A Nonlinear Discriminative Approach to AAM Fitting

Jason Saragih and Roland Goecke
Research School of Information Sciences and Engineering, Australian National University
Canberra, Australia
jason.saragih@rsise.anu.edu.au, roland.goecke@anu.edu.au

Abstract

The Active Appearance Model (AAM) is a powerful generative method for modeling and registering deformable visual objects. Most methods for AAM fitting utilize a linear parameter update model in an iterative framework. Despite its popularity, the scope of this approach is severely restricted, both in fitting accuracy and capture range, due to the simplicity of the linear update models used. In this paper, we present an new AAM fitting formulation, which utilizes a nonlinear update model. To motivate our approach, we compare its performance against two popular fitting methods on two publicly available face databases, in which this formulation boasts significant performance improvements.

1. Introduction

Since its advent by Edwards *et al*. [7], the Active Appearance Model (AAM) has been widely used to match deformable visual objects to images, with applications ranging from medical image analysis [21] to industrial vision problems [16]. The power of this generative model stems from a coupling of compact the representation of appearance, through the use of principle component analysis (PCA) on a set of labeled data, with a rapid fitting procedure.

Compared to other parametric models of deformable visual objects, the AAM is unique in its fitting regime. Here, the relationship between the generative texture error and the parameter updates around the optimum has been shown to be *close* to linear for simple visual objects, which exhibit small intrinsic shape and texture variations. In initial publications [5, 7], this was justified by arguing that since the error image is evaluated in the pose normalized frame, the error function around the true minimum is close to quadratic, allowing an iterative scheme of fixed linear updates with adaptive step sizes to converge.

From these initial publications, research into AAM fitting has diverged into two camps: discriminative and generative. To date, most research on discriminative methods, for example [6, 10, 11, 20], have retained the linearity assumption, focussing mainly on feature representations which adhere better to this assumption. Work on generative methods, for example [2, 4, 14], has focused on reformulations of the analytic optimization problem of AAM fitting to achieve more flexible and efficient linear update models. Some work has also been performed on image filtering, for example [12], to *smooth* the cost function, allowing the fitting process to better avoid local minima.

Although the use of nonlinear update models for parametric model fitting has been demonstrated in a number of recent publications, for example [1, 22, 24], there have been no implementations for AAM fitting as of yet. Perhaps the main reason for this is the storage requirements and evaluation cost of nonlinear functions. For example, the method in [24], which uses the relevance vector machine regressor, affords fast evaluation using raw image pixels since it uses a small image patch and fits only the affine parameters. The methods in [1] and [22] afford efficient evaluation due to their compact feature representation, the shape context descriptors, which are useful mainly for silhouette objects, and hence, not applicable to AAMs.

In this paper, we propose a formulation which allows a nonlinear update model to be implemented efficiently with the AAM. As the nonlinear function class, we use a boosted ensemble of multimodal Haar-like decision stumps, which allow efficient online evaluation using the integral image. To avoid overlearning, we embed the boosting procedure into an iterative framework with an intermediate resampling step. This process affords well regularized update models by limiting the ensemble size and indirectly increasing the sample size. In Section 2, we give a brief overview of the AAM's parameterization. A review of the state of the art in AAM fitting is then presented in Section 3, covering the discriminative and generative approaches. In Section 4, we present our nonlinear discriminative method for AAM fitting. To motivate our method, we compare its performance against two popular AAM fitting approaches on two publicly available databases in Section 5. We conclude in Section 6 with additional remarks and directions of future work.

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

2. Active Appearance Models

The AAM simultaneously models the intrinsic variation in shape and texture of a deformable visual objects as linear combination of basis modes of variation. The reason for this choice is the observation that intrinsic variations in shape and texture of many deformable visual objects follow that of a degenerate Gaussian distribution. As such the modes of variation can be easily calculated by applying PCA to a normalized training set. The result is a compact model, capable of generating large variations in shape and texture with a relatively small parameter set.

The shape of an AAM consists of a set of n 2D landmarks, generated as follows:

$$\mathscr{S}_l(\mathbf{p}_s) = \bar{\mathbf{s}} + \mathbf{B}_s\mathbf{p}_s, \tag{1}$$

where $\bar{\mathbf{s}}$ is the mean shape vector, $\mathbf{B}_s$ is the shape basis matrix with M_s columns obtained by applying PCA to a set of registered training shapes and $\mathbf{p}_s$ are the non-rigid shape parameters. To account for global shape variations, the AAM composes this non-rigid shape model with a similarity transform as follows:

$$\begin{aligned}\mathscr{S}(\mathbf{p}) &= \mathscr{S}_g(a,b,t_x,t_y) \circ \mathscr{S}_l(\mathbf{p}_s) \\ &= \left(\mathbf{I} \otimes \begin{bmatrix} a+1 & -b \\ b & a+1 \end{bmatrix}\right) \mathscr{S}_l(\mathbf{p}_s) + \mathbf{1} \otimes \begin{bmatrix} t_x \\ t_y \end{bmatrix},\end{aligned}$$

where $\mathbf{p}$ is a concatenation of all the parameters, (a, b, t_x, t_y) are similarity transform parameters, $\mathbf{I}$ is the $(n \times n)$ identity matrix and $\mathbf{1}$ is a n-length vector of ones. Here, $\otimes$ is the Kronecker (tiling) product and $\circ$ is the function composition operator.

