3D Morphable Model Fitting from Multiple Views

Nathan Faggian
Melbourne University
nfaggian@unimelb.edu.au

Andrew Paplinski
Monash University
app@csse.monash.edu.au

Jamie Sherrah
Safehouse International
jsherrah@gmail.com

Abstract

This paper presents a method to fit a 3D Morphable Model to a sequence of extracted facial features. The approach is a direct extension of a single view method. The novelty is presented as a new mathematical derivation of the same method but for multiple views where identity and pose are known. The new fitting method exploits point-to-model correspondences and can deal with occlusion since features are not required to correspond across views. The mathematics is explained in detail and the methods single tunable parameter is empirically determined using a database of scanned heads.

1. Introduction

Perhaps the most important recent discovery in computational face modelling is the 3D Morphable Model (3DMM) introduced by Vetter et al. [14]. A 3DMM is a representation of both the 3D shape and 2D texture of the human face. It is a direct extension of the 2D Active Appearance Model (AAM) that allows for more accurate modelling in the presence of pose and illumination variations. A Morphable Model is built from 3D laser scans of human faces that are put into dense correspondence [2] using their pixel intensities and 3D shape information. The correspondence of heads is achieved using a modified optical flow algorithm and provides a dense vertex to vertex mapping. Using the corresponding heads, shape and texture matrices are formed, where each column is a vectorized representation of the 3D data. In all cases, the dimensionality of each shape and texture matrix is very high. The dimensionality of the data must be reduced both for practicality and for model parametrization. This is achieved using the same principles as the AAM to provide the equations for shape and texture variation:

$$\hat{\mathbf{s}} = \overline{\mathbf{s}} + \mathrm{S} \cdot \mathrm{diag}(\sigma)\alpha, \qquad \hat{\mathbf{t}} = \overline{\mathbf{t}} + \mathrm{T} \cdot \mathrm{diag}(\sigma)\gamma \quad (1)$$

where $\hat{\mathbf{s}} \in \Re^{3\mathrm{n}\times 1}$ and $\hat{\mathbf{t}} \in \Re^{3\mathrm{n}\times 1}$ are novel shape and texture vectors, $\overline{\mathbf{s}} \in \Re^{3\mathrm{n}\times 1}$ and $\overline{\mathbf{t}} \in \Re^{3\mathrm{n}\times 1}$ are mean shape and texture vectors, $\mathrm{S} \in \Re^{3\mathrm{n}\times \mathrm{m}}$ and $\mathrm{T} \in \Re^{3\mathrm{n}\times \mathrm{m}}$ are column spaces (eigenvectors) of the shapes and textures, with σ as their corresponding eigenvalues, finally, $\alpha \in \Re^{\mathrm{m}\times 1}$ and $\gamma \in \Re^{\mathrm{n}\times 1}$ are shape and texture coefficients. These linear equations describe the variation of shape and texture within the span of the 3D training data. By varying the shape and texture coefficients, different shapes and textures are formed.

1.1. 3DMM Fitting Algorithms (texture based)

The first to introduce texture based fitting was Vetter et al [14]. Vetter's approach to fitting is holistic and makes use of image pixel intensities alone. In this approach model coefficients are inferred from the difference between the rendered model and the original image, in a similar fashion to AAM fitting. This is done using a modified gradient descent algorithm but is prohibitively slow. To improve convergence times the approach was extended to make use of image features such as edges and specular highlights by Romdhani et al [12]. There have also been many other algorithms for 3DMM fitting [5, 7, 11], where the best fitting times range from within 30 seconds [12] to 5 minutes per input image [5]. Although, the result of such (texture based) fittings is of course a highly accurate shape estimate and often produces a photo-realistic rendering of the human face. However, due to long fitting times, such approaches should be considered only for animation or off-line face recognition tasks.

1.2. 3DMM Fitting Algorithms (feature based)

Since the most significant problem with texture based fitting of the 3DMM is the amount of time required. In real-time applications, such as on-line face recognition, feature-fitting approaches that avoid the use of the 3DMM texture model and fit only on the basis of 2D projections of the 3D shape model have been developed. The benefit of not using the texture information in the 3DMM is two fold. Firstly, the dimensionality of the problem decreases, and secondly, the complexity of the problem also decreases. Although, fitting from feature points assumes that the 2D features in an image can be found without the use of the 3DMM. An-

978-1-4244-2154-1/08/$25.00

other critical assumption is that such 2D image features correspond directly to points in the 3D shape model of the 3DMM. However, given these assumptions, there has been a number of proposed feature-fitting algorithms [4, 10, 15].

