Fitting a Single Active Appearance Model Simultaneously to Multiple Images

Changbo Hu, Jing Xiao, Iain Matthews,
Simon Baker, Jeffrey F. Cohn, and Takeo Kanade
The Robotics Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213, USA

Abstract

Active Appearance Models (AAMs) are a well studied 2D deformable model. One recently proposed extension of AAMs to multiple images is the Coupled-View AAM. Coupled-View AAMs model the 2D shape and appearance of a face in two or more views simultaneously. The major limitation of Coupled-View AAMs, however, is that they are specific to a particular set of cameras, both in geometry and the photometric responses. In this paper, we describe how a single AAM can be fit to multiple images, captured simultaneously by cameras with arbitrary geometry and response functions. Our algorithm retains the major benefits of Coupled-View AAMs: the integration of information from multiple images into a single model, and improved fitting robustness.

1 Introduction

Active Appearance Models (AAMs) are a well studied model [3] which have a wide variety of applications, including face recognition, pose estimation, expression recognition, and lip-reading [7, 9].

One recently proposed extension of AAMs is the Coupled-View AAM (CVAAM) [4]. CVAAMs model the 2D shape and appearance of a face in two or more views simultaneously. The main motivation for CVAAMs is to take advantage of multiple cameras. For example, a CVAAM that is fit to a frontal and a profile view of a face integrates the data from the two images into a single set of model parameters. As shown in [5] (albeit using a slightly different technique), combining information from multiple images can improve face recognition performance.

The major limitation of CVAAMs is that they are specific to a particular set of cameras. A CVAAM can only be used with the same camera setup used to collect the training data. If a different number of cameras are used, the cameras are moved, or cameras with different photometric responses are used, the CVAAM cannot be re-used.

In this paper, we describe how a single AAM can be fit to multiple images, captured simultaneously by cameras with arbitrary geometry and response functions. Our algorithm removes the restriction on the cameras, but retains the major benefits of Coupled-View AAMs: the integration of information from multiple images into a single set of model parameters, and improved fitting robustness.

BMVC 2004 doi:10.5244/C.18.46

The main technical challenge is relating the 2D AAM shape parameters in one view with the corresponding parameters in the other views. This relationship is particularly complex because shape is modelled in 2D. It would be far easier if we had a 3D shape model. Fortunately, AAMs have recently been extended to 2D+3D AAMs [11]. A 2D+3D AAM contains *both* a 2D shape model and a 3D shape model. The 2D+3D AAM is fit using an extension of the inverse compositional algorithm [8]. As the algorithm runs, the 2D shape model is fit in the usual manner, but subject to the constraint that the 2D shape is a valid projection of the 3D shape model. The 2D+3D algorithm recovers the 2D shape parameters, the appearance parameters, the 3D shape parameters, and the camera parameters (which include 3D pose). Because it uses a fundamentally 2D fitting algorithm, the 2D+3D AAM can be fit very efficiently in real-time.

To generalise the 2D+3D fitting algorithm to multiple images, we use a separate set of 2D shape parameters for each image, but just a single, global set of 3D shape parameters. We impose the constraints that for each view separately, the 2D shape model for that view must approximately equal the projection of the single 3D shape model. Imposing these constraints indirectly couples the 2D shape parameters for each view in a physically consistent manner. Our algorithm can use any number of cameras, positioned arbitrarily. The cameras can be moved and replaced with different cameras without any retraining. The computational cost of the multi-view 2D+3D algorithm is only approximately N times more than the single-view algorithm where N is the number of cameras.

2 2D+3D Active Appearance Models

2.1 2D Active Appearance Models

The *2D shape* $\mathbf{s}$ of an AAM is a 2D triangulated mesh, and in particular the vertex locations of the mesh. AAMs allow linear shape variation. This means that the shape $\mathbf{s}$ can be expressed as a base shape $\mathbf{s}_0$ plus a linear combination of m shape matrices $\mathbf{s}_i$:

$$\mathbf{s} \;=\; \mathbf{s}_0 + \sum_{i=1}^{m} p_i \, \mathbf{s}_i \tag{1}$$

where the coefficients p_i are the shape parameters. AAMs are normally computed from training data consisting of a set of images with the shape mesh (hand) marked on them [3]. The Procrustes alignment algorithm and Principal Component Analysis (PCA) are then applied to compute the the base shape $\mathbf{s}_0$ and the shape variation $\mathbf{s}_i$.