The texture of an AAM is defined within the so called "shape free" frame. It consists of N pixels, usually chosen to lie within the convex hull of the mean shape $\bar{\mathbf{s}}$. As with the shape model, the texture is also generated using a linear combination of basis variation vectors:

$$\mathscr{T}_l(\mathbf{p}_t) = \bar{\mathbf{t}} + \mathbf{B}_t\mathbf{p}_t, \tag{2}$$

where $\bar{\mathbf{t}}$ is the vectorized mean image, $\mathbf{B}_t$ is the texture basis matrix with M_t columns obtained by applying PCA to a set of images, warped to the shape free frame and normalized, and $\mathbf{p}_t$ are the local texture parameters. To generate the texture in the image frame, the linear model is usually composed with a linear lighting function as follows:

$$\mathscr{T}(\mathbf{p}) = \mathscr{T}_g(\alpha,\beta) \circ \mathscr{T}_l(\mathbf{p}_t) = \alpha\mathscr{T}_l(\mathbf{p}_t) + \beta, \tag{3}$$

where $\mathbf{p}$ is a concatenation of all the parameters and (α, β) are the global lighting gain and bias parameters.

It is also common to take into account the correlation between shape and texture by applying a second level of PCA on the concatenated $\mathbf{p}_s$ and $\mathbf{p}_t$ parameters. This procedure usually yields a more compact representation. The interested reader is referred to [7] for details.

3. Review of AAM Fitting

AAM fitting is the process of finding the model parameters $\mathbf{p} = [\alpha, \beta, a, b, t_x, t_y, \mathbf{p}_s, \mathbf{p}_t]$ which best fit a given image $\mathscr{I}$. This is usually an iterative process which sequentially updates the model parameters $\mathbf{p}$ through an update function:

$$\Delta\mathbf{p} = \mathscr{U}(\mathbf{p}) \circ \mathscr{F}(\mathscr{I};\mathbf{p}). \tag{4}$$

Here, $\mathscr{U}$ is a vector valued update function, with optional dependence on the current parameters, and $\Delta\mathbf{p}$ are the updates to be applied to the current parameters, for example in an additive fashion:

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}. \tag{5}$$

$\mathscr{F}$ is a feature extraction function which represents the image $\mathscr{I}$ from the perspective of the AAM at parameter settings $\mathbf{p}$. A good coupling between $\mathscr{U}$ and $\mathscr{F}$ is required to ensure good predictions of the updates.

There are two general approaches to AAM fitting: discriminative and generative. In the following, we will briefly discuss each in turn.

3.1. Discriminative Fitting

Discriminative methods directly learn a fixed relationship between the features $\mathscr{F}(\mathscr{I};\mathbf{p})$ and the parameter updates $\Delta\mathbf{p}$, given a training set of perturbed model parameters:

$$\left\{\mathscr{F}(\mathscr{I};\mathbf{p}^* - \Delta\mathbf{p})\ ,\ \Delta\mathbf{p}\right\}_i^{N_d}, \tag{6}$$

where $\mathbf{p}^*$ is the optimal parameter setting for the i^{th} sample and N_d is the total number of perturbations in the training set. The advantage of this approach is that if $\mathscr{U}$ belongs to a simple function class, then the computation of the updates can be done efficiently, since $\mathscr{U}$ is fixed. Its main drawback is that the functional form of $\mathscr{U}$ and $\mathscr{F}$ that supports it, must be chosen heuristically.

The original AAM [7] was formulated based on the observation that, for some visual objects, the relationship between the residual texture:

$$\mathscr{F}(\mathscr{I};\mathbf{p}) = \mathscr{T}(\mathbf{p}) - \mathscr{I} \circ \mathscr{W}(\mathbf{p}) \tag{7}$$

and the parameter updates $\Delta\mathbf{p}$, where $\mathscr{W}$ is a warping function, is *close* to linear around the optimal parameter settings of a given image. As such, $\mathscr{U}$ can be easily found through linear regression on the data set in Equation (6). Since this linear relationship holds only loosely, fitting here is an iterative process which incorporates adaptive step size scaling:

$$\mathbf{p} \leftarrow \mathbf{p} + \eta\ \Delta\mathbf{p} \tag{8}$$

where η is progressively halved until a reduction in $\|\mathscr{F}(\mathscr{I};\mathbf{p})\|^2$ is attained.

From this basic approach, there have been a number of methods which boast improvements through a more suitable choice of $\mathscr{F}$. The direct appearance model method [10], for example, performs PCA on the residual vectors and performs linear regression between the principle components of the residual vector and the pose parameters. The method in [6] learns a linear regression between the canonical projections of the texture residuals and parameter updates. The method utilizes canonical correlation analysis to find the subspaces which best adheres to a linear regression. These methods have been shown to exhibit faster convergence and better accuracy compared to the original formulation in [7].