The fundamental theory behind feature based fitting is that 3DMM shape coefficients can be extracted from an incomplete set of shape feature points in either 3D or 2D. In this vein, one of the first feature-based fitting algorithms was demonstrated by Hwang et al. [8] by reducing the shape basis vectors to the same dimensionality as the small set of texture or shape features. This can be achieved in a straightforward way after re-writing the problem as:

$$\begin{aligned} \mathrm{VS}\alpha &= \hat{\mathbf{s}} \\ \alpha &= (\mathrm{VS}^{\mathrm{T}}\mathrm{VS})^{-1}\mathrm{VS}^{\mathrm{T}}\hat{\mathbf{s}} \end{aligned} \tag{2}$$

where S are the reduced shape basis vectors, reduced with the mapping matrix V, α are shape coefficients and $\hat{\mathbf{s}}$ is the occluded shape. This shows a solution for shape coefficient estimation from partial information as a pseudo-inverse and in [8] it is shown to be plausible even when face models were significantly occluded. However, an important oversight of the approach is that the shape coefficients are not constrained to be valid shape model parameters. To achieve this, the parameters must be restricted in some fashion, which a pseudo-inversion does not achieve.

As in [8] where 2D shapes were addressed, by using 2D image features that correspond to 3DMM vertices it is possible to determine 3DMM shape coefficients directly. This is an approach presented by Blanz et al. [3] as a technique to linearize what would otherwise be a non-linear 3DMM fitting problem. Using corresponding points the solution for 3DMM shape coefficients can be found using a regularized inversion. As the first step [3] defines the measurements in an image plane as the 2D projection of a subset of the 3DMM shape model:

$$\mathbf{r} = \mathrm{L}\mathrm{V}\bar{\mathbf{s}} + \mathrm{LVS}\cdot\mathrm{diag}(\sigma)\alpha \tag{3}$$

where $\mathrm{L} \in \Re^{2\mathrm{p}\times 3\mathrm{p}}$ is a camera matrix, containing the full set of intrinsic and extrinsic parameters, $\mathrm{V} \in \Re^{3\mathrm{p}\times 3\mathrm{n}}$ is a subset selection matrix, $\mathrm{S} \in \Re^{3\mathrm{n}\times \mathrm{m}}$ is the column (eigenvectors) space of the 3DMM training shapes, σ are the corresponding eigenvalues, $\alpha \in \Re^{\mathrm{m}\times 1}$ are shape coefficients and $\mathbf{r} \in \Re^{2\mathrm{p}\times 1}$ set of feature points. It is clear in this representation that model coefficients can easily be derived from Equation 3 using only the pseudo-inverse $(+)$ operation. However, it is shown in [3] that simply applying the pseudo-inverse will not produce a perceptually pleasing result. This is because the method only minimizes the 2D re-projection error between the 3D model and the 2D features, $\mathbf{r}$. Specifically, this perceptually poor result is the result of shape coefficients α that are not suitably restricted to the span of the 3DMM shape model. Actually, it was shown in [3] that the correct approach is a statistical one which guarantees the maximum posterior probability coefficient estimate. This derivation presented a modified cost function, namely:

$$\epsilon = \parallel \mathrm{LVSdiag}(\sigma)\alpha - (\mathbf{r} - \mathrm{LV}\bar{\mathrm{s}}) \parallel^{2} + \eta \parallel \alpha \parallel^{2} \tag{4}$$

In [3], for clarity, matrices are combined to form a simpler cost function:

$$\epsilon = \parallel \mathrm{Q}\alpha - \mathbf{y} \parallel^{2} + \eta \parallel \alpha \parallel^{2} \tag{5}$$

where $\mathrm{Q} = \mathrm{LVS}\cdot\mathrm{diag}(\sigma)$ and $\mathbf{y} = \mathbf{r} - \mathrm{LV}\bar{\mathbf{s}}$. In this form the cost function can be solved in a straight forward fashion:

$$\begin{aligned} \bigtriangledown\epsilon &= 2\mathrm{Q}^{\mathrm{T}}\mathrm{Q}\alpha - 2\mathrm{Q}^{\mathrm{T}}\mathbf{y} + 2\alpha = 0 \\ \mathrm{Q}^{\mathrm{T}}\mathbf{y} &= \mathrm{Q}^{\mathrm{T}}\mathrm{Q}\alpha + \eta\alpha \end{aligned}$$

