Using 3D Morphable Models for face recognition in video

R.T.A. van Rootseler, L.J. Spreeuwers, R.N.J. Veldhuis
University of Twente
Signals & Systems Group
Drienerlolaan 5, P.O.Box 217, 7500 AE Enschede, The Netherlands
r.t.a.vanrootseler@ewi.utwente.nl

Abstract

The 3D Morphable Face Model (3DMM)[1] has been used for over a decade for creating 3D models from single images of faces. This model is based on a PCA model of the 3D shape and texture generated from a limited number of 3D scans. The goal of fitting a 3DMM to an image is to find the model coefficients, the lighting and other imaging variables from which we can remodel that image as accurately as possible. The model coefficients consist of texture and of shape descriptors, and can without further processing be used in verification and recognition experiments. Until now little research has been performed into the influence of the diverse parameters of the 3DMM on the recognition performance.

In this paper we will introduce a Bayesian-based method for texture backmapping from multiple images. Using the information from multiple (non-frontal) views we construct a frontal view which can be used as input to 2D face recognition software. We also show how the number of triangles used in the fitting process influences the recognition performance using the shape descriptors.

The verification results of the 3DMM are compared to state-of-the-art 2D face recognition software on the MultiPIE dataset. The 2D FR software outperforms the Morphable Model, but the Morphable Model can be useful as a preprocessor to synthesize a frontal view from a non-frontal view and also combine images with multiple views to a single frontal view. We show results for this preprocessing technique by using an average face shape, a fitted face shape, with a MM texture, with the original texture and with a hybrid texture. The preprocessor has improved the verification results significantly on the dataset.

1 Introduction

The state-of-the-art in forensic face recognition is based on comparing a single image from a video with a suspect. Information from all the other video frames is lost. Moreover it is unlikely that suspects look straight into the camera, so a clear, well illuminated, frontal view of the suspect is not present. Therefore, in the Person Verification 3D project we aim to improve this process by taking 3D information of a face into account in image sequences from uncalibrated cameras.

A 3D Morphable Face Model (3DMM) provides a method to generate a 3D reconstruction of a face based on a single or on multiple images. In an analysis by synthesis loop the variables of the Morphable Model, the lighting and the camera model are optimized by using a cost function to minimize the difference between a target image or video and the image that is generated by the illumination and projection of the model.
This paper is organized as follows. Section 2 gives a short introduction to the 3DMM and the cost function that is minimized for a single image and for multiple images. Since much valuable texture information is not contained within the Morphable Model we introduce a method to extract this texture from the target images within a Bayesian framework. In section 3, the experiments and the results are given. In the final section, the results are discussed.

2 Methods

The morphable face model [1] is a vector space representation of both the shape and the texture of faces. Principle Component Analysis is applied to \( m \) aligned 3D scans to create a face description based on the average face shape and face texture and the most likely variations. A face with identity \( i \) can then be described as a combination of its shape \( S_i \) and texture \( T_i \):

\[
S_i = \bar{s} + \sum_{j=1}^{m-1} \alpha_{j,i} \cdot s_j,
T_i = \bar{t} + \sum_{j=1}^{m-1} \beta_{j,i} \cdot t_j
\]

(1)

where \( \bar{s} \) and \( \bar{t} \) denote the mean shape and texture respectively, and \( s_j \) and \( t_j \) denote the eigenvectors. The identity can now be described in a lower dimensional subspace using the vectors \( \alpha_1, \alpha_2, \ldots, \alpha_{m-1} \) and \( \beta_1, \beta_2, \ldots, \beta_{m-1} \). We will refer to these as the shape \((\alpha_i)\) and texture vector \((\beta_i)\).

A 3D shape and texture can be projected to a 2D image using a rigid transformation, applying a lighting model, a colour correction and a perspective projection[2]. The goal of fitting a Morphable Model to an image is to find the vectors \( \alpha_i, \beta_i \) and simultaneously estimate the pose and the lighting parameters. The vectors \( \alpha_i, \beta_i \) can be used directly for identification and verification using for instance the cosine distance measure.

