Multi-View AAM Fitting and Camera Calibration

Seth Koterba, Simon Baker, Iain Matthews,
Changbo Hu, Jing Xiao, Jeffrey Cohn, and Takeo Kanade
The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213
{skoterba, simonb, iainm, changbo, jxiao, jeffcohn, tk}@cs.cmu.edu

Abstract

In this paper we study the relationship between multi-view Active Appearance Model (AAM) fitting and camera calibration. In the first part of the paper we propose an algorithm to calibrate the relative orientation of a set of $N > 1$ cameras by fitting an AAM to sets of N images. In essence, we use the human face as a (non-rigid) calibration grid. Our algorithm calibrates a set of 2×3 weak-perspective camera projection matrices, projections of the world coordinate system origin into the images, depths of the world coordinate system origin, and focal lengths. We demonstrate that the performance of this algorithm is comparable to a standard algorithm using a calibration grid. In the second part of the paper we show how calibrating the cameras improves the performance of multi-view AAM fitting.

1. Introduction

Model-based face analysis is a general paradigm with numerous applications. A face model is typically constructed from either a set of training images [3] or a set of range images [2]. The face model is then fit to the input image(s) and the model parameters are used in whatever the application is. For example, in [9], the same model was used for face recognition, pose estimation, and expression recognition.

Perhaps the most well known face models are 2D Active Appearance Models (AAMs) [3] and 3D Morphable Models (3DMMs) [2]. More recently, [12] introduced 2D+3D Active Appearance Models, a model with the real-time fitting speed of 2D AAMs and the 3D modeling of 3DMMs.

Face models are usually fit to a single image of a face. In many application scenarios, however, it is possible to set up two or more cameras and acquire multiple views of the face. If we integrate the information in the multiple views, we can possibly obtain better application performance [5].

Although both multi-view face models [4] and algorithms to fit a single view model simultaneously to multiple views [2, 8] have been proposed, little work has been performed on the role of camera calibration in face model fitting. In this paper we study the relationship between face model fitting and camera calibration. Camera calibration can be divided into two sub-topics: (1) extrinsic calibration: computing the relative orientations of a set of $N > 1$ cameras, and (2) intrinsic calibration: computing the focal length, principal point, radial distortion, etc, of each camera. Although we do calibrate a focal length for each camera, the main goal of this paper is extrinsic calibration; i.e. the computation of the relative orientations of N cameras.

In the first part of this paper we propose an algorithm to calibrate the relative orientations of a set of N cameras using multi-view AAM fitting. In essence, we use the human face as a (non-rigid) calibration grid. Such an algorithm may be useful in a surveillance setting where we wish to install the cameras on the fly, but avoid walking around the scene with a calibration grid. We use the weak perspective camera model used by most 3D face modeling papers [11, 12]. Our algorithm calibrates the 2×3 camera projection matrices, the focal lengths, the projections of the world coordinate system origin into the images, and the depths of the world coordinate system origin. Our algorithm is an extension of the multi-view AAM fitting algorithm proposed by Hu *et al.* in [8]. The algorithm requires at least two sets of multi-view images of the face at two different locations. More images can be used to improve the accuracy if they are available. We evaluate our algorithm by comparing it with an algorithm that uses a calibration grid and show the performance to be comparable.

In the second part of the paper we show how camera calibration can improve the performance of face model fitting. We present a multi-view AAM fitting algorithm that takes advantage of calibrated cameras. We demonstrate that our algorithm (also an extension of [8]) results in far better fitting performance than either single-view fitting [12] or uncalibrated multi-view fitting [8]. We consider two performance measures: (1) the robustness of fitting - the likelihood of convergence for a given magnitude perturbation from the ground-truth, and (2) speed of fitting - the average number

of iterations required to converge from a given magnitude perturbation from the ground-truth.

We begin with a brief review of 2D Active Appearance Models (AAMs) [3], 3D Morphable Models (3DMMs) [2], 2D+3D AAMs [12], and the efficient inverse compositional algorithms to fit 2D AAMs [10] and 2D+3D AAMs [12]. In Section 3 we describe our calibration algorithm. Our algorithm uses 2D+3D AAMs [12], however any 3D face model could be used instead. In Section 4 we describe and evaluate our calibrated multi-view AAM fitting algorithm.

2. Background

2.1. 2D Active Appearance Models

The *2D shape* $\mathbf{s}$ of a 2D AAM [3] is a 2D triangulated mesh. In particular, $\mathbf{s}$ is a column vector containing the vertex locations of the mesh. AAMs allow linear shape variation. This means that the 2D shape $\mathbf{s}$ can be expressed as a base shape $\mathbf{s}_0$ plus a linear combination of m shape vectors $\mathbf{s}_i$:

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

where the coefficients $\mathbf{p} = (p_1, \ldots, p_m)^{\mathrm{T}}$ 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 base shape $\mathbf{s}_0$ and the shape variation $\mathbf{s}_i$.

The *appearance* of a 2D 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 normalized) training images [3].