Now SVD can be applied to compute the inverse, specifically in the case $\mathrm{Q}^{\mathrm{T}}\mathrm{Q}$, where $\mathrm{Q} = \mathrm{U}\cdot\mathrm{WV}^{\mathrm{T}}$ and $\mathrm{Q}^{\mathrm{T}} = \mathrm{V}\cdot\mathrm{SU}^{\mathrm{T}}$.

$$\begin{aligned} \mathrm{Q}^{\mathrm{T}}\mathbf{y} &= \mathrm{Q}^{\mathrm{T}}\mathrm{Q}\alpha + \eta\alpha \\ \mathrm{V}\cdot\mathrm{WU}^{\mathrm{T}}\mathbf{y} &= \mathrm{V}\cdot\mathrm{WU}^{\mathrm{T}}\mathrm{U}\cdot\mathrm{WV}^{\mathrm{T}}\alpha + \eta\alpha \\ \mathrm{V}\cdot\mathrm{WU}^{\mathrm{T}}\mathbf{y} &= \mathrm{V}\cdot\mathrm{W}^{2}\mathrm{V}^{\mathrm{T}}\alpha + \eta\alpha \end{aligned} \tag{6}$$

This leads to the solution (after multiplying both sides of Equation 6 by V^{T}) for α in Equation 5:

$$\alpha = \mathrm{V}\mathrm{diag}\left(\frac{\mathrm{w_i}}{\mathrm{w_i}^2 + \eta}\right)\mathrm{U}^{\mathrm{T}}\mathbf{y} \tag{7}$$

This solution adds the regularization constant to the diagonal of the singular values of the SVD factorization of Q. It solves the problem of estimating shape coefficients from a sparse set of 2D features while also assuring a plausible result. However, the solution is dependent on the selection of the regularization term η and a correct estimation of the camera matrix, L.

(a) $\hat{\mathbf{s}}$ (b) $\eta = 10^{-3}$ (c) $\eta = 10^{1}$ (d) $\eta = 10^{3}$ (e) $\bar{\mathbf{s}}$

Figure 1. The variation of η, on a random 3D head, $\hat{\mathrm{s}}$, demonstrating how the estimate approaches the mean, $\bar{\mathbf{s}}$.

As η increases the 3D shape estimate tends to the mean 3DMM shape, or rather, the norm of the estimated coefficients approach zero. This is shown in Figure 1 where η is increased to demonstrate its regularizing effect. However, introducing a regularization introduces a trade-off between a perceptually smooth 3D shape estimates and a minimum re-projection error between the 3DMM and the extracted 2D image features.

2. Fitting Morphable Models to *2*-Views

A significant problem with the approach presented in [3] is that it is only derived for a single set of 2D features and consequently can only be applied to a single image. Fortunately, it is straightforward to extended the method to estimate shape coefficients given multiple images which allows 3DMM fitting to extend to multiple views of the same face over time and across pose changes. In the first case the problem of estimating 3DMM shape coefficients using two views is shown. The key difference is to solve a modification of Equation 5, for example:

$$\begin{aligned}\epsilon &= \parallel \mathrm{L}_1\mathrm{VSdiag}(\sigma)\alpha - (\mathbf{r_2} - \mathrm{L}_1\mathrm{V}\bar{\mathbf{s}}) \parallel^2 + \\ &\quad \parallel \mathrm{L}_2\mathrm{VSdiag}(\sigma)\alpha - (\mathbf{r_2} - \mathrm{L}_2\mathrm{V}\bar{\mathbf{s}}) \parallel^2 + \dots \\ &\quad \eta \parallel \alpha \parallel^2 \end{aligned} \tag{8}$$