A Morphable Model is fitted to an input image in an analysis-by-synthesis loop, minimizing a cost function

\[
E = \arg \min_{\alpha, \beta, \rho} \frac{1}{\sigma_I^2} \sum_{k} \|I^m(k; \alpha, \beta, \rho) - I(x_k, y_k)\|^2 + \sum_{\forall i} \frac{\alpha_i^2}{\sigma_{S,i}^2} + \sum_{\forall i} \frac{\beta_i^2}{\sigma_{T,i}^2}
\]

(2)

The first part of this cost function describes the pixel error between the model and the input image. The parameters for the pose and lighting have been concatenated in \( \rho \) for brevity. The second part is for regularization. Since the parameters of the shape and texture vector are PCA coefficients, we assume they have a normal distribution with a mean of 0 and a variation equal to the eigenvalues. This regularization prevents overfitting by pulling it towards the mean face, but on the other hand can also cause underfitting. For the PCA Morphable Model we use the Basel Face Model (BFM) [3], which is based on 100 males and 100 females. The model has 53490 vertices and 106466 triangles.

The cost function in equation 2 will find the model parameters for a single image. If we want to fit the model to multiple images of the same individual, we simply sum the pixel difference error over all available images:

\[
E = \arg \min_{\alpha, \beta, \rho} \frac{1}{\sigma_I^2} \left( \sum_{\forall \text{images}} \sum_{k} \|I^m(k; \alpha, \beta, \rho) - I(x_k, y_k)\|^2 \right) + \sum_{\forall i} \frac{\alpha_i^2}{\sigma_{S,i}^2} + \sum_{\forall i} \frac{\beta_i^2}{\sigma_{T,i}^2}
\]

(3)
This analysis by synthesis method is computationally expensive. The speed of the fitting algorithm is most of the time increased by only optimizing equations 2 and 3 for a subset of the visible triangles in the target image. This subset is randomly chosen at each iteration. The used cost function is highly non-convex, which can result in fitting algorithms finding a local minimum instead of the global minimum. Figure 1 shows the cost landscape: the value for the first shape parameter $\alpha_1$ versus the value of the cost function. These figures show that there is much noise on the value of the cost function if fewer triangles are used for fitting. This can influence both the convergence and the quality of the fit.

![Figure 1: Cost landscape for $\alpha_1$ for varying number of triangles](image)

2.1 Texture backmapping

The texture of a Morphable Model $T_i$ lacks detail after fitting it to an image or to multiple images. A mole or scar can be a very distinctive biometric feature, but it is unlikely that a PCA model based on 200 faces is able to model this trait. By using the estimated shape $\hat{S}_i$ we can improve the texture detail by extracting it from the input image. For every model vertex that is visible in the input image we can undo the effects of lighting. For vertices that are not visible we stick to the estimate of the Morphable Model. The resulting refined Morphable Model can be used to generate a frontal image with controlled lighting conditions. This frontal image can then be used as input to a 2D FR algorithm. Let $T_{i,k}$ be the estimated texture of the $k$th vertex and assume that this vertex is visible in the input image at position $x, y$. The RGB-value at the location $x, y$ is written as $I_{x,y}$. The texture can now be calculated as follows, using the notation from [2]:

$$\hat{T}_{i,k} = \frac{M^{-1}(I_{x,y} - [o_r \ o_g \ o_b]^T) - \kappa_s \cdot L_{\text{dir}} \langle r_k, \hat{v}_k \rangle^\nu}{L_{\text{amb}} + L_{\text{dir}} \cdot (n_k, l)}$$

In these equations the matrix $M$ and the vector $[o_r \ o_g \ o_b]^T$ are used for colour correction, $\kappa_s \cdot L_{\text{dir}} \langle r_k, \hat{v}_k \rangle^\nu$ is the contribution of specular highlights, $L_{\text{amb}}$ is the ambient light contribution and $L_{\text{dir}}$ is the direct light contribution.