3.2. Generative Fitting

Generative methods pose fitting as minimizing some measure of error between the model's texture and the warped image. Essentially a nonlinear optimization problem, this approach affords the utilization of general purpose function optimizers, the convergence properties of which, are well understood. The most common measure of error utilized in AAM fitting is the least squares fit (or a robust variation thereof):

$$\sum_{\mathbf{x}\in\Omega} \left[\mathscr{T}(\mathbf{x};\mathbf{p}) - \mathscr{I}\circ\mathscr{W}(\mathbf{x};\mathbf{p})\right]^2, \tag{9}$$

where Ω is the domain over which the AAM's texture is defined. As such, the Gauss-Newton method, which results in a linear update model, has become the optimizer of choice for generative AAM fitting. However, a straight forward implementation is computationally expensive. As such, most generative approaches to AAM fitting either assume some parts of the method are fixed or reformulate the problem such that they are.

The original generative approach in [5], assumes that the Jacobian of Equation (9), is fixed. This allowed a fixed linear update model to be pre-computed through a pseudo-inverse of the fixed Jacobian. More recently, in a method coined adaptive AAM [4], the fixed Jacobian assumption is relaxed by decomposing the Jacobian and assuming only the component pertaining to the derivative of the warping function is fixed. The resulting method exhibited improved accuracy, however, as the linear update model depends on the current texture parameters, the fitting procedure is computationally expensive.

Another direction of the generative approach, which has gained much attention recently, is the adaptation of the inverse-compositional image alignment [3] to AAM fitting problems. By reversing the roles of the image and the model in the error function, the derivative of the warping function is fixed, resulting in significant computational savings. The project-out method [14], for example, minimizes the cost in a subspace orthogonal to the modes of texture variation, resulting in an analytically fixed linear update model over the shape parameters exclusively. Despite being the fastest method to date, it works well only for objects exhibiting small amounts of variation [9]. This problem is overcome by the simultaneous method [2], which solves for the shape and texture parameters simultaneously. However, similar to the method in [4], its update model depends on the current texture parameters, again resulting in a computationally expensive fitting procedure

4. Nonlinear Discriminative Updates

Current methods for AAM fitting, which utilize linear update models, require adaptive models to achieve high fidelity but prefer fixed models for efficiency reasons. In this section, we propose a nonlinear method where the trade-off between fidelity and efficiency can be better managed. Our method takes inspiration from the work in [25], but departs from it significantly in some parts, including the fitting procedure, formulation of the objective function and the type of learners used. The main components of this method are the iterative embedding, which encourages good generalization, and the use of multimodal weak learners, which afford efficient evaluation whilst maintaining good functional capacity.

4.1. Iterative Discriminative Learning

The aim of discriminative learning, here, is to find a nonlinear regressor from feature to parameter update space, given the training set in Equation (6). We propose learning the multivariate regressor through a boosting procedure, where the update function for the k^{th} parameter takes the following form:

$$\mathscr{U}^k(\mathbf{f}) = \sum_{t=1}^{N_f} \alpha_t^k \mathscr{L}_t^k(\mathbf{f}) \;\;;\;\; \mathscr{L}_t^k \in \mathcal{L}, \tag{10}$$

where

$$\mathbf{f} = \mathscr{F}(\mathscr{I};\mathbf{p}) = \mathscr{N}\circ\mathscr{I}\circ\mathscr{W}(\mathbf{p}) \tag{11}$$

is our feature vector of raw pixel values (normalized by $\mathscr{N}$ to mean zero and a variance of one) and $\mathscr{L}_t^k$ is a weak nonlinear learner, a number of which can combine to form a strong ensemble $\mathscr{U}^k$. Here, $\mathcal{L}$ is a *dictionary* of weak learners, the details of which will be discussed in Section 4.2. Starting with an empty ensemble, we add one weak learner at a time

$$\mathscr{U}_{t+1}^k = \mathscr{U}_t^k + \alpha_t^k \mathscr{L}_t^k, \tag{12}$$

choosing $(\alpha_t^k, \mathscr{L}_t^k)$ to maximally decrease the objective function for each addition. The final update model is a concatenation of the updates for every parameter:

$$\Delta\mathbf{p} = \mathscr{U}\circ\mathscr{F}(\mathscr{I};\mathbf{p}) = \left[\mathscr{U}^1(\mathbf{f});\ldots;\mathscr{U}^{N_p}(\mathbf{f})\right], \tag{13}$$

where N_p is the total number of parameters. In this work, since we are using the raw pixel feature in Equation (11), we estimate only the shape parameters $\mathbf{p} = [a, b, t_x, t_y, \mathbf{p}_s]$.

One of the main difficulties in boosting for regression is the tendency to overlearn the data. To overcome this problem we employ two measures. First, we perform shrinkage on the ensemble [8]. This common regularizing method involves shrinking the optimal α for the newly selected $\mathscr{L}$ by a factor $\eta \in [0, 1]$ before adding it to the ensemble. For the second measure, we note that overlearning in boosting for regression is related to the number of weak learners used [8]. Therefore, if we can reduce the total number of weak learners we can guard against overlearning. One of the peculiarities of AAM fitting compared to general image based regression methods is that the function defining the true regression become *simpler* the less spread the distribution of samples is about the optimum. This is because variations in pixel values becomes more constrained. Simpler functions generally require fewer weak learners to describe it.