Although Equations (1) and (2) describe the shape and appearance variation, they do not describe how to generate a *model instance.* The AAM instance with shape parameters $\mathbf{p}$ and appearance parameters λ_i is created by warping the appearance A from the base mesh $\mathbf{s}_0$ onto 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 normalize the shape [3]. In this paper we include the normalizing warp in $\mathbf{W}(\mathbf{u};\mathbf{p})$ and the similarity normalization parameters in $\mathbf{p}$. See [10] for the details of how to do this.

2.2. 3D Morphable Models

The *3D shape* $\overline{\mathbf{s}}$ of a 3DMM [2] is a 3D triangulated mesh. In particular, $\overline{\mathbf{s}}$ is a column vector containing the vertex locations of the mesh. 3DMMs also allow linear shape variation. The 3D shape vector $\overline{\mathbf{s}}$ can be expressed as a base shape $\overline{\mathbf{s}}_0$ plus a linear combination of $\overline{m}$ shape vectors $\overline{\mathbf{s}}_i$:

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

where the coefficients $\overline{p}_i$ are the shape parameters. 3DMMs are normally computed from training data consisting of a set of *3D range* images with the mesh vertices located in them [2]. Note that what we have described as a 3D Morphable Model can also be regarded as a 3D AAM.

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

To generate a 3DMM *model instance*, an image formation model is needed to convert the 3D shape $\overline{\mathbf{s}}$ into a 2D mesh, onto which the appearance is warped. In [11] the following *scaled orthographic* imaging model was used:

$$\mathbf{u} \;=\; \mathbf{P}_{\mathrm{so}}(\mathbf{x}) \;=\; \sigma \begin{pmatrix} i_x & i_y & i_z \\ j_x & j_y & j_z \end{pmatrix} \mathbf{x} + \begin{pmatrix} o_u \\ o_v \end{pmatrix}. \tag{4}$$

where (o_u, o_v) is an offset to the origin, the projection axes $\mathbf{i} = (i_x, i_y, i_z)$ and $\mathbf{j} = (j_x, j_y, j_z)$ are orthonormal ($\mathbf{i}\cdot\mathbf{i} = \mathbf{j}\cdot\mathbf{j} = 1, \mathbf{i}\cdot\mathbf{j} = 0$), and σ is the scale. The 3DMM instance is computed by first projecting every 3D shape vertex $\mathbf{x} = (x, y, z)^{\mathrm{T}}$ onto a 2D vertex $\mathbf{u}$ using Equation (4). The appearance $A(\mathbf{u})$ is then warped onto the 2D mesh (taking into account visibility) to generate the final model instance.

2.3. 2D+3D Active Appearance Models

A 2D+3D AAM [12] 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 $\overline{\mathbf{s}}_i$ of a 3DMM 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 $\overline{\mathbf{s}}_i$ is constructed from the 2D shape variation $\mathbf{s}_i$ and a collection of tracking data using non-rigid structure-from-motion [13].

2.4. Efficient Fitting Algorithms

We now review the efficient inverse compositional algorithms to fit 2D AAMs [10] and 2D+3D AAMs [12]. The goal of fitting a 2D AAM to an image I [10] is to minimize:

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

with respect to the 2D shape $\mathbf{p}$ and appearance λ_i parameters. In [10] it was shown that the "project out" algorithm [6] can be used to break the optimization into two steps. The first step consists of optimizing:

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

with respect to the shape parameters $\mathbf{p}$ where the subscript $\mathrm{span}(A_i)^\perp$ 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 using the closed form expression:

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

assuming that the appearance vectors A_i have been orthonormalized. In [10] it was also shown that the inverse compositional algorithm [1] can be used to optimize the expression in Equation (6). The final algorithm operates at over 200 frames-per-second on a standard 3.4GHz PC.

The goal of 2D+3D AAM fitting [12] is to minimize:

$$\left\|A_0(\mathbf{u})+\sum_{i=1}^{l}\lambda_i A_i(\mathbf{u})-I(\mathbf{W}(\mathbf{u};\mathbf{p}))\right\|^2$$
$$+K\cdot\left\|\mathbf{s}_0+\sum_{i=1}^{m}p_i\,\mathbf{s}_i-\mathbf{P}_{\mathrm{so}}\left(\overline{\mathbf{s}}_0+\sum_{i=1}^{\overline{m}}\overline{p}_i\,\overline{\mathbf{s}}_i\right)\right\|^2 \quad (8)$$

with respect to $\mathbf{p}$, λ_i, $\mathbf{P}_{\mathrm{so}}$, and $\overline{\mathbf{p}}$ where K is a large constant weight. Equation (8) should be interpreted as follows. The first term in Equation (8) 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 scaled orthographic projection $\mathbf{P}_{\mathrm{so}}$. Note that in the optimization, it is the component parameters of $\mathbf{P}_{\mathrm{so}}$ (σ, $\mathbf{i}$, $\mathbf{j}$, o_u, and o_v) that are optimized. See Equation (4).

In [12] it was shown that the 2D AAM fitting algorithm [10] can be extended to fit a 2D+3D AAM. The "project out" algorithm can also be used on Equation (8). The resulting algorithm requires slightly more computation per iteration to process the second term in Equation (8). The final algorithm still operates comfortably in real-time, at around 60Hz on a standard 3.4GHz PC.