A specular highlight is a bright spot of light that can appear on a face if the normal at a point is oriented precisely halfway between the direction of incoming light and the camera viewing direction. When saturation occurs at such a point, it is not possible to calculate the original texture from equation 4. We assume that saturation occurs if two out of three colour channels have the maximum value of 255. At these points we don’t use texture backmapping, but we use the texture from the Morphable Model.
Figure 2(a) shows an input image and the fitted Morphable Model in 2(b). The result of texture backmapping and rendering a frontal view is given in figure 2(c).

![Figure 2: Texture backmapping](image)

If there is only a single input image, this method is straightforward. The texture might however not be uniquely defined if there are multiple input images in which the same vertex is visible. We propose a Bayesian-based method in which the normal of a vertex with respect to the camera viewing direction is used to determine the reliability of a pixel. The general idea is that a vertex with a normal pointing towards the camera can be more accurately estimated than a vertex with a normal under an angle.

By \( p(\hat{T}_k) \) we denote the prior probability of the RGB value of vertex \( k \). A uniform distribution can be used as prior, discarding the MM texture. This makes sense if the results from the MM are not reliable. The prior can also be learned and takes into account the texture from the MM. In order to calculate the posterior probability we need to learn the conditional probability density function \( p(z|\hat{T}_k) \). This measurement vector consists for each image of the estimated texture and the \( z \)-component of the normal of the vertex. The total measurement vector \( z \) does not have a fixed length, since it depends on the number of images in which a vertex is visible. Therefore we propose to estimate the conditional probability density function for each colour channel in which the vertex is visible by a Gaussian distribution:

\[
N(\hat{T}_k; f_{n_{k,z}}),
\]

in which \( n_{k,z} \) is the \( z \)-component of the normal of the vertex and \( \sigma_f \) is the standard deviation for a pixel with a normal pointing straight to the camera.

Using the Bayes rule and denoting by \( T_{k,n} \) the texture of the \( k \)th vertex in the \( n \)th image and assuming that we have only two images:

\[
p(\hat{T}_k, T_{k,1}, T_{k,2}) = p(\hat{T}_k) \cdot p(T_{k,1}, T_{k,2}|\hat{T}_k) = p(\hat{T}_k) \cdot p(T_{k,1}|\hat{T}_k) \cdot p(T_{k,2}|T_{k,1}, \hat{T}_k)
\]

and also

\[
p(\hat{T}_k, T_{k,1}, T_{k,2}) = p(\hat{T}_k|T_{k,1}, T_{k,2}) \cdot p(T_{k,1}, T_{k,2})
\]

Combining these two equations yields the desired posterior pdf:

\[
p(\hat{T}_k|T_{k,1}, T_{k,2}) = \frac{p(T_{k,1}|\hat{T}_k) \cdot p(T_{k,2}|\hat{T}_k) \cdot p(\hat{T}_k)}{p(T_{k,1}) \cdot p(T_{k,2})}
\]

Both terms in the denominator are constants which reduces this equation to:

\[
p(\hat{T}_k|T_{k,1}, T_{k,2}) \propto p(T_{k,1}|\hat{T}_k) \cdot p(T_{k,2}|\hat{T}_k) \cdot p(\hat{T}_k)
\]
We assume a uniform distribution for $p(\hat{T}_k)$ and maximize the previous equation by minimizing the negative of its logarithm

$$\hat{T}_k = \arg \min_{\hat{T}_k} \left(- \ln p(T_{k,1}|\hat{T}_k) - \ln p(T_{k,2}|\hat{T}_k)\right)$$  \hspace{1cm} (9)

It can be shown that for $N$ images the minimization can be written as

$$\hat{T}_k = \arg \min_{\hat{T}_k} \left(- \sum_{n=1}^{N} \ln p(T_{k,n}|\hat{T}_k)\right)$$  \hspace{1cm} (10)

Figure 3 shows the pdf's of the texture (RGB channels) of the tip of a nose based on 6 images and $\sigma_f = 10$. Using these technique we can combine the images in figure 4(a) and 4(b) to a combined frontal view as shown in figure 4(c).