Therefore, we propose embedding the boosting procedure into an iterative framework, where at each iteration, the boosting procedure needs only learn a function which can reduce the spread of the samples about the optimum. A similar training paradigm on the linear update model has been applied sucessfully in [19]. To this end, for the T^{th} learner to be added to an ensemble at a given iteration, we minimize the following objective for each AAM parameter:

$$\mathscr{C}(\alpha_T, \mathscr{L}_T) = \sum_{i=1}^{N_d} \frac{1}{\epsilon - |\Delta p_i - \sum_{t=1}^{T} \alpha_t \mathscr{L}_t(\mathbf{f}_i)|}, \quad (14)$$

subject to $\alpha_T \in [a, b]$ and $\mathscr{L}_T \in \mathcal{L}$, where the parameter index k has been dropped for clarity. Here,

$$a = \max\left(\frac{|\Delta\hat{p}_i| - \epsilon}{\mathscr{L}_T(\mathbf{f}_i)}\right) \quad , \quad b = \min\left(\frac{|\Delta\hat{p}_i| + \epsilon}{\mathscr{L}_T(\mathbf{f}_i)}\right) \quad (15)$$

and

$$\epsilon = \max|\Delta p_i| + \delta, \quad (16)$$

where δ is a small positive constant and

$$\Delta\hat{p}_i = \Delta p_i - \sum_{t=1}^{T-1} \alpha_t \mathscr{L}_t(\mathbf{f}_i). \quad (17)$$

is the current residual target updates after $T - 1$ learners have been added to the ensemble. This objective asymptotically penalizes the distance of each sample from the optimum, placing more emphasis on samples with large perturbations compared to, for example, the quadratic loss (see Figure 1). As each entry in the sum is convex, the objective of each round of boosting is also convex. Therefore, for a given $\mathscr{L}_T(\mathbf{f})$, the optimal α_T can be found through a 1D line search between a and b.

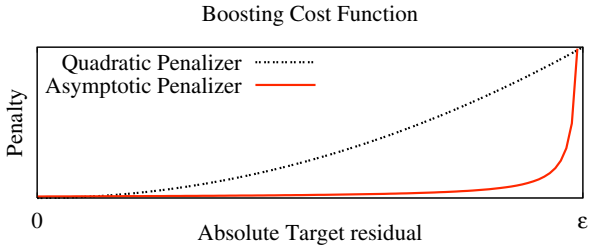


Figure 1. Cost function for every element in the sum of Equation (14). The asymptotic penalizer reduces spread by penalizing large residuals more severely.

After the capacity of $\mathcal{L}$ on the current iteration is exhausted or a fixed number of learners have been chosen, we resample a new set of perturbations, propagate them through previously learnt iterations and train the current iterations using this data. This resampling process further regularizes the solution as new iterations also correct predictions on samples which were poorly learnt previously due to overlearning. The complete training algorithm is outlined in Algorithm 1. Note that steps 10 to 20 apply to each AAM parameter independently.

Algorithm 1 Iterative Discriminative Training Algorithm

Require: N_i, N_d, N_s and N_f

1: **for** $i = 1$ to N_i **do**
2: $\{\Delta\mathbf{p}\}_{j=1}^{N_d}$ {sample perturbations}
3: **for** $l = 1$ to $i - 1$ **do**
4: **for** $j = 1$ to N_d **do**
5: $\mathbf{f}_j = \mathscr{N} \circ \mathscr{I}_j \circ \mathscr{W}(\mathbf{p}_j^* - \Delta\mathbf{p}_j)$ {get feature}
6: $\Delta\mathbf{p}_j \leftarrow \Delta\mathbf{p}_j - \mathscr{U}_l(\mathbf{f}_j)$ {propagate samples}
7: **end for**
8: **end for**
9: $\mathscr{U}_i = 0$ {initialize ensemble of i^{th} iteration}
10: **for** $t = 1$ to N_f **do**
11: $\alpha^* = 0$ and $\mathscr{L}^* = 0$
12: **for** $s = 1$ to N_s **do**
13: Sample $(\alpha, \mathscr{L})$ {see Algorithm 2}
14: **if** $\mathscr{C}(\alpha, \mathscr{L}) < \mathscr{C}(\alpha^*, \mathscr{L}^*)$ **then**
15: $(\alpha^*, \mathscr{L}^*) \leftarrow (\alpha, \mathscr{L})$
16: **end if**
17: **end for**
18: $\mathscr{U}_i \leftarrow \mathscr{U}_i + \alpha^* \mathscr{L}^*$ {Update i^{th} ensemble}
19: **end for**
20: **end for**
21: **return** $\mathscr{U}_1, \ldots, \mathscr{U}_{N_i}$

4.2. Weak Function Set

There are two requirements of the weak function set $\mathcal{L}$ in our method. Firstly, their evaluation must be computationally cheap, such that efficient fitting can be achieved with

a reasonably sized ensemble. Secondly, they must be sufficiently rich, such that complex regression functions can be accurately estimated by a linear combination of them. The Haar-like feature set $\mathcal{H}$, popularized by Viola and Jones in [23], act as a good basis for our weak function set as they fulfill both of the required criteria: efficient evaluation using the integral image and a capacity for complex representations through their similarity to Haar wavelets. In fact, in this work we utilize the extended Haar-like features [13] which include diagonal features, useful for approximating rotations.