The *appearance* of an AAM is defined within the base mesh $\mathbf{s}_0$. Let $\mathbf{s}_0$ also denote the set of pixels $\mathbf{u} = (u,v)^{\mathrm{T}}$ that lie inside the base mesh $\mathbf{s}_0$, a convenient abuse of terminology. The appearance of the AAM is then an image $A(\mathbf{u})$ defined over the pixels $\mathbf{u} \in \mathbf{s}_0$. AAMs allow linear appearance variation. This means that the appearance $A(\mathbf{u})$ can be expressed as a base appearance $A_0(\mathbf{u})$ plus a linear combination of l appearance images $A_i(\mathbf{u})$:

$$A(\mathbf{u}) \;=\; A_0(\mathbf{u}) + \sum_{i=1}^{l} \lambda_i A_i(\mathbf{u}) \tag{2}$$

where the coefficients λ_i are the appearance parameters. The base (mean) appearance A_0 and appearance images A_i are usually computed by applying Principal Components Analysis to the (shape normalised) training images [3].

Although Equations (1) and (2) describe the AAM shape and appearance variation, they do not describe how to generate a *model instance*. The AAM model instance with shape parameters $\mathbf{p}$ and appearance parameters λ_i is created by warping the appearance A from the base mesh $\mathbf{s}_0$ to the model shape mesh $\mathbf{s}$. In particular, the pair of meshes $\mathbf{s}_0$ and $\mathbf{s}$ define a piecewise affine warp from $\mathbf{s}_0$ to $\mathbf{s}$ denoted $\mathbf{W}(\mathbf{u};\mathbf{p})$.

Note that for ease of presentation we have omitted any mention of the 2D similarity transformation that is used with an AAM to normalise the shape [3]. In this paper we include the normalising warp in $\mathbf{W}(\mathbf{u};\mathbf{p})$ and the similarity normalisation parameters in $\mathbf{p}$. See [8] for a description of how to include the normalising warp in $\mathbf{W}(\mathbf{u};\mathbf{p})$ and also how to fit an AAM with such a normalising warp.

2.2 3D Active Appearance Models

The *3D shape* $\bar{\mathbf{s}}$ of a 3D AAM is a 3D triangulated mesh and in particular the vertex locations of the mesh. 3D AAMs also allow linear shape variation. The shape matrix $\bar{\mathbf{s}}$ can be expressed as a base shape $\bar{\mathbf{s}}_0$ plus a linear combination of $\overline{m}$ shape matrices $\bar{\mathbf{s}}_i$:

$$\bar{\mathbf{s}} \;=\; \bar{\mathbf{s}}_0 + \sum_{i=1}^{\overline{m}} \overline{p}_i\, \bar{\mathbf{s}}_i \tag{3}$$

where the coefficients $\overline{p}_i$ are the shape parameters. 3D AAMs are normally computed from training data consisting of a number of *3D range* images with the mesh vertices (hand) marked in them [2]. Note that there is no difference between the definition of a 3D AAM in this section and a 3D Morphable Model (3DMM) as described in [2].

The appearance of a 3D AAM is an image $A(\mathbf{u})$ just like the appearance of a 2D AAM. The appearance variation of a 3D AAM is also governed by Equation (2) and is computed in a similar manner by applying Principal Components Analysis to the input texture maps.

To generate a 3D AAM *model instance*, an image formation model is needed to convert the 3D shape $\bar{\mathbf{s}}$ into a 2D mesh, onto which the appearance is warped. In [10] the following weak perspective imaging model was used:

$$\mathbf{u} \;=\; \mathbf{P}\mathbf{x} \;=\; \begin{pmatrix} i_x & i_y & i_z \\ j_x & j_y & j_z \end{pmatrix} \mathbf{x} + \begin{pmatrix} o_x \\ o_y \end{pmatrix}. \tag{4}$$

where (o_x, o_y) is an offset to the origin and the projection axes $\mathbf{i} = (i_x, i_y, i_z)$ and $\mathbf{j} = (j_x, j_y, j_z)$ are equal length and orthogonal: $\mathbf{i}\cdot\mathbf{i} = \mathbf{j}\cdot\mathbf{j}$; $\mathbf{i}\cdot\mathbf{j} = 0$, and $\mathbf{x} = (x, y, z)$ is a 3D vertex location. The model instance is then computed by projecting every 3D shape vertex onto a 2D vertex using Equation (4). The appearance $A(\mathbf{u})$ is finally warped onto the 2D mesh (taking into account visibility.)