Figure 3: pdf’s of the nose tip texture

![Figure 3: pdf’s of the nose tip texture](image)

(a) −15° view  (b) +15° view  (c) Backmapping

Figure 4: Texture backmapping

### 3 Experiments and Results

The verification experiments were performed on a subset of 49 people in the CMU MultiPie dataset, including 6 different views for each individual. These views differ in yaw rotation $-45^\circ, -30^\circ, -15^\circ, 0^\circ, 15^\circ, 30^\circ$, in which $0^\circ$ is the frontal view, see figure 5. The inter-ocular distance in the frontal views is 73 ($\pm 3.5$) pixels. For each image or set of images we fit the Morphable Model to obtain the shape and texture parameters. These parameters are used as feature vectors. The normalized similarity score ($\in [0, 1]$) between two feature vectors $f_1, f_2$ is calculated by:
The similarity scores between the feature vectors of images are divided into two sets. The first set is the genuine set with scores for feature vectors that came from images with the same identity and the second set is an impostor set with scores for feature vectors that came from images with different identities. The genuine and impostor scores are used to calculate the Receiver Operating Characteristic and from this we extract the equal error rate (EER) as scalar performance measure.

\[
\frac{f_1 \cdot f_2^T}{2||f_1|| \cdot ||f_2||} + \frac{1}{2}
\]  

(11)

The similarity scores between the feature vectors of images are divided into two sets. The first set is the genuine set with scores for feature vectors that came from images with the same identity and the second set is an impostor set with scores for feature vectors that came from images with different identities. The genuine and impostor scores are used to calculate the Receiver Operating Characteristic and from this we extract the equal error rate (EER) as scalar performance measure.

![Figure 5: 6 views from the dataset](image)

### 3.1 Single vs Single image for varying number of triangles

In section 2 we have indicated that the number of triangles used in the fitting process can influence the quality of the fit. In this experiment we varied the number of triangles used for fitting and we also investigated the influence on the performance of both the shape and texture individually. Table 1 shows the EER’s for this experiment.

<table>
<thead>
<tr>
<th>#tri</th>
<th>All vs All</th>
<th>0 vs -15</th>
<th>0 vs -30</th>
<th>0 vs -45</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>s</td>
<td>t</td>
<td>s + t</td>
<td>s</td>
</tr>
<tr>
<td>2000</td>
<td>32.4</td>
<td>20.4</td>
<td>21.4</td>
<td>12.2</td>
</tr>
<tr>
<td>10000</td>
<td>22.7</td>
<td>16.6</td>
<td>16.7</td>
<td>12.2</td>
</tr>
<tr>
<td>40000</td>
<td>16.2</td>
<td>19.9</td>
<td>15.4</td>
<td>8.2</td>
</tr>
<tr>
<td>100000</td>
<td>16.1</td>
<td>23.0</td>
<td>14.4</td>
<td>10.2</td>
</tr>
</tbody>
</table>

Based on these results we conclude that there is a clear performance improvement going from 2,000 triangles to 10,000 triangles. This improvement is especially visible in the 0 vs -30 and the 0 vs -45 experiments. In the All vs All experiment the performance gain between 2,000 and 100,000 triangles in the s+t experiment is 7% (21.4% → 14.4%). The results in the experiments with 40,000 and 100,000 triangles suggest that the texture is overfitted if using many triangles, e.g. the EER in All vs All increases 3.1% (19.9% → 23.0%). This is a problem that occurs often in PCA based approaches.

### 3.2 Single image vs Multiple images

In the single image vs multiple images experiment we use multiple images under a non-frontal angle to fit a Morphable Model and compare these feature vectors to those of a single frontal image. The results are given in table 2. The results show, as expected, an improvement with respect to the results in table 1. The EER of the s+t experiment with images at a view of ± 30° and 2,000 triangles is 14.2%, compared to 24.5% if only the −30° image is used. Combination of multiple images with different views yields better results.
Table 2: EER of single images with different number of triangles used for fitting

<table>
<thead>
<tr>
<th>#triangles</th>
<th>0 vs ±15</th>
<th>0 vs ±30</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>t</td>
<td>s+t</td>
</tr>
<tr>
<td>2000</td>
<td>10.2</td>
<td>4.1</td>
</tr>
<tr>
<td>10000</td>
<td>8.2</td>
<td>4.1</td>
</tr>
<tr>
<td>40000</td>
<td>10.2</td>
<td>6.1</td>
</tr>
</tbody>
</table>