In the same fashion as [3], the cost function matrices are combined to show a simple solution for its minimum exists:

$$\epsilon = \frac{1}{2} \parallel \mathrm{Q}_0\alpha - \mathbf{y_0} \parallel^2 + \frac{1}{2} \parallel \mathrm{Q}_1\alpha - \mathbf{y_1} \parallel^2 + \eta \parallel \alpha \parallel^2 \tag{9}$$

where:

$$\begin{aligned}\mathrm{Q}_0 &= \mathrm{L}_1\mathrm{VS} \cdot \mathrm{diag}(\sigma) \\ \mathrm{Q}_1 &= \mathrm{L}_2\mathrm{VS} \cdot \mathrm{diag}(\sigma) \\ \mathbf{y_0} &= \mathbf{r_1} - \mathrm{L}_1\mathrm{V}\bar{\mathbf{s}} \\ \mathbf{y_1} &= \mathbf{r_2} - \mathrm{L}_2\mathrm{V}\bar{\mathbf{s}}\end{aligned}$$

Representing the cost function for a two view case, where $\mathbf{r_1}, \mathbf{r_2}$ are the set of 2D image features and $\mathrm{L}_1, \mathrm{L}_2$ approximately align the 3D model to these features. As in [3], it is possible to solve for the coefficients α in a straight forward fashion. Unfortunately, the elegance of the SVD identities cannot be exploited, because of multiple instances of the Q matrix. Instead a solution is derived:

$$\begin{aligned}\bigtriangledown\epsilon &= \mathrm{Q}_0^{\mathrm{T}}\mathrm{Q}_0\alpha - \dots \\ &\quad \mathrm{Q}_0^{\mathrm{T}}\mathbf{y_0} + \mathrm{Q}_1^{\mathrm{T}}\mathrm{Q}_1\alpha - \mathrm{Q}_1^{\mathrm{T}}\mathbf{y_1} + 2\alpha = 0 \\ \mathrm{Q}_0^{\mathrm{T}}\mathbf{y_0} + \mathrm{Q}_1^{\mathrm{T}}\mathbf{y_1} &= \mathrm{Q}_0^{\mathrm{T}}\mathrm{Q}_0\alpha + \mathrm{Q}_1^{\mathrm{T}}\mathrm{Q}_1\alpha + 2\eta\alpha\end{aligned} \tag{10}$$

This equation simplifies by factorizing and re-arranging with respect to η:

$$\mathrm{Q}_0^{\mathrm{T}}\mathbf{y_0} + \mathrm{Q}_1^{\mathrm{T}}\mathbf{y_1} = \left(\mathrm{Q}_0^{\mathrm{T}}\mathrm{Q}_0 + \mathrm{Q}_1^{\mathrm{T}}\mathrm{Q}_1 + 2\mathrm{I}\eta\right)\alpha$$

In a similar fashion to [3] the solution forms a regularized pseudo-inverse:

$$\alpha = \left(\mathrm{Q}_0^{\mathrm{T}}\mathrm{Q}_0 + \mathrm{Q}_1^{\mathrm{T}}\mathrm{Q}_1 + 2\mathrm{I}\eta\right)^{\dagger}\left(\mathrm{Q}_0^{\mathrm{T}}\mathbf{y_0} + \mathrm{Q}_1^{\mathrm{T}}\mathbf{y_1}\right) \tag{11}$$

Importantly, the principle assumption when solving for shape coefficients across multiple views is that both the identity and expression is fixed (images of the same person in rigid motion). In this case the problem is somewhat simplified and the possible source of error is reduced to model alignment due to a poor estimate of the camera matrix L. In this solution the regularization cost ($\eta \parallel \alpha \parallel^2$) is also ensured. This forces a plausible set of coefficients. It must also be assumed that the cost associated with each view is equal, hence the scaling of $\frac{1}{2}$ for the 2-view case.

3. Fitting the Morphable Model to *n*-Views

The *2*-view case can be made into a generic solution and extended to *n*-views. This is achieved in a similar fashion to §2 and requires the cost function shown in Equation 9 to be modified. In its new form the cost function is a now the summation:

$$\epsilon = \frac{1}{n}\sum_{\mathrm{i}=1}^{n} \parallel \mathrm{Q_i}\alpha - \mathbf{y_i} \parallel^2 + \eta \parallel \alpha \parallel^2 \tag{12}$$

where *n* is the number of views, $\mathrm{Q_i} = \mathrm{L_i}\mathrm{VSdiag}(\sigma)$ and $\mathbf{y_i} = \mathbf{r_i} - \mathrm{L_i}\mathrm{V}\bar{\mathbf{s}}$. As in§2, the regularization cost is added. However, an important difference is the introduction of a view scaling constant, $\frac{1}{n}$. The role of the view scaling constant is to allow for the individual views to have the same importance in the summed cost function. It also has the desirable effect of reducing the cost difference between the summed views and the additional regularization cost. After expanding the cost function it can be solved in a similar manner to §2, where:

$$\begin{aligned}\epsilon &= \frac{1}{n}\sum_{\mathrm{i}=1}^{n} \parallel \mathrm{Q_i}\alpha - \mathbf{y_i} \parallel^2 + \eta \parallel \alpha \parallel^2 \\ &= \frac{1}{n}\sum_{\mathrm{i}=1}^{n} \left(\mathrm{Q_i}\alpha - \mathrm{y_i}\right)^{\mathrm{T}}\left(\mathrm{Q_i}\alpha - \mathbf{y_i}\right) + \eta \parallel \alpha \parallel^2 \\ &= \frac{1}{n}\sum_{\mathrm{i}=1}^{n} \left(\alpha^{\mathrm{T}}\mathrm{Q_i}^{\mathrm{T}}\mathrm{Q_i}\alpha - 2\mathrm{Q_i}^{\mathrm{T}}\mathrm{y_i}\alpha + \mathbf{y_i}^{\mathrm{T}}\mathbf{y_i}\right) + \eta\alpha^2 \\ \bigtriangledown\epsilon &= \frac{1}{n}\sum_{\mathrm{i}=1}^{n} \left(2\mathrm{Q_i}^{\mathrm{T}}\mathrm{Q_i}\alpha - 2\mathrm{Q_i}^{\mathrm{T}}\mathbf{y_i}\right) + 2\eta\alpha\end{aligned} \tag{13}$$

Now the optimum, $\bigtriangledown\epsilon = 0$, can be solved in a similar fashion to §2:

$$\begin{aligned} 0 &= \frac{1}{n}\sum_{\mathrm{i}=1}^{n} \left(2\mathrm{Q_i}^{\mathrm{T}}\mathrm{Q_i}\alpha - 2\mathrm{Q_i}^{\mathrm{T}}\mathbf{y_i}\right) + 2\eta\alpha \\ \frac{1}{n}\sum_{\mathrm{i}=1}^{n} \left(\mathrm{Q_i}^{\mathrm{T}}\mathbf{y_i}\right) &= \frac{1}{n}\sum_{\mathrm{i}=1}^{n} \left(\mathrm{Q_i}^{\mathrm{T}}\mathrm{Q_i}\alpha\right) + \eta\alpha \\ \frac{1}{n}\sum_{\mathrm{i}=1}^{n} \left(\mathrm{Q_i}^{\mathrm{T}}\mathbf{y_i}\right) &= \frac{1}{n}\sum_{\mathrm{i}=1}^{n} \left(\mathrm{Q_i}^{\mathrm{T}}\mathrm{Q_i}\alpha + n\mathrm{I}\eta\right)\alpha \end{aligned}$$

The solution is, as before, the pseudo inversion:

$$\alpha = \left(\sum_{\mathrm{i}=1}^{n} \left(\mathrm{Q_i^T Q_i} + n\mathrm{I}\eta \right) \right)^{+} \sum_{\mathrm{i}=1}^{n} \left(\mathrm{Q_i^T} \mathbf{y_i} \right) \qquad (14)$$

Equation 14 therefore demonstrates a method for estimating 3DMM shape coefficients from multiple sets of measurements. Significantly, the solution also ensures that the shape coefficients are within the span of the 3DMM shape model by including the regularization cost ($\eta \parallel \alpha \parallel^2$) in the summed cost function of Equation 12.

3.1. Missing 2D Features

In [3] shape coefficients are inferred from a single view using correspondence between the 3D Model and the 2D image features. In the extensions presented in §2 and §3 many sets of 2D features and different correspondences with the 3D model are used to estimate shape coefficients. In many ways the extension to [3] is similar to the popular 'structure from motion' (SfM) approach introduced by Tomasi and Kanade [13]. In SfM the structure and camera properties of a 3D object are inferred from a set of corresponding 2D features. However, an important limitation of this method is that the features must be tracked across a sequence, without failure. When features are missed because of failed tracking their values are estimated through a process called imputation. The common problem with SfM is the reliance on feature correspondence, either inferred or actual. This is an important problem that this paper implicitly solves.