3. Camera Calibration

3.1. Image Formation Model

The scaled orthographic image formation model in Equation (4) is sufficient when working with either a single camera or multiple cameras capturing a single image. When working with multiple cameras capturing multiple images, it is better to use the *weak perspective* model:

$$\mathbf{u} = \mathbf{P}_{\mathrm{wp}}(\mathbf{x}) = \frac{f}{o_z+\overline{z}}\begin{pmatrix} i_x & i_y & i_z \\ j_x & j_y & j_z \end{pmatrix}\mathbf{x}+\begin{pmatrix} o_u \\ o_v \end{pmatrix} \quad (9)$$

because Equation (9) models how the scale σ in Equation (4) varies from image to image in terms of the focal length f and average depth of the scene $o_z+\overline{z}$. In this last expression, o_z is the depth of the origin of the world coordinate system and $\overline{z}$ is the average depth of the scene points measured relative to the world coordinate origin. The "z" (depth) direction is $\mathbf{k}=\mathbf{i}\times\mathbf{j}$ where $\times$ is the vector cross product, $\mathbf{i}=(i_x,i_y,i_z)$, and $\mathbf{j}=(j_x,j_y,j_z)$. The average depth relative to the world origin $\overline{z}$ equals the average value of $\mathbf{k}\cdot\mathbf{x}$ computed over all points $\mathbf{x}$ in the scene.

3.2. Camera Calibration Goal

Suppose we have N cameras $n=1,\ldots,N$. The goal of our camera calibration algorithm is to compute the 2×3 camera projection matrix ($\mathbf{i}$, $\mathbf{j}$), the focal length f, the projection of the world coordinate system origin into the image (o_u, o_v), and the depth of the world coordinate system origin (o_z) for each camera. If we superscript the camera parameters with n we need to compute $\mathbf{P}^n_{\mathrm{wp}}=\mathbf{i}^n$, $\mathbf{j}^n$, f^n, o^n_u, o^n_v, and o^n_z. There are 7 unknowns in $\mathbf{P}^n_{\mathrm{wp}}$ (rather than 10) because there are only 3 degrees of freedom in choosing the 2×3 camera projection matrix ($\mathbf{i}$, $\mathbf{j}$) such that it is orthonormal.

3.3. Uncalibrated Multi-View Fitting

Suppose we have N images $I^n : n=1,\ldots,N$ captured by the N cameras $\mathbf{P}^n : n=1,\ldots,N$. We assume that the images are captured *simultaneously* by synchronized, but uncalibrated cameras. The naive approach is to fit the 2D+3D AAM *independently* to each of the images I^n. Since the images are captured simultaneously, however, the 3D shape of the face should be the same whichever image it is computed in. The 2D+3D AAM can therefore be fit simultaneously [8] to the N images by minimizing:

$$\sum_{n=1}^{N}\left(\left\|A_0(\mathbf{u})+\sum_{i=1}^{l}\lambda^n_i A_i(\mathbf{u})-I^n(\mathbf{W}(\mathbf{u};\mathbf{p}^n))\right\|^2+\right.$$

$$K \cdot \left\| \mathbf{s}_0 + \sum_{i=1}^{m} p_i^n \, \mathbf{s}_i - \mathbf{P}_{\text{so}}^n \left(\overline{\mathbf{s}}_0 + \sum_{i=1}^{\overline{m}} \overline{p}_i \, \overline{\mathbf{s}}_i \right) \right\|^2 \Bigg) \quad (10)$$

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 $\mathbf{P}_{\text{so}}^n$, 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 manner through the single set of 3D shape parameters $\overline{\mathbf{p}}$. The optimization in Equation (10) uses the scaled orthographic camera matrices $\mathbf{P}_{\text{so}}^n$ in Equation (4) and optimizes over the N scale parameters σ^n. Using Equation (9) and optimizing over the focal lengths f^n and origin depths o_z^n is ambiguous. Multiple values of f^n and o_z^n yield the same value of $\sigma^n = \frac{f^n}{o_z^n + \overline{z}^n}$.

3.4. Calibration using Two Time Instants

For ease of understanding, we first describe an algorithm that uses two sets of multi-view images captured at two time instants. Deriving this algorithm also allows us to show that two sets of images are needed and derive the requirements on the motion of the face between the two time instants. In Section 3.5 we describe an algorithm that uses an arbitrary number of multi-view image sets and in Section 3.6 another algorithm that poses calibration as a single optimization.