3.3 Sensitivity to landmarking

The results obtained in the previous sections are based on manually landmarking. According to the results in [4] automatic landmarking methods have an error of 4% of the inter-ocular distance. We therefore added Gaussian noise with $\sigma = 3$ pixels to the position of the manually labelled landmarks to investigate the robustness of Morphable Model fitting in systems where an automatic landmarking method would be used. The average EER in the previous experiments increased with 1.2%.

3.4 Texture backmapping

After applying texture backmapping we can render a frontal view under controlled lighting conditions and use this as an input to a 2D FR system. For the experiments we use the FaceVACS B5 algorithm to calculate matching scores. According to the technical specs of FaceVACS-SDK 8.4, the face recognition engine is robust against pose variations up to 15 degrees off the frontal pose. The following experiments have been conducted:

1. **Baseline**
   The baseline FaceVACS B5 results are obtained by directly applying the 2D FR software to the images, so without any pose correction or Morphable Models. For a number of images (with a pose $> 15^\circ$) the B5 algorithm was not able to locate the eyes. These images have been regarded as failure to enroll (FTE) and are not considered in the calculation of the EER.

2. **Average face shape, without MM**
   The average face shape has been used to correct the pose of an image to render a frontal view. Pixels that are missing to reconstruct a full frontal view have been left black.

3. **Average face shape, with MM**
   The average face shape has been used to correct the pose of an image to render a frontal view. Missing pixels in the frontal view have been replaced with the average texture from the model.

4. **With fitted MM**
   A MM has been fitted to the image and texture backmapping has been applied.

5. **Multiple input images**
   The MM has been fitted to multiple images simultaneously and backmapping using the Bayesian framework has been applied.

The results of these experiments are summarized in table 3. The experiments show how a state-of-the-art 2D face recognition system can benefit from a Morphable Model, especially for faces under an angle of $\geq 30^\circ$. The baseline 2D FR software achieves an EER of 4.2% on the single view (0 vs -30). Using a combination of multiple views, texture backmapping and the 2D FR software we have achieved a 0% EER.
Table 3: EER (%) with texture backmapping

<table>
<thead>
<tr>
<th>Experiment</th>
<th>FTE (%)</th>
<th>All vs All</th>
<th>0 vs -15</th>
<th>0 vs -30</th>
<th>0 vs -45</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. baseline</td>
<td>3.1</td>
<td>16.9</td>
<td>2.1</td>
<td>4.2</td>
<td>17.9</td>
</tr>
<tr>
<td>2. avg. shape, no MM</td>
<td>3.4</td>
<td>13.7</td>
<td>2.1</td>
<td>2.1</td>
<td>15.0</td>
</tr>
<tr>
<td>3. avg. shape + MM</td>
<td>0</td>
<td>10.5</td>
<td>2.1</td>
<td>2.1</td>
<td>11.8</td>
</tr>
<tr>
<td>4. with fitted MM</td>
<td>0</td>
<td>6.3</td>
<td>2.1</td>
<td>2.1</td>
<td>5.9</td>
</tr>
<tr>
<td>5. multiple input images</td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

4 Discussion

In this paper we have introduced a Bayesian-based method for texture backmapping to optimally combine the information from multiple images in order to generate a frontal view, which can be used in 2D FR systems. We have also shown the influence of the number of triangles used for fitting a Morphable Model on the verification performance of a face recognition system. The number of triangles needed for fitting might be decreased if the cost function can be made less non-convex and less noisy [5]. Fitting a model to multiple images under different angles simultaneously yields a higher verification performance. In future work we will explore the possible benefits of this approach for low-resolution images.

5 Acknowledgement

We would like to thank Cognitec Systems GmbH for supporting our research by providing the FaceVACS software. Results obtained for FaceVACS were produced in experiments conducted by the University of Twente, and should therefore not be construed as a vendor’s maximum-effort full-capability result.

References