2.3 2D+3D Active Appearance Models

A 2D+3D AAM [11] consists of the 2D shape variation $\mathbf{s}_i$ of a 2D AAM governed by Equation (1), the appearance variation $A_i(\mathbf{u})$ of a 2D AAM governed by Equation (2), and the 3D shape variation $\bar{\mathbf{s}}_i$ of a 3D AAM governed by Equation (3). The 2D shape variation $\mathbf{s}_i$ and the appearance variation $A_i(\mathbf{u})$ of the 2D+3D AAM are constructed exactly as for a 2D AAM. The 3D shape variation $\bar{\mathbf{s}}_i$ we use is automatically constructed from the 2D shape variation $\mathbf{s}_i$ using a non-rigid structure-from-motion algorithm [11].

3 Fitting Algorithms

Our algorithm to fit a 2D+3D AAM to multiple images is an extension of the algorithm to fit a 2D+3D AAM to a single image [11] which itself is an extension to the algorithm to fit a 2D AAM to a single image [8]. To describe our algorithm, we first need to review the algorithms on which it is based.

3.1 Fitting a 2D AAM to a Single Image

The goal of fitting a 2D AAM to an image I [8] is to minimise:

$$\sum_{\mathbf{u}\in\mathbf{s}_0}\left[A_0(\mathbf{u})+\sum_{i=1}^{l}\lambda_i A_i(\mathbf{u})-I(\mathbf{W}(\mathbf{u};\mathbf{p}))\right]^2 = \left\|A_0(\mathbf{u})+\sum_{i=1}^{l}\lambda_i A_i(\mathbf{u})-I(\mathbf{W}(\mathbf{u};\mathbf{p}))\right\|^2 \quad (5)$$

with respect to the 2D shape $\mathbf{p}$ and appearance λ_i parameters. In [8] it was shown that the inverse compositional algorithm [1] can be used to optimise the expression in Equation (5). The algorithm uses the "project out" algorithm [6, 8] to break the optimisation into two steps. The first step consists of optimising:

$$\|A_0(\mathbf{u})-I(\mathbf{W}(\mathbf{u};\mathbf{p}))\|^2_{\text{span}(A_i)^\perp} \quad (6)$$

with respect to the shape parameters $\mathbf{p}$ where the subscript $\text{span}(A_i)^\perp$ means project the vector into the subspace orthogonal to the subspace spanned by A_i, $i=1,\ldots,l$. The second step consists of solving for the appearance parameters:

$$\lambda_i = -\sum_{\mathbf{u}\in\mathbf{s}_0} A_i(\mathbf{u})\,[A_0(\mathbf{u})-I(\mathbf{W}(\mathbf{u};\mathbf{p})] \quad (7)$$

where the appearance vectors A_i have been orthonormalised. Optimising Equation (6) itself can be performed by iterating the following two steps. Step 1 consists of computing:

$$\Delta\mathbf{p} = -H_{2\text{D}}^{-1}\Delta\mathbf{p}_{\text{SD}} \quad \text{where} \quad \Delta\mathbf{p}_{\text{SD}} = \sum_{\mathbf{u}\in\mathbf{s}_0}[\mathbf{SD}_{2\text{D}}(\mathbf{u})]^{\text{T}}\,[A_0(\mathbf{u})-I(\mathbf{W}(\mathbf{u};\mathbf{p})] \quad (8)$$

and the following two terms can be pre-computed to achieve high efficiency:

$$\mathbf{SD}_{2\text{D}}(\mathbf{u}) = \left[\nabla A_0\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]_{\text{span}(A_i)^\perp} \quad , \quad H_{2\text{D}} = \sum_{\mathbf{u}\in\mathbf{s}_0}[\mathbf{SD}_{2\text{D}}(\mathbf{u})]^{\text{T}}\,\mathbf{SD}_{2\text{D}}(\mathbf{u}). \quad (9)$$

Step 2 consists of updating the warp:

$$\mathbf{W}(\mathbf{u};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{u};\mathbf{p})\circ\mathbf{W}(\mathbf{u};\Delta\mathbf{p})^{-1}. \quad (10)$$