The uncalibrated multi-view fitting algorithm defined by Equation (10) computes $\mathbf{i}^n$, $\mathbf{j}^n$, o_u^n, o_v^n, and σ^n. All that remains to calibrate the cameras is to compute f^n and o_z^n. These values can be computed by applying (a slightly modified version of) the uncalibrated multi-view fitting algorithm a second time with the face at a different location. With the first set of images we compute $\mathbf{i}^n$, $\mathbf{j}^n$, o_u^n, o_v^n. Suppose that $\sigma^n = \sigma_1^n$ is the scale at this location. Without loss of generality we also assume that the face model is at the world coordinate origin at this first time instant. Finally, without loss of generality we assume that the mean value of $\mathbf{x}$ computed across the face model (both mean shape $\mathbf{s}_0$ and all shape vectors $\mathbf{s}_i$) is zero. It follows that $\overline{z}$ is zero and so:

$$\frac{f^n}{o_z^n} \; = \; \sigma_1^n. \quad (11)$$

Suppose that at the second time instant the face has undergone a global rotation $\mathbf{R}$ and translation $\mathbf{T}$. Both the rotation $\mathbf{R}$ and translation $\mathbf{T}$ have three degrees of freedom. We then perform a modified multi-view fit, minimizing:

$$\sum_{n=1}^{N} \left(\left\| A_0(\mathbf{u}) + \sum_{i=1}^{l} \lambda_i^n A_i(\mathbf{u}) - I^n(\mathbf{W}(\mathbf{u}; \mathbf{p}^n)) \right\|^2 + K \cdot \right.$$

$$\left. \left\| \mathbf{s}_0 + \sum_{i=1}^{m} p_i^n \, \mathbf{s}_i - \mathbf{P}_{\text{so}}^n \left(\mathbf{R} \left(\overline{\mathbf{s}}_0 + \sum_{i=1}^{\overline{m}} \overline{p}_i \, \overline{\mathbf{s}}_i \right) + \mathbf{T} \right) \right\|^2 \right) \quad (12)$$

with respect to the N sets of 2D shape parameters $\mathbf{p}^n$, the N sets of appearance parameters λ_i^n, the one global set of 3D shape parameters $\overline{\mathbf{p}}$, the rotation $\mathbf{R}$, the translation $\mathbf{T}$, and the N scale values $\sigma^n = \sigma_2^n$. In this optimization all of the camera parameters ($\mathbf{i}^n$, $\mathbf{j}^n$, o_u^n, and o_v^n) except the scale (σ) in the scaled orthographic model $\mathbf{P}_{\text{so}}^n$ are held fixed to the values computed in the first time instant. Since the object underwent a global translation $\mathbf{T}$ then $\overline{z}^n = \mathbf{k}^n \cdot \mathbf{T}$ where $\mathbf{k}^n = \mathbf{i}^n \times \mathbf{j}^n$ is the z-axis of camera n. It follows that:

$$\frac{f^n}{o_z^n + \mathbf{k}^n \cdot \mathbf{T}} \; = \; \sigma_2^n. \quad (13)$$

Equations (11) and (13) are two linear simultaneous equations in the two unknowns (f^n and o_z^n). Assuming that $\mathbf{k}^n \cdot \mathbf{T} \neq 0$ (the global translation $\mathbf{T}$ is not perpendicular to any of the camera z-axes), these two equations can be solved for f^n and o_z^n to complete the camera calibration. Note also that to uniquely compute all three components of $\mathbf{T}$ using the optimization in Equation (12) at least one pair of the cameras must be verged (the axes ($\mathbf{i}^n$, $\mathbf{j}^n$) of the camera matrices $\mathbf{P}_{\text{so}}^n$ must not all span the same 2D subspace.)

3.5. Multiple Time Instant Algorithm

Rarely are two sets of multi-view images sufficient to obtain an accurate calibration. The approach just described can easily be generalized to N time instants. The first time instant is treated as above and used to compute $\mathbf{i}^n$, $\mathbf{j}^n$, o_u^n, o_v^n and to impose the constraint on f^n and o_z^n in Equation (11). Equation (12) is then applied to the remaining $N-1$ frames to obtain additional constraints:

$$\frac{f^n}{o_z^n + \mathbf{k}^n \cdot \mathbf{T}_j} \; = \; \sigma_j^n \quad (14)$$

for $j = 2, 3, \ldots, N$ where T_i is the translation estimated in the j^{th} time instant and σ_j^n is the scale. Equations (11) and (14) are then re-arranged to obtain an over-constrained linear system which can then be solved to obtain f^n and o_z^n.

3.6. Calibration as a Single Optimization

The algorithms in Section 3.4 and 3.5 have the disadvantage of being two stage algorithms. First they solve for $\mathbf{i}^n$, $\mathbf{j}^n$, o_u^n, and o_v^n, and then for f^n and o_z^n. It is better to pose calibration as the single large non-linear optimization of:

$$\left\| A_0(\mathbf{u}) + \sum_{i=1}^{l} \lambda_i^{n,j} A_i(\mathbf{u}) - I^{n,j}(\mathbf{W}(\mathbf{u}; \mathbf{p}^{n,j})) \right\|^2 + K \cdot$$

$$\left\| \mathbf{s}_0 + \sum_{i=1}^{m} p_i^{n,j} \, \mathbf{s}_i - \mathbf{P}_{\mathrm{wp}}^{n,j} \left(\mathbf{R}^j \left(\overline{\mathbf{s}}_0 + \sum_{i=1}^{\overline{m}} \overline{p}_i^j \, \overline{\mathbf{s}}_i \right) + \mathbf{T}^j \right) \right\|^2 \tag{15}$$

