left(\tfrac{dW}{dq}\right)^t \nabla A(W(v,q)) g(v,q,c) \\
H_{qc}^{GN} &= \sum_v \left(\nabla T(W(v,q)) \left(\tfrac{dW}{dq}\right)\right)^t A \\
H_{qc} &= H_{qc}^{Newton} + H_{qc}^{GN}.
\end{aligned}
$$

In the case of a piecewise affine warp, $\frac{d^2 W}{d^2 q} = 0$, hence the expression of H_{qq}^{Newton} simplifies to

$$
H_{qq}^{Newton} = \sum_{v \in s_0} \left(\tfrac{dW}{dq}\right)^t \left(\nabla^2 T(W(v,q))\right) \left(\tfrac{dW}{dq}\right) g(v,q,c).
$$

Using Schur's complement the following update rules are obtained:

$$
\begin{aligned}
\Delta q &= \left(H_{qq} - H_{qc} H_{cc}^{-1} H_{cq}\right)^{-1} \left(-J_q^t + H_{qc} H_{cc}^{-1} J_c^t\right), \\
\Delta c &= H_{cc}^{-1} \left(-J_c^t - H_{cq} \Delta q\right).
\end{aligned}
$$

Finally, after simplification, we derive the following update rules:

$$
\begin{aligned}
\Delta q &= \left(H_{qq} - H_{qc} H_{qc}^t\right)^{-1} \left(-J_q^t + H_{qc} J_c^t\right), \\
\Delta c &= \left(-J_c^t - H_{cq} \Delta q\right).
\end{aligned}
$$

Note that if we set $H_{qq} = H_{qq}^{GN}$ and $H_{qc} = H_{qc}^{GN}$, then we obtain the fast Gauss-Newton algorithm used in [5]. Hence, our main aim hereafter is to study the effect of the additional terms introduced by the Newton formulation on both fitting accuracy and convergence. Finally, we note that the cost of computing H_{qc}^{Newton} is $O(mnN)$ as $\left(\frac{dW}{dq}\right) \nabla A(W(v,q))$ can be pre-computed leaving only a dot product to do at each iteration while the computational cost of H_{qq}^{Newton} is simply $O(n^2 N)$.

4. COMBINING NEWTON AND GAUSS-NEWTON

As mentioned above, the main aim of our experiments was to investigate the performance of the additional terms (with respect to Gauss-Newton) introduced by the Newton formulation on both fitting accuracy and speed of convergence. In particular, the full Newton method uses $H_{qq} = H_{qq}^{Newton} + H_{qq}^{GN}$ and $H_{qc} = H_{qc}^{Newton} + H_{qc}^{GN}$, and hence the additional terms introduced by Newton's method are H_{qq}^{Newton} and H_{qc}^{Newton}. To investigate the performance of each additional term introduced by the Newton method, we set $H_{qq} = H_{qq}^{GN}$ and $H_{qc} = H_{qc}^{Newton} + H_{qc}^{GN}$, which we coin "Newton without H_{qq}". Similarly, we investigated the performance of the setting $H_{qq} = H_{qq}^{Newton} + H_{qq}^{GN}$ and $H_{qc} = H_{qc}^{GN}$, which we coin "Newton without H_{qc}".

Additionally, as we show below, the terms introduced by the Newton method, although in some cases add information, in some other cases, they tend to decrease performance. To prevent such cases, one can employ a Levenberg-Marquardt modification which puts more weight on the diagonal terms of the Hessian. We experimented with such an approach; however our experiments have shown that such a modification performed very similar to the original full Newton method. Hence, inspired by Levenberg-Marquardt's method [6], we opted to get the most of both methods by "adding only the required quantity of Newton". In particular, we set $H_{qq} = H_{qq}^{GN} + \gamma H_{qq}^{Newton}$ and $H_{qc} = H_{qc}^{GN} + \gamma H_{qc}^{Newton}$ and initialise $\gamma = 1$. At each step, if the error (please see next section for the definition of the error employed) decreases, we set $\gamma = \gamma \times 2$ if $\gamma < 1$, while if the error increases, we go back to the previous step and set $\gamma = \gamma/2$. Clearly, when $\gamma = 1$ the method reduces to pure Newton, whereas when $\gamma = 0$ the method reduces to Gauss-Newton. In the general case, our formulation incorporates the additional terms introduced by Newton's method only when necessary.

5. EXPERIMENTS

We tested the proposed algorithms on two very challenging data sets. For training, we used the training set of LFPW data set [7]. For testing, we used the test set of LFPW and also verified our findings on Helen [8]. For both data sets, we used the 68-point landmark annotations provided in [10]. In all cases, fitting was initialized by the face detector recently proposed in [11]. Finally, we fitted AAMs in two scales with 7 and 14 shape eigenvectors, and 50 and 400 texture eigenvectors, respectively.

We measured fitting accuracy by producing the familiar cumulative curve corresponding to the percentage of test images for which the error between the ground truth landmarks and the fitted shape was less than a specific value. As error metric, we used the point-to-point error normalized by the face size [11]. To measure speed of convergence, we considered that an algorithm converged when

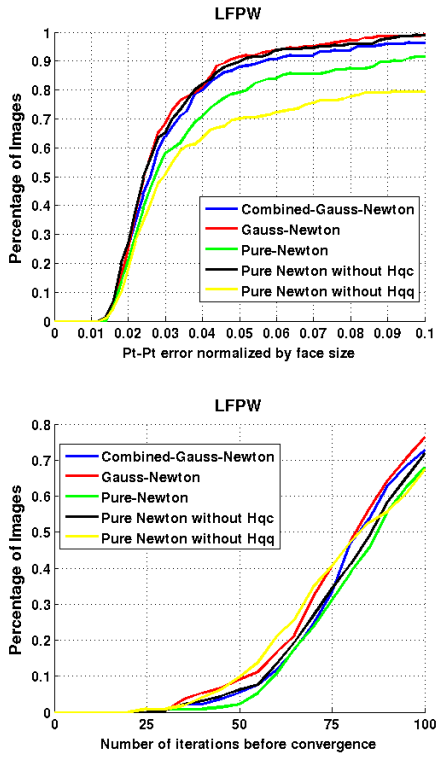

Fig. 2. Results on the LFPW dataset. Top: Average pt-pt Euclidean error (normalized by the face size) Vs fraction of images. Bottom: Convergence rate Vs fraction of images.