This is shown in the solution presented in §2 which is not coupled to features corresponding between views or across an image sequence. Unlike SfM there is no need to impute or predict the location of 2D features that have not been tracked in different views. This is due to the product of $\mathrm{Q^T Q}$ and $\mathrm{Q^T} \mathbf{y}$ in Equation 14. Since the outer dimensions are fixed, the number of features per view is independent of the estimated 3DMM shape coefficients. Furthermore the solution presented in §2 also has the benefit of being domain specific. Since it uses a face model to estimate the 3D structure of faces its solutions will only ever be plausible 3D faces. SfM is not constrained in this way and can, in practice, yield inappropriate 3D estimates.

3.2. The Shape Update Algorithm

Using the derivations from §3 a novel algorithm to incrementally fit a 3DMM from set of images features is presented. Called the Shape-Update algorithm, the new approach is a modification up of the solution presented to Equation 14 to allow for incremental computation of 3DMM shape coefficients. In doing so, the algorithm relies on prior information about each view being known. Specifically, the information required is the current measurement vector $\mathrm{r_i}$, the camera matrix $\mathrm{L_i}$, the mapping matrix $\mathrm{V_i}$, the number of views n and the regularization constant η. With this information it is possible to split the summed cost function of §3 and the process is shown in Algorithm 1.

Algorithm 1 shapeUpdate ($\mathbf{r_i}$, $\mathrm{L_i}$, $\mathrm{V_i}$, n, η)

Require: $n \geq 1$.

Ensure: A and B are **static variables**.

$\mathrm{Q_i} = \mathrm{L_i V_i S} \cdot \mathrm{diag}(\sigma)$ {Compute the subselection of projected eigenvectors}

$\mathbf{y_i} = \mathbf{r_i} - \mathrm{L_i V_i} \bar{\mathbf{s}}$ {Compute the demeaned measurement vector}

if $n = 1$ **then**

  $\mathrm{USV^T} = \mathrm{Q_i}$ {Apply SVD to $\mathrm{Q_i}$}

  $\mathrm{A} = \mathrm{Q_i^T Q_i}$

  $\mathrm{B} = \mathrm{Q_i^T} \mathbf{y_i}$

  return $\alpha = \mathrm{V diag}\left(\frac{\mathrm{w_i}}{\mathrm{w_i^2} + \eta} \right) \mathrm{U^T} \mathbf{y_i}$ {Equation 7}

else

  $\mathrm{A} = \mathrm{A} + \mathrm{Q_i^T Q_i}$

  $\mathrm{B} = \mathrm{B} + \mathrm{Q_i^T} \mathbf{y_i}$

  return $\alpha = (\mathrm{A} + \mathrm{n}\mathrm{I}\eta)^{+} \mathrm{B}$ {Equation 14}

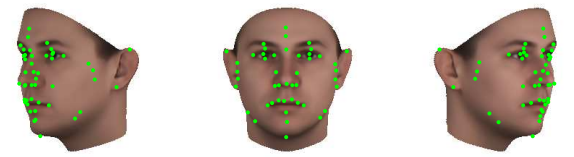
end if

Importantly, in the first view $\mathrm{n} = 1$ the optimal solution for shape coefficients is to use [3] regularized fitting. In this step it is also important to cache the products A and B. However, when n increases the Shape-Update algorithm must switch to minimize the modified cost function shown in Equation 12 (also caching the products A, B). The caching of the products A and B allow the approach to be applied to any number of views without the dimensionality of the problem increasing. This fixed dimensionality is a feature of the approach that was briefly explained in §3.1.

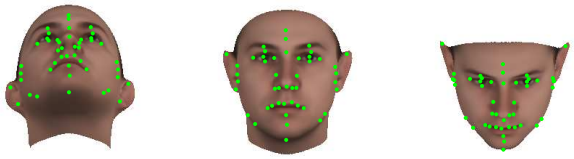
A computational aspect of the algorithm is the initial SVD (for inversion) operation. This is an order N^3 operation that is used to compute both the single view fitting [3] and the multiple view pseudo inversion. However, the SVD is computed on a constant size matrix. This implies that the effective dimensionality of the of the solution does not increase as it processes more views from an image sequence.