3.2 Fitting a 2D+3D AAM to a Single Image

The goal of fitting a 2D+3D AAM to an image I [11] is to minimise:

$$\left\|A_0(\mathbf{u})+\sum_{i=1}^{l}\lambda_i A_i(\mathbf{u})-I(\mathbf{W}(\mathbf{u};\mathbf{p}))\right\|^2 + K\cdot\left\|\mathbf{s}_0+\sum_{i=1}^{m}p_i\,\mathbf{s}_i-\mathbf{P}\left(\bar{\mathbf{s}}_0+\sum_{i=1}^{\overline{m}}\overline{p}_i\,\bar{\mathbf{s}}_i\right)\right\|^2 \quad (11)$$

with respect to $\mathbf{p}$, λ_i, $\mathbf{P}$, and $\overline{\mathbf{p}}$ where K is a large constant weight. Equation (11) should be interpreted as follows. The first term in Equation (11) is the 2D AAM fitting criterion. The second term enforces the (heavily weighted, soft) constraints that the 2D shape $\mathbf{s}$ equals the projection of the 3D shape $\overline{\mathbf{s}}$ with projection matrix $\mathbf{P}$. See Equation (4).

In [11] it was shown that the 2D AAM fitting algorithm [8] can be extended to a 2D+3D AAM. The resulting algorithm only requires approximately 20% more computation per iteration to process the second term in Equation (11). Empirically, however, the 3D constraints in the second term result in the algorithm requiring approximately 40% fewer iterations.

As with the 2D AAM algorithm, the "project out" algorithm [8] is used to break the optimisation into two steps, the first optimising:

$$\|A_0(\mathbf{u}) - I(\mathbf{W}(\mathbf{u};\mathbf{p}))\|^2_{\text{span}(A_i)^\perp} \; + \; K \cdot \sum_i F_i^2(\mathbf{p};\mathbf{P};\overline{\mathbf{p}}) \tag{12}$$

with respect to $\mathbf{p}$, $\mathbf{P}$, and $\overline{\mathbf{p}}$, where $F_i(\mathbf{p};\mathbf{P};\overline{\mathbf{p}})$ is the error inside the L2 norm in the second term in Equation (11) for each of the mesh x and y vertices. The second step solves for the appearance parameters using Equation (7). The 2D+3D has more unknowns to solve for than the 2D algorithm. As a notational convenience, concatenate all the unknowns into one vector $\mathbf{q} = (\mathbf{p};\mathbf{P};\overline{\mathbf{p}})$. Optimising Equation (12) is then performed by iterating the following two steps. Step 1 consists of computing[1]:

$$\Delta\mathbf{q} \; = \; -H_{3\text{D}}^{-1}\Delta\mathbf{q}_{\text{SD}} \; = \; -H_{3\text{D}}^{-1}\left[\begin{pmatrix} \Delta\mathbf{p}_{\text{SD}} \\ \mathbf{0} \end{pmatrix} + K \cdot \sum_i \left(\frac{\partial F_i}{\partial \mathbf{q}}\right)^{\mathbf{T}} F_i(\mathbf{q})\right] \tag{13}$$

where:

$$H_{3\text{D}} \; = \; \begin{pmatrix} H_{2\text{D}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + K \cdot \sum_i \left(\frac{\partial F_i}{\partial \mathbf{q}}\right)^{\mathbf{T}} \frac{\partial F_i}{\partial \mathbf{q}}. \tag{14}$$

Step 2 consists of first extracting the parameters $\mathbf{p}$, $\mathbf{P}$, and $\overline{\mathbf{p}}$ from $\mathbf{q}$, and then updating the warp using Equation (10), and the other two sets of parameters $\mathbf{P}$ and $\overline{\mathbf{p}}$ additively.

3.3 Fitting a Single 2D+3D AAM to Multiple Images

Suppose that we have N images $I^n : n = 1,\ldots,N$ of a face that we wish to fit the 2D+3D AAM to. We assume that the images are captured *simultaneously* by synchronised, but uncalibrated cameras. The naive algorithm is to fit the 2D+3D AAM *independently* to each of the images. This algorithm can be improved upon, however, by noticing that, since the images I^n are captured simultaneously, the 3D shape of the face should be the same whichever image it is computed in. We therefore pose fitting a single 2D+3D AAM to multiple images as minimising:

$$\sum_{n=1}^{N} \left(\left\| A_0(\mathbf{u}) + \sum_{i=1}^{l} \lambda_i^n A_i(\mathbf{u}) - I^n(\mathbf{W}(\mathbf{u};\mathbf{p}^n)) \right\|^2 + K \cdot \left\| \mathbf{s}_0 + \sum_{i=1}^{m} p_i^n \mathbf{s}_i - \mathbf{P}^n \left(\overline{\mathbf{s}}_0 + \sum_{i=1}^{\overline{m}} \overline{p}_i \overline{\mathbf{s}}_i \right) \right\|^2 \right) \tag{15}$$

