3D Morphable Model Fitting For Low-Resolution Facial Images

Pouria Mortazavian p.mortazavian@surrey.ac.uk
Josef Kittler j.kittler@surrey.ac.uk
William Christmas W.Christmas@surrey.ac.uk
Centre for Vision, Speech and Signal Processing, University of Surrey, United Kingdom.

Abstract

This paper proposes a new algorithm for fitting a 3D morphable face model on low-resolution (LR) facial images. We analyse the criterion commonly used by the main fitting algorithms and by comparing with an image formation model, show that this criterion is only valid if the resolution of the input image is high. We then derive an imaging model to describe the process of LR image formation given the 3D model. Finally, we use this imaging model to improve the fitting criterion. Experimental results show that our algorithm significantly improves fitting results on LR images and yields similar parameters to those that would have been obtained if the input image had a higher resolution. We also show that our algorithm can be used for face recognition in low-resolutions where the conventional fitting algorithms fail.

1. Introduction

Since the pioneering work of Blanz and Vetter ([14], [3], [4]), 3D morphable face models have been at the centre of attention in many research areas involving human faces. For many applications it is necessary to estimate the parameters of the model from a single 2D image, a process known as model fitting. The aim of model fitting is to estimate the parameters of a 3D morphable face model such that the model would represent the input 2D face image. The process includes estimating the 3D shape and texture of the given face. Different approaches have been proposed for this challenging task (e.g. [4], [10], [9]).

The problem of fitting a 3D morphable model to a 2D image is an inverse, ill-posed problem. It is generally approached by minimising an energy function designed to model the error between the model's appearance and the input image in a maximum a posteriori (MAP) estimation framework. In general, such an energy function would be highly non-convex with numerous local minima, and to make things worse, the ambiguity of the problem increases in low resolutions due to the lack of information in the input images.

Low resolution faces lose a significant amount of useful information due to many factors including the optical blur caused by camera optics, finite density of CCD elements, and noise from various sources. Most applications involving low-resolution images deal with these problems by explicitly modelling the image formation process. For instance, in the super-resolution work of [2], the problem of constructing a high-resolution (HR) image from a number of low-resolution (LR) images is approached by constraining the result such that the sought HR image would yield the LR observations through an image formation model. In another example which deals specifically with low-resolution faces, Dedeoglu *et al.*[5] formulated a method for fitting Active Appearance Models to LR faces by modelling the low-resolution image formation process.

We approach the problem of fitting a 3D morphable model to a low-resolution face image in a similar manner by modelling the low-resolution image formation process. We start by critically analysing two of the main fitting algorithms proposed in the literature [3], [10]. By comparing the image synthesis model assumed by these conventional fitting algorithms with the continuous image formation model, we argue that although these algorithms work well for high-resolution inputs, their application becomes less relevant as the resolution of the input image decreases. We then propose an alternative imaging model which takes into account the point-spread function of the virtual camera and use this model in the fitting algorithm. Experimental results show that incorporating this imaging model into the fitting algorithm improves performance for low-resolution images.

2. Background

2.1. 3D Morphable Model

A morphable model can be constructed from a collection of 3D face scans each represented by a shape ($\mathbf{S}_i$) and a texture ($\mathbf{T}_i$) vector. Any face in the collection as well as any new 3D face can be synthesised as a linear combination

of the faces in the collection.

$$\mathbf{S}_{new} = \sum_i a_i \mathbf{S}_i \quad , \quad \mathbf{T}_{new} = \sum_i b_i \mathbf{T}_i \qquad (1)$$
$$(\sum_i a_i = \sum_i b_i = 1)$$

where all the 3D scans are assumed to be in dense point-to-point correspondence such that for any given j, the j^{th} vertex corresponds to the same location on all face scans. The face scans used by Blanz and Vetter are represented in cylindrical coordinates and the authors introduced a dedicated optical flow based algorithm to put these scans in dense correspondence. Tena *et al.*[13], [7] proposed an alternative method called the *Iterative Multi-Resolution Dense 3D Registration* (IMDR) which they used for building a 3D morphable model [7], [12].