4. Experiments

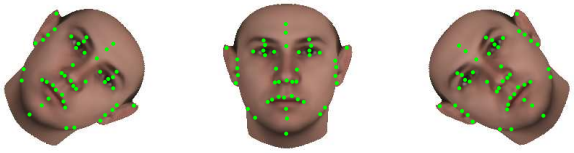
Using a 3D head database [1] the set of 75 aligned heads was randomly divided into 4 subsets. In accordance with the cross-validation algorithm permutations, these subsets were used to provide training and testing data to construct 3DMMs for subsequent Shape-Update experiments. Since the Shape-Update approach to 3DMM fitting from



(a) Yaw Rotation Axis



(b) Pitch Rotation Axis



(c) Roll Rotation Axis

Figure 2. Examples of the yaw, pitch and roll axis rotation and the select 2D feature points.

addresses the problem of fitting given a sequence of measurements, the testing data was used to provide accurate sets of measurement vectors r and camera matrices L. For each head, three tracking sequences were generated that correspond to yaw, pitch and roll motions in the range of -50 to 50 degrees. The same subset of anthropometrically meaningful feature points were then tracked through these motions, demonstrated in Figure 2.

During 3DMM fitting, it is important to obtain an estimate of all camera projections L which align the reduced 3DMM shape model to each measurement vector r in the tracking sequences. Such an alignment process is otherwise known as exterior orientation and must be applied to each tracking in the four testing sets. For the sake of evaluating the Shape-Update algorithm, a modification of the method from [9] is applied.

4.1. Shape Estimation

Using the experimental setup described in §4 the Shape-Update algorithm is evaluated with respect to 2D (re-projection and 3D (shape) errors. To compute the 2D errors the (aligned) estimated 3D heads are projected to 2D for each face in the 3 tracking sequences. Notably, in this case the 2D re-projection error is directly minimized by the algorithm. As a by product, the method also minimizes the 3D Euclidean error between the test and estimated 3D head. This is computed as the Euclidian distance from the estimate head, fitted using the features from a tracking, to the test head. Importantly, the outcome of the 2D and 3D experiments is the selection of a good value for the regularization constant η. The only way to accurately select a value for η is to measure both the 2D and 3D error and pick the trade-off point where the error in both measures is minimized.

To demonstrate the dependence on η a series of cross validation experiments are performed using a suitable range for η extracted from [3]. The result is presented in Table 1:

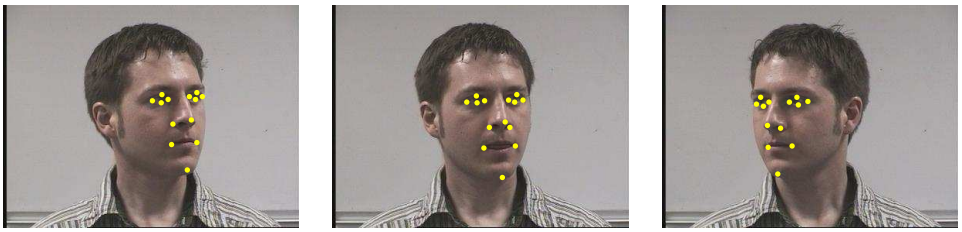
	YAW	PITCH	ROLL
η	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$
0.0001	1.108 ± 0.218	1.045 ± 0.181	0.664 ± 0.129
0.01	1.112 ± 0.220	1.050 ± 0.179	0.662 ± 0.129
1	1.202 ± 0.203	1.142 ± 0.191	0.856 ± 0.148
10	1.596 ± 0.241	1.548 ± 0.253	1.342 ± 0.226
100	2.275 ± 0.388	2.170 ± 0.368	2.043 ± 0.384
1000	2.717 ± 0.544	2.578 ± 0.530	2.510 ± 0.601

Table 1. The 2D re-projection error (pixels) for various η values, for a yaw, pitch and roll motions. Where μ is the mean 2D fitting error and σ represents one standard deviation.

	YAW	PITCH	ROLL
η	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$
0.0001	5.659 ± 1.476	5.313 ± 1.576	6.314 ± 1.745
0.01	5.740 ± 1.501	5.385 ± 1.602	6.486 ± 1.826
1	4.598 ± 1.203	4.461 ± 1.255	4.797 ± 1.263
10	4.281 ± 1.106	4.220 ± 1.127	4.392 ± 1.138
100	4.484 ± 1.120	4.469 ± 1.144	4.509 ± 1.144
1000	4.741 ± 1.161	4.738 ± 1.165	4.741 ± 1.166

Table 2. The 3D Euclidean distance of the estimated head from ground truth head (millimeters). For various η values and yaw, pitch and roll motions. Where μ is the mean 3D-3D Euclidian distance and σ represents one standard deviation.