[1]For ease of presentation, in this paper we omit any mention of the additional correction that needs to be made to $F_i(\mathbf{p};\mathbf{P};\overline{\mathbf{p}})$ to use the inverse compositional algorithm. See [11] for more details.

simultaneously with respect to the N sets of 2D shape parameters $\mathbf{p}^n$, the N sets of appearance parameters λ_i^n (the appearance may be different in different images due to different camera response functions), the N sets of camera matrices $\mathbf{P}^n$, and the one, global set of 3D shape parameters $\overline{\mathbf{p}}$. Note that the 2D shape parameters in each image are not independent, but are coupled in a physically consistent[2] manner through the single set of 3D shape parameters $\overline{\mathbf{p}}$. Optimising Equation (15) therefore cannot be decomposed into N independent optimisations. The appearance parameters λ_i^n can, however, be dealt with using the "project out" algorithm in the usual way; i.e. we first optimise:

$$\sum_{n=1}^{N} \left(\left\| A_0(\mathbf{u}) - I^n(\mathbf{W}(\mathbf{u};\mathbf{p}^n)) \right\|^2_{\mathrm{span}(A_i)^\perp} \; + \; K \cdot \left\| \mathbf{s}_0 + \sum_{i=1}^{m} p_i^n \mathbf{s}_i - \mathbf{P}^n \left(\overline{\mathbf{s}}_0 + \sum_{i=1}^{\overline{m}} \overline{p}_i \overline{\mathbf{s}}_i \right) \right\|^2 \right) \tag{16}$$

with respect to $\mathbf{p}^n$, $\mathbf{P}^n$, and $\overline{\mathbf{p}}$, and then solve for the appearance parameters:

$$\lambda_i^n \; = \; - \sum_{\mathbf{u} \in \mathbf{s}_0} A_i(\mathbf{u}) \cdot [A_0(\mathbf{u}) - I^n(\mathbf{W}(\mathbf{u};\mathbf{p}^n))] \, . \tag{17}$$

Organise the unknowns in Equation (16) into a single vector $\mathbf{r} = (\mathbf{p}^1;\mathbf{P}^1;\ldots;\mathbf{p}^N;\mathbf{P}^N;\overline{\mathbf{p}})$. Also, split the single-view 2D+3D AAM terms into parts that correspond to the 2D image parameters ($\mathbf{p}^n$ and $\mathbf{P}^n$) and the 3D shape parameters ($\overline{\mathbf{p}}$):

$$\Delta \mathbf{q}^n_{\mathrm{SD}} \; = \; \begin{pmatrix} \Delta \mathbf{q}^n_{\mathrm{SD,2D}} \\ \Delta \mathbf{q}^n_{\mathrm{SD},\overline{\mathbf{p}}} \end{pmatrix} \quad \text{and} \quad H^n_{\mathrm{3D}} \; = \; \begin{pmatrix} H^n_{\mathrm{3D,2D,2D}} & H^n_{\mathrm{3D,2D},\overline{\mathbf{p}}} \\ H^n_{\mathrm{3D},\overline{\mathbf{p}},\mathrm{2D}} & H^n_{\mathrm{3D},\overline{\mathbf{p}},\overline{\mathbf{p}}} \end{pmatrix} . \tag{18}$$

Optimising Equation (16) can then be performed by iterating the following two steps. Step 1 consists of computing:

$$\Delta \mathbf{r} \; = \; -H^{-1}_{\mathrm{MV}} \Delta \mathbf{r}_{\mathrm{SD}} \; = \; -H^{-1}_{\mathrm{MV}} \begin{pmatrix} \Delta \mathbf{q}^1_{\mathrm{SD,2D}} \\ \vdots \\ \Delta \mathbf{q}^N_{\mathrm{SD,2D}} \\ \sum_{n=1}^{N} \Delta \mathbf{q}^n_{\mathrm{SD},\overline{\mathbf{p}}} \end{pmatrix} \tag{19}$$

where:

$$H_{\mathrm{MV}} \; = \; \begin{pmatrix} H^1_{\mathrm{3D,2D,2D}} & \mathbf{0} & \cdots & \mathbf{0} & H^1_{\mathrm{3D,2D},\overline{\mathbf{p}}} \\ \mathbf{0} & H^2_{\mathrm{3D,2D,2D}} & \cdots & \mathbf{0} & H^2_{\mathrm{3D,2D},\overline{\mathbf{p}}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & H^N_{\mathrm{3D,2D,2D}} & H^N_{\mathrm{3D,2D},\overline{\mathbf{p}}} \\ H^1_{\mathrm{3D},\overline{\mathbf{p}},\mathrm{2D}} & H^2_{\mathrm{3D},\overline{\mathbf{p}},\mathrm{2D}} & \cdots & H^N_{\mathrm{3D},\overline{\mathbf{p}},\mathrm{2D}} & \sum_{n=1}^{N} H^n_{\mathrm{3D},\overline{\mathbf{p}},\overline{\mathbf{p}}} \end{pmatrix} . \tag{20}$$

Step 2 consists of extracting the parameters $\mathbf{p}^n$, $\mathbf{P}^n$, and $\overline{\mathbf{p}}$ from $\mathbf{r}$, and updating the warp parameters $\mathbf{p}^n$ using Equation (10), and the other parameters $\mathbf{P}^n$ and $\overline{\mathbf{p}}$ additively.

The N image algorithm is very similar to N copies of the single image algorithm. Almost all of the computation is just replicated N times, one copy for each image. The

[2] Note that directly coupling the 2D shape models would be difficult due to the complex relationship between the 2D shape in one image and another. Multi-view face model fitting is best achieved with a 3D model. A similar algorithm could be derived for 3D AAMs such as 3D Morphable Models [2]. The main advantage of using a 2D+3D AAM [11] is the far greater fitting speed.

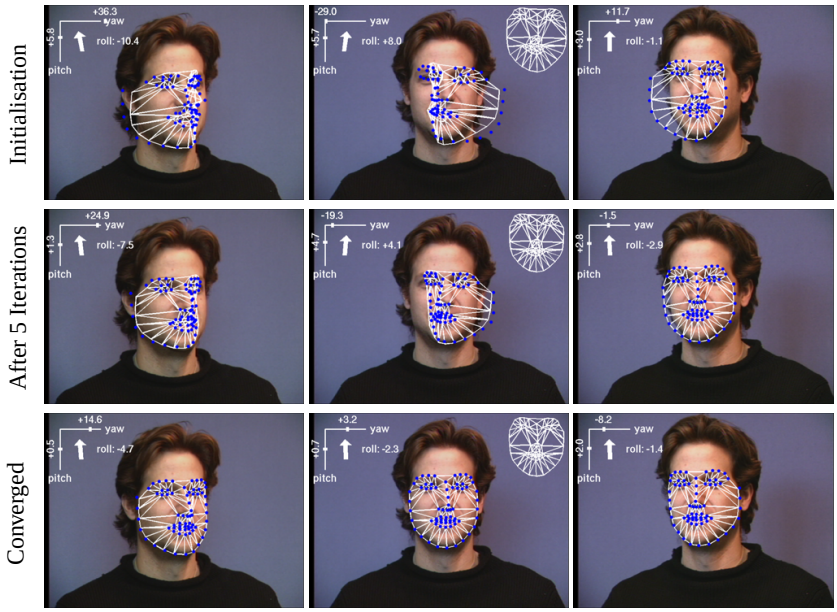


Figure 1: An example of using our algorithm to fit a single 2D+3D AAM to three images of a face. Each image is overlaid with the corresponding 2D shape for that image in blue dots. The head pose (extracted from the camera matrix $\mathbf{P}^N$) is displayed in the top left of each image as roll, pitch and yaw. The single 3D shape $\overline{\mathbf{p}}$ for the current 'frame' is displayed in the top right of the centre image. This 3D shape is also overlaid in each image, using the corresponding $\mathbf{P}^N$, as a white mesh.

only extra computation is adding the N terms in the components of $\Delta\mathbf{r}_{\text{SD}}$ and H_{MV} that correspond to the single set of global 3D shape parameters $\overline{\mathbf{p}}$, inverting the matrix H_{MV}, and the matrix multiply in Equation (19). Overall, the N image algorithm is therefore approximately N times slower than the single image 2D+3D fitting algorithm; i.e. the computational cost is almost identical to performing N independent 2D+3D AAM fits.