summed over all cameras n and time instants j with respect to the 2D shape parameters $\mathbf{p}^{n,j}$, the appearance parameters $\lambda_i^{n,j}$, the 3D shape parameters $\overline{\mathbf{p}}^j$, the rotations $\mathbf{R}^j$, the translations $\mathbf{T}^j$, and the calibration parameters $\mathbf{i}^n$, $\mathbf{j}^n$, f^n, o_u^n, o_v^n, and o_z^n. In Equation (15), $I^{n,j}$ represents the image captured by the n^{th} camera in the j^{th} time instant and the average depth $\overline{z} = \mathbf{k}^n \cdot \mathbf{T}^j$ in $\mathbf{P}_{\mathrm{wp}}^{n,j}$. Finally, we define the world coordinate system by enforcing $\mathbf{R}^1 = \mathbf{I}$ and $\mathbf{T}^1 = \mathbf{0}$.

The expression in Equation (15) can be optimized by iterating two steps: (1) The calibration parameters are optimized given the 2D shape and (rotated translated) 3D shape; i.e. the second term in Equation (15) is minimized given fixed 2D shape, 3D shape, $\mathbf{R}^j$, and $\mathbf{T}^j$. This optimization decomposes into a separate 7 dimensional optimization for each camera. (2) A calibrated multi-view fit (see Section 4) is performed on each frame in the sequence; i.e. the entire expression in Equation (15) is minimized, but keeping the calibration parameters in $\mathbf{P}_{\mathrm{wp}}^{n,j}$ fixed and just optimizing over the 2D shape, 3D shape, $\mathbf{R}^j$, and $\mathbf{T}^j$. The optimization can be initialized using the algorithm in Section 3.5.

3.7. Empirical Evaluation

We tested our algorithms on a trinocular stereo rig. Two example input images from each of the three cameras are shown in Figure 1. We wish to compare our algorithm with an algorithm that uses a calibration grid. Although our calibration algorithm computes 2×3 camera projection matrices, focal lengths, etc, the easiest way to compare two algorithms is using the epipolar geometry. Although we could use the calibration grid data to compute similar camera matrices, the world coordinate origin and units will be different. A direct comparison of the camera matrices therefore requires the units of one of them to be changed, possibly biasing the comparison. Instead, we compute a fundamental matrix from the camera parameters $\mathbf{i}^n$, $\mathbf{j}^n$, f^n, o_u^n, o_v^n, and o_z^n estimated by our algorithm and use the 8-point algorithm [7] to estimate the fundamental matrix from the calibration grid data.

In Figures 2 and 3 we present the results of a quantitative comparison. We compare the fundamental matrices by extracting a set of ground-truth feature point correspondences and computing the RMS distance between each feature point and the corresponding epipolar line predicted by the fundamental matrix. In Figure 2 we present results on 10 images of a calibration grid, similar (but not identical) to that used by the calibration grid algorithm. The ground-truth correspondences are extracted using a corner detector. In Figure 3 we present results on 1400 images of a face at different scales. The ground-truth correspondences are extracted by fitting a single-view AAM *independently* to each image (i.e. no use of the multi-view geometry is used.)

Figure 1. Example inputs to our calibration algorithms: A set of images of a face at a variety of different positions.

Although the optimization algorithm of Section 3.6 performs significantly better than the two stage algorithm in Section 3.5, both AAM-based algorithms perform slightly worse than the 8-point algorithm on the calibration grid data in Figure 2. The main reason is probably that the ground-truth calibration grid data covers a similar volume to the data used by the 8-point algorithm, but a much larger volume than the face data used by the AAM-based algorithms. When compared on the face data in Figure 3 (which covers a similar volume to that used by the AAM-based algorithm), the 8-point algorithm and the optimization algorithm of Section 3.6 perform comparably well.

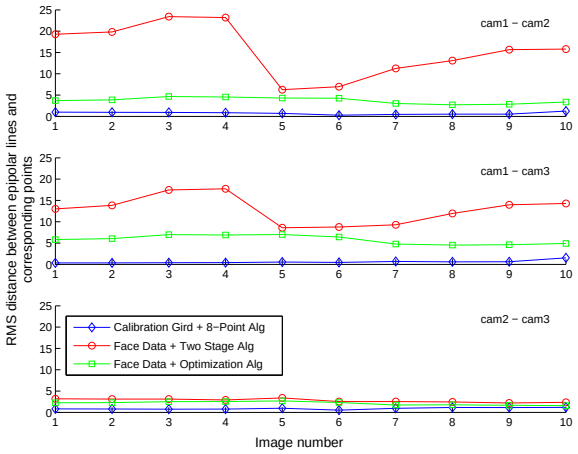


Figure 2. Quantitative comparison between our AAM-based calibration algorithms and the 8-point algorithm [7] using a calibration grid. The evaluation is performed on 10 images of a calibration grid (data similar to, but not used by the 8-point algorithm). The ground-truth is extracted using a corner detector. We plot the RMS distance error between epipolar lines and the corresponding feature points for each of the 10 images.