Importantly, the regularizing constant η also strongly influences the 3D shape error. The result is shown in Table 2. Importantly, the 3D error table shows there is an optimal trade-off between the 2D re-projection error and the 3D distance from test set with an η value of 10. This is the cross-over point for for the average 3D error and the 2D re-projection error using the USF database [1]. It is also shown that for high values of eta η the 3D error is fixed (equivalent to the aligned 3DMM mean shape) but as η decreases the shape coefficients change and the 3D shape also error decreases. The reader should note this is the inverse effect in the case of 2D re-projection errors. An example of multiple-view 3DMM fitting is shown in Figure 3. Given the three sets of features, the 3DMM shape can be computed, either in a batch fashion or incrementally using the Shape-Update algorithm.



(a) Salient feature points on non-frontal faces.

(b) From left to right the incremental shape adaptation using the Shape-Update algorithm can be seen. There is an obvious change in the estimated chin after three updates.

Figure 3. Shape estimation of a subject from the Pointing database [6].

5. Conclusion

This paper has presented a novel multiple-view extension to the 3DMM fitting algorithm presented by Blanz et al. [3].The mathematics for the method has been outlined in the paper and it has been evaluated using ground truth data using a 3D head database. The evaluation shows that the regularization term (the only parameter for the fitting) can be found empirically. The end result is a fast multiple view 3DMM fitting algorithm that can be used to supplement robust facial feature detectors (such as AAM's).

References

[1] USF DARPA HumanID 3D Face Database, Courtesy of Professor Sudeep Sarkar. University of South Florida, Tampa, FL.

[2] C. Basso, T. Vetter, and V. Blanz. Regularized 3D Morphable Models. In *Proceedings of Higher-Level Knowledge in 3D Modeling and Motion Analysis*, 2003.

[3] V. Blanz, A. Mehl, T. Vetter, and H. Seidel. A statistical method for robust 3D surface reconstruction from sparse data. In *Second International Symposium on 3D Data Processing, Visualization and Transmission*, 2004.

[4] V. Blanz and T. Vetter. Reconstructing the complete 3D shape of faces from partial information. Technical report, Universitat Frieburg, 2002.

[5] V. Blanz and T. Vetter. Face Recognition Based on Fitting a 3d Morphable Model. *Pattern Analysis and Machine Intelligence*, 2003.

[6] N. Gourier, D. Hall, and J. L. Crowley. Estimating Face Orientation from Robust Detection of Salient Facial Features. In *Proceedings of Pointing, ICPR, International Workshop on Visual Observation of Deictic Gestures*, 2004.

[7] Y. Hu, B. Yin, S. Chin, and C. Gu. An Improved Morphable Model for 3D Face Synthesis. In *Proceedings of Machine Learning and Cybernetics*, 2004.

[8] B. Hwang and S. Lee. Face Reconstruction with a Morphable Face Model. In *Proceedings of International Conference on Pattern Recognition*, 2006.

[9] T. S. Jebara and A. Pentland. Parametrized Structure from Motion for 3D Adaptive Feedback Tracking of Faces. Technical report, MIT Media Labratory, Perceptual Computing, 1996.

[10] R. Knothe, S. Romdhani, and T. Vetter. Combining PCA and LFA for Surface Reconstruction from a Sparse Set of Control Points. In *Proceedings of Face and Gesture Recognition*, 2006.

[11] J. Lee, R. Machiraju, H. Pfister, and B. Moghaddam. Estimation of 3D Faces and Illumination from Single Photographs Using a Bilinear Illumination Model. In *Proceedings of the Eurographics Symposium on Rendering*, 2005.

[12] S. Romdhani and T. Vetter. Estimating 3D Shape and Texture Using Pixel Intensity, Edges, Specular Highlights, Texture Constraints and a Prior. In *Proceedings of Computer Vision and Pattern Recognition*, 2005.

[13] C. Tomasi and T. Kanade. Shape and motion from image streams: A factorization method. In *Proceedings, National Academy of Sciences, USA*, volume 90, 1992.

[14] T. Vetter and V. Blanz. A Morphable Model For The Synthesis Of 3D Faces. In *Proceedings of SIGGRAPH, Computer Graphics*, 1999.

[15] M. Zhao, T. S. Chua, and T. Sim. Morphable Face Reconstruction with Multiple images. In *Proceedings of Face and Gesture Recognition*, 2006.