4 Experimental Results

An example of using our algorithm to fit a single 2D+3D AAM to three concurrent images of a face is shown in Figure 1. The initialisation is displayed in the top row of the figure, the result after 5 iterations in the middle row, and the final converged result in the bottom row. In each case, all three input images are overlaid with the 2D shape $\mathbf{p}^n$ plotted in blue dots. We also display the recovered pose angles (roll, pitch and yaw) extracted from the weak perspective camera matrix $\mathbf{P}^n$ for each view in the top left of the image. Each camera computes a different relative head pose, illustrating that the estimate of $\mathbf{P}^n$ is view dependent. The single 3D shape $\overline{\mathbf{p}}$ for all views at the current iteration is displayed in the top-right of the centre image. The view-dependent camera projection of this 3D shape is also plotted as a white mesh overlaid on the face.

Applying the multi-view fitting algorithm sequentially to a set of concurrent frames allows us to track the face simultaneously in N video sequences. Some example frames

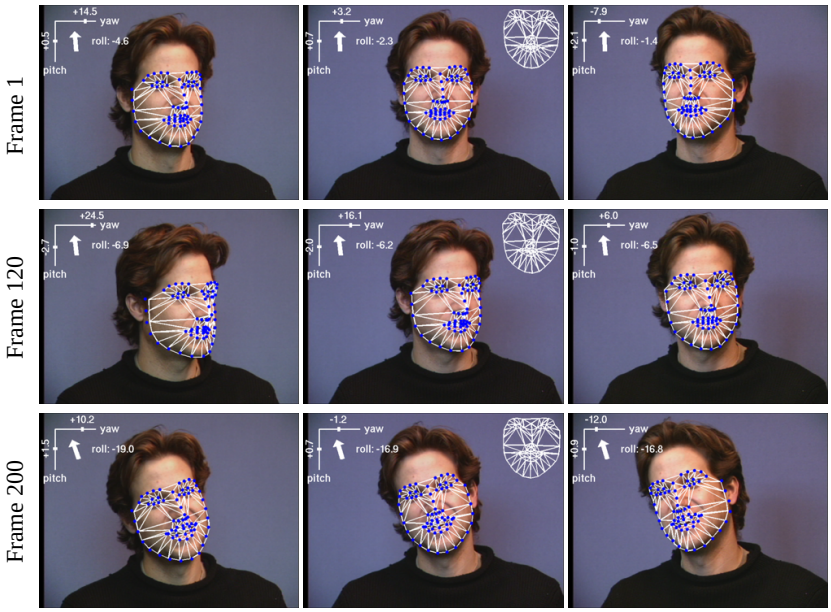
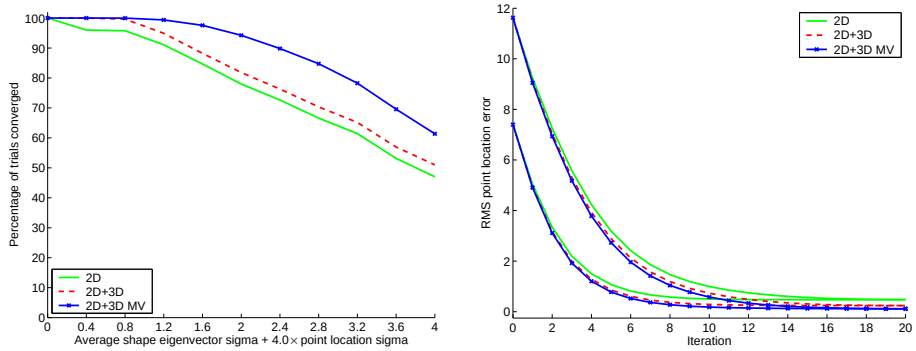


Figure 2: An example of our multi-view fitting algorithm being using to track a face in a trinocular stereo sequence.

of the algorithm being using to track a face in a trinocular stereo sequence is shown in Figure 2. The tracking remains accurate and stable both over time and between views.

In Figure 3 we show quantitative results to demonstrate the increased robustness and convergence rate of our multi-view fitting algorithm. In experiments similar to those described in [8], we generated a large number of test cases by randomly perturbing from a ground-truth obtained by tracking the sequence. For this experiment the 2D parameters of each view were randomly perturbed independently from the other views (but all views were perturbed each time). The 3D initial parameters were computed from the perturbed 2D mesh coordinates. We then run each algorithm from the same perturbed starting point and determine whether they converge by comparing the RMS error between the 2D mesh location of the fit result and the ground-truth 2D mesh coordinates. The algorithm is considered to have converged if this error is less than 1 pixel. We repeat the experiment 20 times for each set of 3 images and average over all 300 image triples in the test sequence. This is repeated for different values of perturbation energy to determine the robustness and convergence properties of each algorithm. The magnitude of the perturbation is chosen to vary on average from 0 to 4 times the 2D shape standard deviation. The 2D similarity parameters are perturbed to introduce a spatial offset of 4 times this value. The multiplier is not a critical value, it simply introduces significant similarity perturbation [8].