A common choice of $\mathcal{L}$ which utilizes these features is the one-dimensional decision stump:

$$\mathscr{L}(\mathbf{f}) = \begin{cases} +1 & \text{if } s\mathscr{H}(\mathbf{f}) \geq s\theta \\ -1 & \text{otherwise} \end{cases}, \tag{18}$$

where $\mathscr{H}$ is a Haar-like filtering function, θ is a decision threshold and $s \in \{1, -1\}$ is a parity direction indicator. Although this weak function has been utilized in many works, for example [13, 23, 25], it has some major drawbacks. Firstly, most functions in this set are non-discriminative in the sense that, for a given $\mathscr{H}$, the best choice of s and θ will still result in a poor $\mathscr{L}$. Secondly, for those which are discriminative, the optimal choice of s and θ must be found through trial and error, an expensive process. This is especially potent in our case where the size of $\mathcal{H}$ is extremely large due to the size of the image region to be analyzed, which is around 5 to 10 times that of the images used in [23] and [25].

Rather than using the weak function set described above, we follow the response binning approach in [18], which maximizes the utility of weak learners derived from the Haar-like features. In their method, the weak learners of a classification problem, take the form of the relative inequality between histograms of the positive and negative examples:

$$\mathscr{L}(\mathbf{f}) = \begin{cases} +1 & \text{if } H_+(\mathscr{H}(\mathbf{f})) > H_-(\mathscr{H}(\mathbf{f})) \\ -1 & \text{otherwise} \end{cases} \tag{19}$$

where H_+ and H_- are 1D histograms of the distribution of the feature evaluations on the positive and negative examples respectively. This method affords a multimodal decision surface whilst maintaining efficiency as it requires only a table lookup for its evaluation.

To adapt this approach to the regression case, a few modifications need to be made. The objective function we want to minimize in Equation (14) aims to reduce the spread of the training data about the optimum. Therefore, in formulating $\mathscr{L}$, preference should be made on reducing the error over samples with large, compared to small, error. To this end, we define $H_{+/-}$ as the histogram of weighted samples with positive/negative target values:

$$H_+(v) = \sum_{\mathscr{H}(\mathbf{f}_i)=v} \frac{1}{\epsilon - \Delta\hat{p}_i} \;;\; \Delta\hat{p}_i > 0 \tag{20}$$

$$H_-(v) = \sum_{\mathscr{H}(\mathbf{f}_i)=v} \frac{1}{\epsilon + \Delta\hat{p}_i} \;;\; \Delta\hat{p}_i < 0, \tag{21}$$

where $\Delta\hat{p}_i$ is given in Equation (17). The idea here to build $\mathscr{L}$, such that the functional direction is in that which reduces the error over the most difficult samples in each bin, with the aim of reducing sample spread.

The only parameter which needs adjusting for this weak function set is the number of bins in the histograms n_b. Too many bins may cause overlearning in sparsely sampled bins, but too few bins may not capture enough of the nonlinearity of the target function, limiting the capacity of these learners. In our work, we fix n_b to an empirically good value and avoid overlearning by setting $\mathscr{L}$ at sparsely sampled bins to zero (i.e. avoid making decisions which are not well supported by the training data). A summary of the generation of our weak learner is given in Algorithm 2.

Algorithm 2 Weak Learner Sampling Algorithm

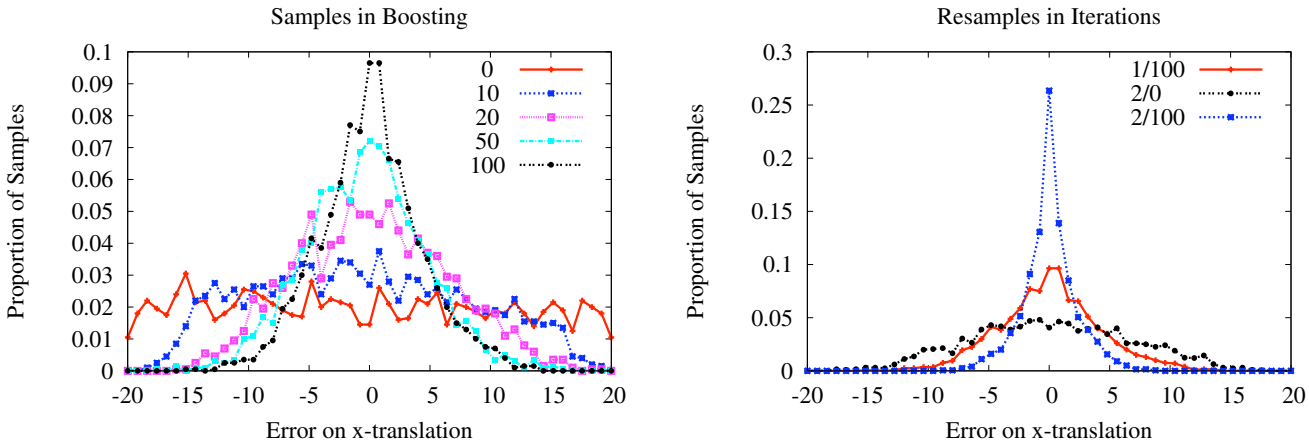
Require: $\{\mathbf{f}, \Delta\mathbf{p}\}_{j=1}^{N_d}$, $\mathcal{H}$ and n_b
1: Calculate weight of each sample: $(\epsilon - |\Delta p_i|)^{-1}$
2: Sample a Haar-like feature $\mathscr{H} \in \mathcal{H}$ {see [13]}
3: Build H_+ and H_- histograms {Eqn. (20) & (21)}
4: Compute weak learner $\mathscr{L}$ {Eqn. (19)}
5: Find optimal α through 1D line search {Eqn. (14)}
6: **return** $(\alpha, \mathscr{L})$