Once the correspondence between the vertices of all scans is established, Principal Components Analysis (PCA) is performed separately on shape and texture vectors to decorrelate the data:

$$\mathbf{S}_{mod} = \bar{\mathbf{S}} + \sum_{i=1}^{N_S} \alpha_i \mathbf{s}_i \qquad \mathbf{T}_{mod} = \bar{\mathbf{T}} + \sum_{i=1}^{N_T} \beta_i \mathbf{t}_i \qquad (2)$$

where $\bar{\mathbf{S}}$ and $\bar{\mathbf{T}}$ are the mean shape and texture values respectively, $\mathbf{s}_i$ and $\mathbf{t}_i$ are the i^{th} eigenvectors of the shape and texture covariance matrices respectively, and N_S and N_T are the number of shape and texture eigenvalues to be used. Also, α_i and β_i are the mixture coefficients known as the model's shape and texture parameters respectively.

The model used in our work is built using using the methodology of [7]. The face scans are obtained using a 3dMD$^{\text{TM}}$sensor. The 3D meshes are registered using the IMDR algorithm.

2.2. Model Fitting

Blanz and Vetter also proposed a method for fitting the 3D morphable model to a given 2D face image [4]. Given a 2D face image, I^{inp}, and a set of manually defined landmarks, L, the fitting algorithm estimates the model parameters, α and β, together with a set of projection parameters, ρ, and a set of illumination parameters, γ, such that rendering the model with the estimated parameters will produce an image which resembles the input image. The projection parameters include 3D rotations and translations, and the focal length of a virtual camera. The illumination parameters include the parameters of the Phong illumination model such as light intensity, direction etc.

The synthesised image is supposed to be closest to the input image in terms of Euclidean distance:

$$E^I = \sum_{\mathbf{m}} \left\| I^{inp}(\mathbf{m}) - I^{mod}(\mathbf{m}) \right\|^2 \qquad (3)$$

where $\mathbf{m} \in \mathbf{Z}^2$ is the 2D coordinates of each pixel. Also, the error associated with the manually defined landmarks is expressed as $E^F = \sum_j \left\| \mathbf{q}_j - \mathbf{x}_{k_j} \right\|^2$ where $\mathbf{q}_j$ is the position of the j^{th} landmark and $\mathbf{x}_{k_j}$ is the image position of its corresponding vertex (k_j). The fitting procedure is then formulated as maximising the posterior probability:

$$\{\alpha^*, \beta^*, \mathbf{\Delta}^*\} = \underset{\alpha,\beta,\mathbf{\Delta}}{\operatorname{argmax}} \left\{ p(\alpha, \beta, \mathbf{\Delta} | I^{inp}, L) \right\}$$

$$= \underset{\alpha,\beta,\mathbf{\Delta}}{\operatorname{argmax}} \left\{ p(I^{inp} | \alpha, \beta, \mathbf{\Delta}) p(L | \alpha, \beta, \mathbf{\Delta}) p(\alpha) p(\beta) p(\mathbf{\Delta}) \right\} \qquad (4)$$

where $\mathbf{\Delta} = \{\rho, \gamma\}$ is the set of projection and illumination parameters. The priors $p(\alpha)$ and $p(\beta)$ are given by PCA, and a Gaussian distribution is assumed for $p(\mathbf{\Delta})$ with ad hoc values for the means, $\bar{\mathbf{\Delta}}_i$, and standard deviations $\sigma_{\mathbf{\Delta},i}$.