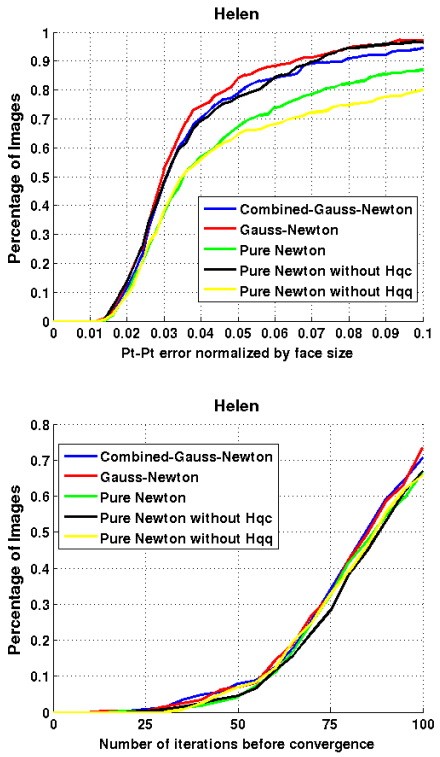

Fig. 3. Results on the Helen dataset. Top: Average pt-pt Euclidean error (normalized by the face size) Vs fraction of images. Bottom: Convergence rate Vs fraction of images.

$abs(\frac{error_k - error_{k+1}}{error_k}) < \epsilon$, with $error_k$ being the value of the objective function $(A_0 + \sum_{i=1}^{m} c_i A_i - I)^2$ at iteration k and ϵ being equal to $10e^{-5}$.

Fig. 2 shows the obtained results on LFPW. As we may observe the additional terms introduced by Newton have mixed positive and negative impact on performance. From Fig. 2 (a), we conclude that the full Newton method is not as accurate as Gauss-Newton in fitting performance; however Fig. 2 (b) shows that when converging to the "correct" solution, the H_{qc} term makes convergence faster. "Newton without H_{qq}" performs the worst in both fitting accuracy and convergence, and this result apparently comes from the term H_{qc}^{Newton} which makes the results worse when initialisation is bad. On the other hand, "Newton without H_{qc}" performs comparably to Gauss-Newton on fitting accuracy and slightly better on the speed of convergence, illustrating the importance of the H_{qq}^{Newton} term. Additionally, our Combined-Gauss-Newton method was able to perform the best among all Newton methods. Finally, from Fig. 3, we can draw similar conclusions for the Helen data set.

6. CONCLUSION AND FUTURE WORK

In this paper, we showed that the problem of AAM fitting via Newton method has structure that can be used to derive a computationally efficient solution. We then compared the derived solution to standard Gauss-Newton fitting. Overall, we found that the additional terms introduced by the Newton formulation have mixed positive and negative impact on performance. Finally, we showed that some of the negative sides can be remedied by combining Newton and Gauss-Newton in a Levenberg-Marquardt fashion.

It seems that the main problem with the Newton approach comes from the accumulated errors due to the piecewise affine warp and the second order gradients of the reconstructed appearance. We are therefore currently investigating a similar Newton method for the Gauss-Newton Deformable Part Model which by-passes the complicated motion model of AAMs [12]. Another future direction is to investigate performance for the case of robust features as in [13].

7. ACKNOWLEDGEMENTS

This work has been funded by the European Community 7th Framework Programme [FP7/2007-2013] under grant agreement no. 611153 (TERESA). The work of Georgios Tzimiropoulos is also funded in part by the European Community 7th Framework Programme [FP7/2007-2013] under grant agreement no. 288235 (FROG).

8. REFERENCES

[1] Timothy F Cootes, Gareth J Edwards, Christopher J Taylor, et al., "Active appearance models," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[2] Iain Matthews and Simon Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135 – 164, November 2004.

[3] S. Baker, R. Gross, and I. Matthews, "Lucas-kanade 20 years on: Part 3," *Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-03-35*, 2003.

[4] R. Gross, I. Matthews, and S. Baker, "Generic vs. person specific active appearance models," *Image and Vision Computing*, vol. 23, no. 12, pp. 1080–1093, 2005.

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

[6] Simon Baker and Iain Matthews, "Lucas-kanade 20 years on: A unifying framework," *IJCV*, vol. 56, no. 3, pp. 221 – 255, March 2004.

[7] Peter N. Belhumeur, David W. Jacobs, David J. Kriegman, and Neeraj Kumar, "Localizing parts of faces using a consensus of exemplars," in *CVPR*, June 2011.

[8] J. Brandt F. Zhou and Z. Lin, "Exemplar-based graph matching for robust facial landmark localization," in *ICCV*, 2013.

[9] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

[10] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic, "A semi-automatic methodology for facial landmark annotation," in *CVPR Workshops*, 2013.

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

[12] G. Tzimiropoulos and M. Pantic, "Gauss-newton deformable part models for face alignment in-the-wild," in *CVPR*, 2014.

[13] G. Tzimiropoulos, J. Alabort i medina, S. Zafeiriou, and M. Pantic, "Generic active appearance models revisited," in *ACCV*, November 2012.