4.3. Nonlinear AAM Fitting

With a model trained according to Algorithm 1, the nonlinear fitting of an AAM follows the following procedure outlined in Algorithm 3. Notice that no checks need to be made regarding the reduction of texture error or the magnitude of parameter updates. The fitting is simply performed for all iterations for which the model is trained for with no early termination.

Algorithm 3 Nonlinear Discriminative AAM Fitting

Require: $\mathscr{I}, \{\mathscr{U}_1, \ldots, \mathscr{U}_{N_i}\}$ and $\mathbf{p}$
1: **for** $i = 1$ to N_i **do**
2: Get feature vector $\mathbf{f}$ {Eqn. (11)}
3: Calculate integral images from $\mathbf{f}$ {see [13]}
4: Calculate parameter updates using $\mathscr{U}_i$ {Eqn. (13)}
5: Update parameters {Eqn. (5)}
6: **end for**
7: **return** $\mathbf{p}$



(a) (b)

Figure 2. Distribution of the training samples of the IMM database throughout the training process on the x-translation. (a). Redistribution of samples about the optimum as weak learners are added to the ensemble of the first iteration. Legend denotes the number of weak learners in the ensemble. (b). The effect of resampling between iterations. Legend denotes (iteration)/(number of weak learners in ensemble of that iteration).

5. Experiments

To motivate the nonlinear update method, we compared its performance against two common methods for AAM fitting: the fixed Jacobian method [5] and the project-out inverse compositional method [14]. The fixed Jacobian method is generally considered a good baseline indicator for evaluating new fitting methods. The project-out method is currently the fastest method for AAM fitting, which we utilize to demonstrate the computational efficiency of our approach. The comparisons with these two methods were performed on two separate databases: the IMM Face database [17] and the XM2VTS database [15].

5.1. Databases

The IMM face database consists of 240 images of 40 individuals, exhibiting variations in pose, expression and lighting. We used 30 randomly selected subjects for training, leaving the rest for testing. Using the supplied 58-point markup, we built an appearance model of 28074 pixels, retaining 95% of shape and texture variation and 98% of combined appearance variation. The resulting model consisted of 19, 94 and 72 modes of shape, texture and combined appearance variation, respectively.

The XM2VTS database consists of 2360 frontal images of 295 subjects with large inter-subject variability. Half of the subjects were used for training and the others for testing. Using a publicly available 68-point markup [1], we built an appearance model of 46677 pixels, retaining the same amount of variation as in the IMM database. The resulting model consisted of 45, 328 and 262 modes of shape, texture and combined appearance variation, respectively.

[1] http://www.isbe.man.ac.uk/~bim/data/xm2vts/xm2vts_markup.html

5.2. Results

For the nonlinear method proposed in Section 4, we set N_i, N_f, N_d and N_s to be 10, 100, 2000 and 200 respectively (see Algorithm 1), chosen as a good trade-off between training time and model quality. For the weak learners we used 32 bin histograms with a threshold of 10 samples/bin. We used a shrinkage factor of $\eta = 0.5$ which we found to sufficiently regularize the solution.

Figure 2 shows the distribution of the IMM training samples at different stages of the training process. Plot (a) illustrates the capacity of the weak function set, described in Section 4.2, to significantly reduce the spread of the training samples despite the relative small value of N_s, which amounts to a very sparse sampling of $\mathcal{H}$. From Figure 2(b), it is clear that with the modest training set size used, the boosting process by itself significantly overlearns the data, as shown by the *spreading-out* of the resampled data in the next iteration. However, this artifact of the boosting process is more than compensated for in the next iteration, where the final distribution is even less spread than its predecessor. The other AAM parameters exhibit a similar trend.