In Figure 3(a) we plot a graph of the likelihood (frequency) of convergence against the magnitude of the random perturbation for the 2D algorithm, the 2D+3D algorithm, and the new multi-view 2D+3D algorithm for the trinocular 300×3 frame sequence shown in Figure 2. The results clearly show that the multi-view algorithm is more robust than the single-view 2D+3D algorithm [11], which itself is more robust than the original



(a) Frequency of Convergence (b) Rate of Convergence

Figure 3: (a) The likelihood (frequency) of convergence plot against the magnitude of a random perturbation to the ground-truth fitting results computed by tracking through the trinocular sequence in Figure 2. The results show that the multi-view 2D+3D algorithm is more robust than the single-view 2D+3D algorithm, which itself is more robust than the 2D algorithm. (b) The rate of convergence is estimated by plotting the average error after each iteration against the iteration number. The results show that the multi-view 2D+3D algorithm converges slightly faster than the single-view 2D+3D algorithm, which converges slightly faster than the 2D algorithm.

2D algorithm [8]. We also compute how fast the algorithms converge by computing the average RMS mesh location error after each iteration. Only trials that actually converge are used in this computation. The results are included in Figure 3(b) and show that the multi-view algorithm converges slightly faster than the single-view 2D+3D algorithm, which converges slightly faster than the 2D algorithm.

5 Discussion

We have described an algorithm to fit a single 2D+3D AAM to N concurrent images captured simultaneously by N uncalibrated cameras. In the process, our algorithm computes: 2D shape parameters for each image, a single set of global 3D shape parameters, the weak-perspective camera matrix for each view (3D pose), and appearance parameters for each image (which may be different due to different camera response functions.) Our algorithm enforces the constraints that all of these quantities are physically consistent in the 3D scene. The algorithm operates approximately N times slower than the real-time (over 60fps) single image 2D+3D AAM fitting algorithm [11]. We have shown our multi-view 2D+3D AAM algorithm to be both more robust and converge more quickly than the single-view 2D+3D AAM algorithm, which is itself more robust than the single-view 2D AAM algorithm [8].

Because our fitting algorithm computes the camera matrices, it can also be used to calibrate the cameras (as weak perspective). Once they are calibrated, it may be possible to reformulate the fitting problem with less unknowns, and hopefully achieve even greater robustness. We plan to investigate this in a future paper. Other areas for future work include fitting 2D+3D AAMs simultaneously over time. The difficulty with a temporal sequence is that the face can deform non-rigidly. Note that fitting a single AAM to an entire sequence is one large global optimisation into which we can add global temporal

smoothness constraints which may improve fitting performance, whereas tracking a head through a video is a sequence of independent optimisations that may yield inconsistent results at each frame. The best way to do this is an interesting research question.

Acknowledgments

The research described in this paper was supported by U.S. DoD contract N41756-03-C4024, NIMH grant R01 MH51435, and DENSO Corporation.

References

[1] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221 – 255, 2004.

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Computer Graphics, Annual Conference Series*, pages 187–194, 1999.

[3] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[4] T. Cootes, G. Wheeler, K. Walker, and C. Taylor. Coupled-view active appearance models. In *Proceedings of the British Machine Vision Conference*, volume 1, pages 52–61, 2000.

[5] R. Gross, I. Matthews, and S. Baker. Appearance-based face recognition and light-fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4):449 – 465, 2004.

[6] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1025–1039, 1998.

[7] A. Lanitis, C. J. Taylor, and T. F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):742–756, 1997.

[8] I. Matthews and S. Baker. Active Appearance Models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004. In Press. Also appeared as Carnegie Mellon University Robotics Institute Technical Report CMU-RI-TR-03-02.

[9] I. Matthews, T. Cootes, J. Bangham, S. Cox, and R. Harvey. Extraction of visual features for lipreading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):198–213, 2002.

[10] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3D morphable model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 59–66, 2003.

[11] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2D+3D active appearance models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 535–542, 2004.