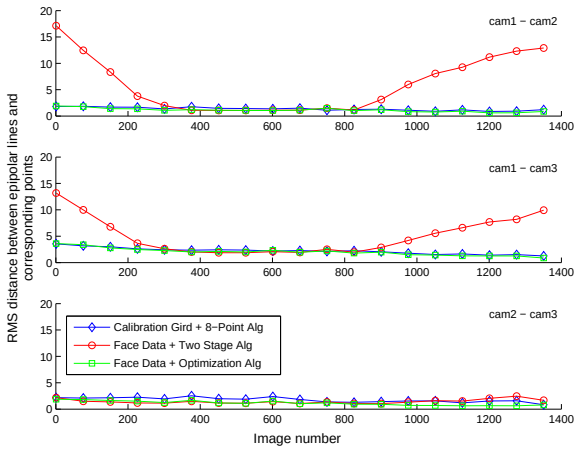
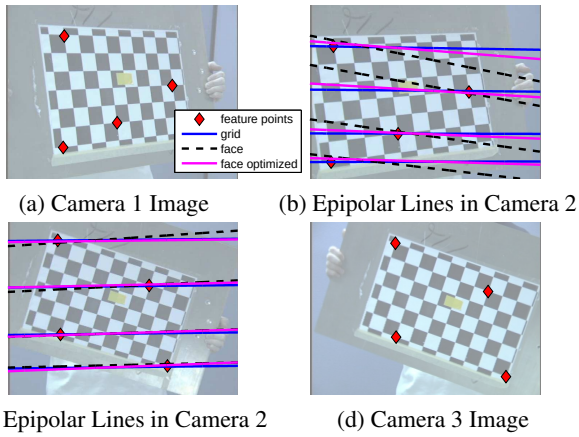


Figure 3. Quantitative comparison between our AAM-based calibration algorithms and the 8-point algorithm [7] using a calibration grid. The evaluation is performed on 1400 images of a face. The ground-truth is extracted using a *single-view* AAM fitting algorithm. We plot the RMS distance error between epipolar lines and the corresponding feature points for each of the 1400 images.



(a) Camera 1 Image (b) Epipolar Lines in Camera 2

(c) Epipolar Lines in Camera 2 (d) Camera 3 Image

Figure 4. Qualitative comparison between our AAM-based calibration algorithms and the 8-point algorithm [7]. (a) An input image captured by the first camera with several feature points marked on it. (b) The corresponding epipolar lines. The solid blue epipolar lines are computed using the 8-point algorithm, the dashed black epipolar lines using the two stage multiple time instant algorithm, and the solid magenta epipolar lines are computed using the optimization algorithm. (d) Shows the input image of the third camera, and (c) the corresponding epipolar lines for the second camera.

In Figure 4 we show a set of epipolar lines computed by the algorithms. In Figure 4(a) we show an input image captured by camera 1, with a few feature points marked on it. In Figure 4(b) we show the corresponding epipolar lines. The solid blue epipolar lines are computed using the 8-point algorithm. The dashed black epipolar lines are computed using the two stage multiple time instant algorithm of Section 3.5. The solid magenta epipolar lines are computed using the optimization algorithm of Section 3.6. Figures 4(d) and (c) are similar for feature points marked in camera 3. The epipolar lines for the optimization algorithm are far closer to those for the 8-point algorithm than those of the two stage algorithm.

4. Calibrated Multi-View Fitting

Once we have calibrated the cameras and computed $\mathbf{i}^n$, $\mathbf{j}^n$, f^n, o_u^n, o_v^n, and o_z^n, we can then use a weak perspective calibrated multi-view fitting algorithm. We can optimize:

$$\sum_{n=1}^{N} \left(\left\| A_0(\mathbf{u}) + \sum_{i=1}^{l} \lambda_i^n A_i(\mathbf{u}) - I^n(\mathbf{W}(\mathbf{u};\mathbf{p}^n)) \right\|^2 + K \cdot \right.$$

$$\left. \left\| \mathbf{s}_0 + \sum_{i=1}^{m} p_i^n \, \mathbf{s}_i - \mathbf{P}_{\mathrm{wp}}^n \left(\mathbf{R} \left(\overline{\mathbf{s}}_0 + \sum_{i=1}^{\overline{m}} \overline{p}_i \, \overline{\mathbf{s}}_i \right) + \mathbf{T} \right) \right\|^2 \right) \tag{16}$$

with respect to the N sets of 2D shape parameters $\mathbf{p}^n$, the N sets of appearance parameters λ_i^n, the one global set of 3D shape parameters $\overline{\mathbf{p}}$, the global rotation $\mathbf{R}$, and the global translation $\mathbf{T}$. In this optimization, $\mathbf{P}_{\mathrm{wp}}^n$ is defined by Equation (9) where $\overline{z} = \mathbf{k}^n \cdot \mathbf{T}$. It is also possible to formulate a similar scaled orthographic calibrated algorithm in which $\mathbf{P}_{\mathrm{wp}}^n$ is replaced with $\mathbf{P}_{\mathrm{so}}^n$ defined in Equation (4) and the optimization is also performed over the N scales σ_n.