To compare the nonlinear method against the fixed Jacobian and project-out methods, we randomly perturbed all AAM parameters from their optimal settings 100 times for every test image in both databases. The perturbations were taken randomly within $\pm 10°$, ± 0.1, ± 20 pixels, ± 0.1, ± 20 and ± 1.5 standard deviations for the rotation, scale, transla-

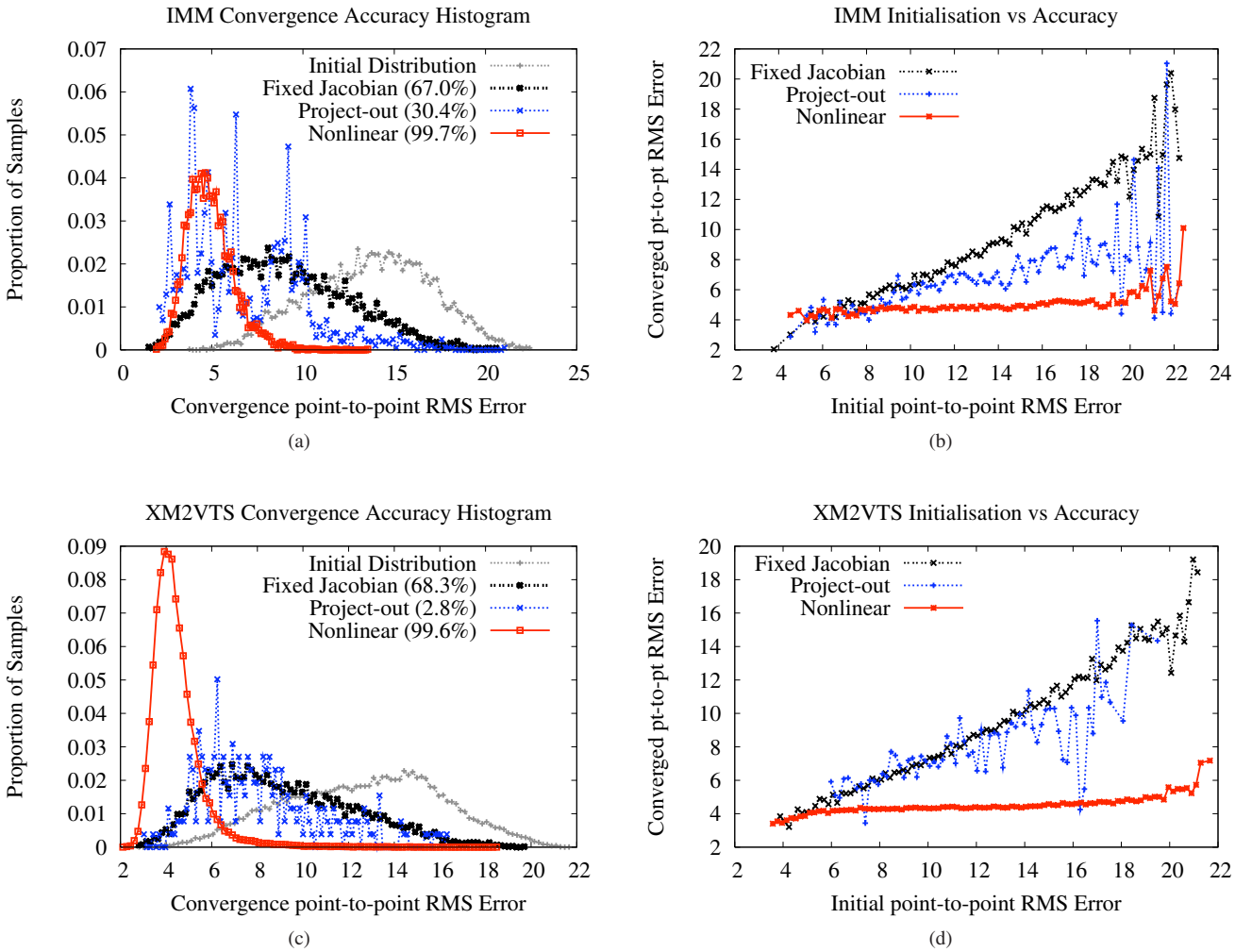


Figure 3. Convergence performance of the fixed Jacobian, project-out and nonlinear methods on the IMM and XM2VTS databases. Plots (a) and (c) : histograms of accuracy at convergence (percentage in legend is the convergence frequency). Plots (b) and (d) : initial error vs. average convergence accuracy.

tion, lighting gain, lighting bias and combined appearance parameters respectively. The fixed Jacobian and project-out methods were fitted using 3 levels of a Gaussian pyramid to avoid local minima, but the nonlinear method was fitted to the highest resolution image only, using the procedure in Algorithm 3.

Figure 3 shows the convergence performance of the three tested methods on both databases. Plots (a) and (c) show the accuracy at convergence of the different methods with their convergence frequencies given in the legends. Here, we declare convergence if the final point-to-point RMS error is smaller than at initialization. On both databases, the nonlinear method significantly outperforms the other methods, both in convergence accuracy and frequency. The project-out method does exhibit larger proportions of its converged samples at the lower end of the error range in the IMM database, but it also exhibits significant proportions at the larger error range. Furthermore, as the convergence rate of this method is extremely low, especially on the XM2VTS database, the small number of samples with good convergence can be attributed to images of subjects which are close to the mean texture, one of the conditions under which the project-out method works well.

Despite the large discrepancy in performance between the project-out and nonlinear methods, the computational complexity of both methods are similar, 376 ms for the project-out and 403 ms for the nonlinear method on a 1.8GHz machine [2]. With optimized code, we see no reason why the nonlinear method cannot achieve frame rate speeds (the project-out implementation in [14] is claimed to run at

[2]Reported processing times do not include the computation of the Gaussian pyramid for the project-out method.

230 frames/sec for a person specific AAM).