4.1. Empirical Evaluation

An example of using our calibrated multi-view fitting algorithm to track by fitting a single 2D+3D AAM to three concurrently captured images of a face is shown in Figure 5. The top row of the figure shows the tracking result for one frame. The bottom row shows the the tracking result for a frame later in the sequence. In each case, all three input images are overlaid with the 2D shape $\mathbf{p}^n$ plot in blue dots. The single 3D shape $\overline{\mathbf{p}}$ at the current frame is displayed in the top-right of the center image. The view-dependent camera projection of this 3D shape is also plotted as a white mesh overlaid on the face. We also display the recovered roll, pitch, and yaw of the face (extracted from the global rotation matrix $\mathbf{R}$) in the top left of the center image. The three cameras combine to compute a single head pose, unlike [8] where the pose is view dependent.

In Figure 6 we show quantitative results to demonstrate the increased robustness and convergence rate of our calibrated multi-view fitting algorithm. In experiments sim-

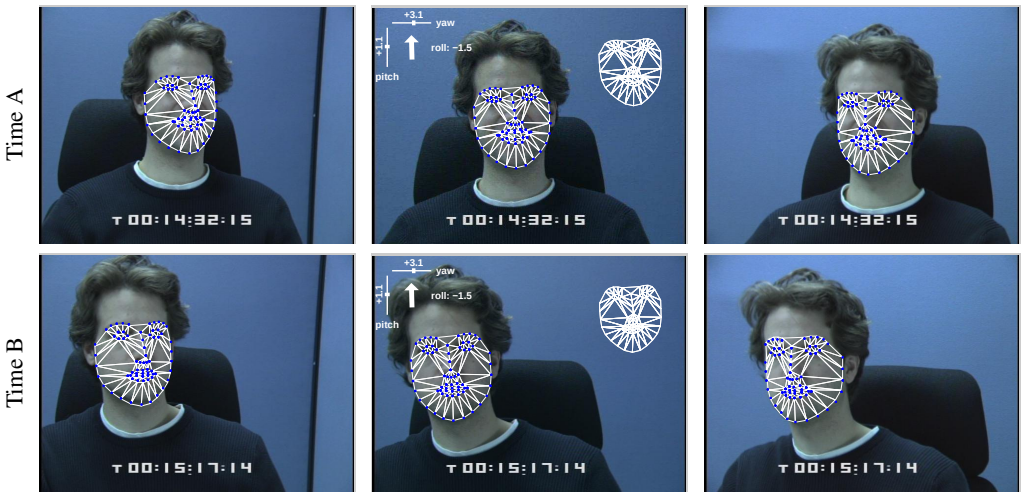


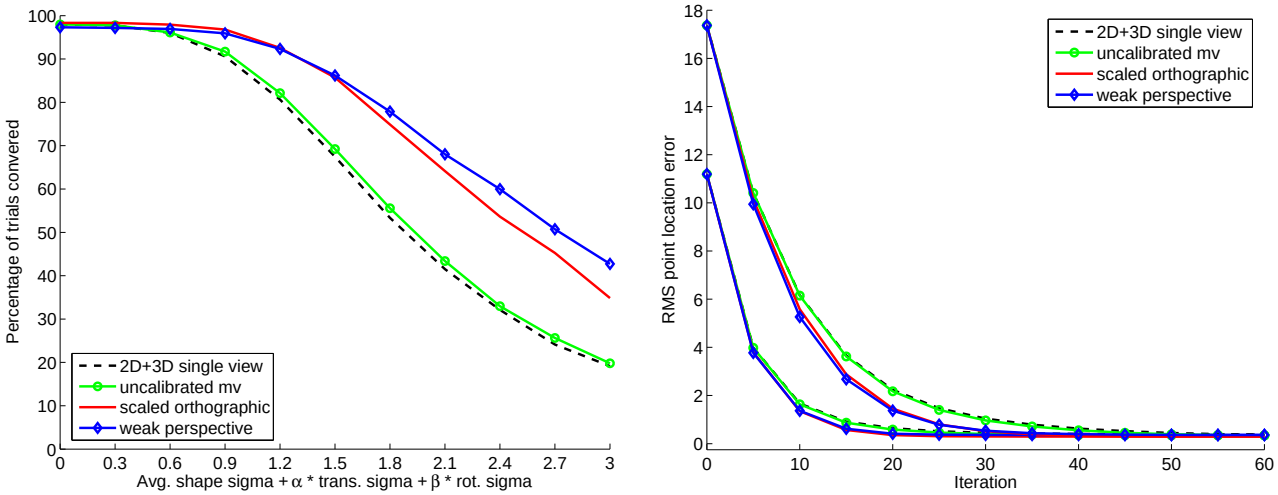
Figure 5. An example of using our calibrated multi-view fitting algorithm to fit a single 2D+3D AAM to three simultaneously captured images of a face. Each image is overlaid with the corresponding 2D shape for that image in blue dots. The single 3D shape $\overline{\mathbf{p}}$ for the current triple of images is displayed in the top right of the center image. This 3D shape is also projected into each image using the corresponding $\mathbf{P}^n$, and displayed as a white mesh. The single head pose (extracted from the rotation matrix $\mathbf{R}$) is displayed in the top left of the center image as roll, pitch, and yaw. This should be compared with the algorithm in [8] in which there is a separate head pose for each camera.