Apart from the fitting efficiency, perhaps one of the main strengths of the nonlinear method is illustrated through the results in Figure 3(b) and (d). Here, the average convergence accuracy is plotted against the point-to-point error at initialization. Whereas the average convergence accuracy of the project-out and fixed Jacobian methods deteriorate the further the initialization is from the optimum, due to the increasing likelihood of getting trapped in local minima, the nonlinear method maintains the accuracy of convergence until around 20 pixels point-to-point RMS initialization error.

6. Conclusion

In this paper, we have proposed a new nonlinear discriminative approach to AAM fitting which is fast, accurate and exhibits exceptionally good convergence rates with a large capture range. We compared the method against two popular fitting methods on two publicly available databases, on which the nonlinear method significantly outperformed them in both accuracy and convergence frequency. Through the use of multimodal weak learners, based on the Haar-like features, the method achieves good efficiency with reasonable training time, while regularization is maintained by embedding the training, and later the fitting procedure, in an iterative framework.

Future directions of research for this method include investigations into different types of features, for example the texture residuals, and weak learners, for example kernel functions on principle components of the residual vectors.

References

[1] A. Agarwal and B. Triggs. Recovering 3D Human Pose from Monocular Images. *PAMI*, 28(1), 2006.

[2] S. Baker, R. Gross, and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework: Part 3. Technical report, Robotics Institute, Carnegie Mellon University, 2003.

[3] S. Baker and I. Matthews. Equivalence and Efficiency of Image Alignment Algorithms. In *CVPR*, pages 1090–1097, 2001.

[4] A. Batur and M. Hayes. Adaptive Active Appearance Models. *IEEE Transactions on Image Processing*, 14(11):1707–1721, 2005.

[5] T. F. Cootes, G. Edwards, C. J. Taylor, H. Burkhardt, and B. Neuman. Active Appearance Models. In *ECCV*, volume 2, pages 484–489, 1998.

[6] R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof. Fast Active Appearance Model Search Using Canonical Correlation Analysis. *PAMI*, 28(10):1690–1694, 2006.

[7] G. Edwards, C. J. . Taylor, and T. F. Cootes. Interpreting Face Images Using Active Appearance Models. In *FG'98*, pages 300–305, 1998.

[8] J. H. Friedman. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

[9] R. Gross, I. Matthews, and S. Baker. Generic vs. Person Specific Active Appearance Models. *IVC*, 23(11):1080–1093, 2005.

[10] X. Hou, S. Li, H. Zhang, and Q. Cheng. Direct Appearance Models. In *CVPR*, volume 1, pages 828–833, 2001.

[11] P. Kittipanya-ngam and T. Cootes. The Effect of Texture Representations on AAM Performance. In *ICPR*, volume 2, pages 328–331, 2006.

[12] S. Le Gallou, G. Breton, C. Garcia, and R. Séguier. Distance Maps: A Robust Illumination Preprocessing for Active Appearance Models. In *VISAPP*, volume 2, pages 35–40, 2006.

[13] R. Lienhart and J. Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. In *ICIP*, pages 900–903, 2002.

[14] I. Matthews and S. Baker. Active Appearance Models Revisited. Technical report, Robotics Institute, Carnegie Mellon University, 2003.

[15] K. Messer, J. Matas, J. Kittler, J. Lüttin, and G. Maitre. Xm2vtsdb: The extended m2vts database. In *AVBPA*, pages 72–77, 1999.

[16] P. Mittrapiyanuruk, G. N. DeSouza, and A. C. Kak. Accurate 3D Tracking of Rigid Objects with Occlusion Using Active Appearance Models. In *WACV/MOTION*, pages 90–95, 2005.

[17] M. M. Nordstrøm, M. Larsen, J. Sierakowski, and M. B. Stegmann. The IMM Face Database - An Annotated Dataset of 240 Face Images. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, May 2004.

[18] B. Rasolzadeh, L. Petersson, and N. Pettersson. Response Binning: Improved Weak Classifiers for Boosting. In *IEEE Intelligent Vehicles Symposium*, pages 344–349, 2006.

[19] J. Saragih and R. Goecke. Iterative Error Bound Minimisation for AAM Alignment. In *ICPR*, volume 2, pages 1192–1195, 2006.

[20] M. B. Stegmann and R. Larsen. Multi-band Modelling of Appearance. *IVC*, 21(1):61–67, 2003.

[21] M. B. Stegmann and H. B. Larsson. Fast registration of Cardiac Perfusion MRI. In *International Society of Magnetic Resonance In Medicine*, page 702, Toronto, Canada, 2003.

[22] A. Thayananthan, R. Navaratnam, B. Stenger, P. Torr, and R. Cipolla. Multivariate Relevance Vector Machines for Tracking. In *ECCV*, pages 124–138, 2006.

[23] P. Viola and M. Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features. In *CVPR*, pages 511–518, 2001.

[24] O. Williams, A. Blake, and R. Cipolla. A Sparse Probabilistic Learning Algorithm for Real-time Tracking. In *ICCV*, pages 353–360, 2003.

[25] S. Zhou, B. Georgescu, X. S. Zhou, and D. Comaniciu. Image Based Regression Using Boosting Method. In *ICCV*, pages 541–548, 2005.