ilar to those in [10], we generated a large number of test cases by randomly perturbing from a ground-truth obtained by tracking the face in the multi-view video sequences. The global 3D shape parameters $\overline{\mathbf{p}}$, global rotation matrix $\mathbf{R}$, and global translation $\mathbf{T}$ were all perturbed and projected into each of the three views. This ensures the initial perturbation is a valid starting point for all algorithms. We then run each algorithm from the same perturbed starting point and determine whether they converged or not by computing the RMS error between the mesh location of the fit and the ground-truth mesh coordinates. The algorithm is considered to have converged if the total spatial error is less than 1 pixel. We repeat the experiment 10 times for each set of 3 images and average over all 300 image triples in the test sequences. This procedure is repeated for different values of perturbation energy. The magnitude of the perturbation is chosen to vary on average from 0 to 3 times the 3D shape standard deviation. The global rotation $\mathbf{R}$, and global translation $\mathbf{T}$ are perturbed by scalar multiples α and β of this value. The values of α and β were chosen so that the rotation and translation components introduce the same amount of pertubation energy as the shape component [10].

In Figure 6(a) we plot a graph of the likelihood (frequency) of convergence against the magnitude of the random perturbation for the the 2D+3D algorithm [12] applied independently to each camera, the uncalibrated multi-view 2D+3D algorithm of [8], and the two calibrated multi-view algorithms described in this paper: scaled orthographic, Equation (12), and weak-perspective, Equation (15). The results clearly show that the calibrated multi-view algorithms are more robust than the uncalibrated algorithm, which is more robust than the 2D+3D algorithm [12]. The main source of the increased robustness is imposing the constraint that the head pose is consistent across all N cameras. 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 6(b) and show that the calibrated multi-view algorithms converge faster than the uncalibrated algorithm, which converges faster than the 2D+3D algorithm. Overall the weak-perspective calibrated algorithm performs the best.

5. Conclusion

We have shown how multi-view face model fitting can be used to calibrate the relative orientations of a set of $N > 1$ cameras. In essence, we use the human face as a (non-rigid) calibration grid. Specifically, we used the weak perspective camera model used by most 3D face modeling papers [11, 12] and calibrated the 2×3 camera projection matrices, the focal lengths, the projection of the world coordinate system origin into the images, and the depths of the world coordinate system origins. We demonstrated that the resulting calibration is of comparable accuracy to that obtained using a calibration grid. We have also shown how the resulting calibration can be used to improve the performance of multi-view face model fitting.



(a) Frequency of Convergence

(b) Rate of Convergence

Figure 6. (a) The likelihood (frequency) of convergence plot against the magnitude of a random perturbation to the ground-truth fitting results computed by tracking through a trinocular sequence. The results show that the calibrated multi-view algorithms proposed in this paper are more robust than the uncalibrated multi-view algorithm proposed in [8], which itself is more robust than the 2D+3D algorithm [12]. (b) The rate of convergence is estimated by plotting the average error after each iteration against the iteration number. The results show that the calibrated multi-view algorithms converge faster than the uncalibrated algorithm, which converges faster than the 2D+3D algorithm.

In order to compute the focal lengths and depths of the world origin and so fully calibrate the cameras, we need at least two cameras and at least two time instants. It is not possible to compute the focal lengths with just a single camera (however many images are captured) or just a single time-instant (however many cameras are used.) However, with two or more cameras and two more more time instants the cameras can be fully calibrated. The only requirements are that: (1) for each camera, the translation of the face between at least one pair of time instants is not perpendicular to the z-axis, and (2) not all of the camera z-axes are parallel. The second condition is required to compute all three components of $\mathbf{T}$ using the optimization in Equation (15).

Acknowledgments

The research described in this paper was supported in part by DENSO Corporation, the U.S. Department of Defense contract N41756-03-C4024, and the National Institute of Mental Health grant R01 MH51435.

References

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

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, 1999.

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

[4] T. Cootes, G. Wheeler, K. Walker, and C. Taylor. Coupled-view active appearance models. In *Proc. of the British Machine Vision Conference*, 2000.

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

[6] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 20:1025–1039, 1998.

[7] R. Hartley. In defence of the 8-point algorithm. In *ICCV*, 1995.

[8] C. Hu, J. Xiao, I. Matthews, S. Baker, J. Cohn, and T. Kanade. Fitting a single active appearance model simultaneously to multiple images. In *Proceedings of the British Machine Vision Conference*, 2004.

[9] A. Lanitis, C. Taylor, and T. Cootes. Automatic interpretation and coding of face images using flexible models. *PAMI*, 19(7):742–756, 1997.

[10] I. Matthews and S. Baker. Active Appearance Models revisited. *IJCV*, 60(2):135–164, 2004.

[11] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3D morphable model. In *Proc. of the International Conference on Computer Vision*, 2003.

[12] 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*, 2004.

[13] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. In *Proc. of the European Conf. on Computer Vision*, 2004.