For independent, identically distributed Gaussian pixel noise with standard deviation σ_I, we have $p(I^{inp}|\alpha, \beta, \mathbf{\Delta}) \propto exp\left(\frac{-1}{2\sigma_I^2} E^I\right)$. Furthermore, the locations of the landmarks are assumed to be affected by Gaussian noise, so: $p(L|\alpha, \beta, \mathbf{\Delta}) \propto exp\left(\frac{-1}{2\sigma_F^2} E^F\right)$. Finally, model fitting is performed by minimising:

$$E = \frac{1}{\sigma_I^2} E_I + \frac{1}{\sigma_F^2} E_F + \sum_i^{N_S} \frac{\alpha_i^2}{\sigma_{S,i}^2} + \sum_i^{N_S} \frac{\beta_i^2}{\sigma_{T,i}^2} + \sum_i \frac{(\Delta_i - \bar{\mathbf{\Delta}}_i)^2}{\sigma_{R,i}^2} \qquad (5)$$

The above fitting criterion is non-convex and suffers from a large number of local minima. Hence, in order to reduce the risk of being stuck in local minima a stochastic optimisation method is used.

In [10] Romdhani and Vetter proposed an alternative fitting framework which in addition to the input image and landmarks, uses multiple features extracted from the input image. The objective function obtained this way is generally smoother and has fewer local minima. This *Multi-Features Fitting* (MFF) algorithm maximises the posterior probability $p(\theta | f^1(I^{inp}), f^2(I^{inp}), ..., f^N(I^{inp}))$ where θ is the set of all sought parameters and each $f^i(I^{inp})$ extracts a specific feature. Assuming the features are statistically independent and that deterministic feature extractors are used, it can be shown that maximising the posterior is equivalent to minimising:

$$-\ln p(f^1(I^{inp})|\theta) - ... - \ln p(f^N(I^{inp})|\theta) - \ln p(\theta) \qquad (6)$$

where each negative logarithm of a probability defines a cost function for a specific feature. The features used by Romdhani and Vetter include pixel colour, edges, landmarks, and specular highlights. In addition to these image-based features and the cost corresponding to prior probabilities of model parameters, an additional cost is added to constrain the admissible range of texture values.

Among the various cost functions used in [8], we mainly focus on the pixel colour value cost function E^I which is

defined similarly to the fitting algorithm of [4] as the Euclidean distance between the synthesised and input images. We also present a novel edge cost function.

The synthesised image in the above two fitting algorithms at a given pixel, $\mathbf{m}$, can be expressed ([8]) in terms of model parameters as:

$$I^{mod}(\mathbf{m}) = T^c(\mathbf{u}; \alpha, \beta, \rho, \gamma) \circ P^{-1}(\mathbf{x}; \alpha, \rho) \qquad (7)$$

where $\mathbf{x}$ is the 2D location of the centre of pixel $\mathbf{m}$ and T^c is the model texture defined in the model's reference coordinates ($\mathbf{u}$), illuminated by the Phong reflectance model. P^{-1} denotes the inverse of a mapping $P(\mathbf{u}; \alpha, \rho) = \mathbf{x}$ that projects each point $\mathbf{u}$ from the model's reference coordinates to the 2D image coordinates. The operator '$\circ$' denotes composition with this mapping.

Equation 7 states that in order to obtain the value of a point on the image plane, the 2D coordinates, $\mathbf{x}$, of the point are projected to the model's coordinates and the value of the illuminated texture at the obtained point is taken as the image value at point $\mathbf{x}$. By investigating an imaging model, in the next section, we argue that this process is not suitable for modelling the image formation process in LR images. Hence, any objective function that uses such a synthesised image is not suitable for model fitting on low-resolution input images.

3. The Low-Resolution Problem

We consider the image formation model and argue that the conventional fitting criteria described previously are not suitable for low-resolution images since the imaging model they assume for the synthesised image becomes invalid as the resolution of the input image decreases.

Let $\mathbf{m} = (m, n) \in \mathbf{Z}^2$ be the pixel coordinates on the image plane of I. The continuous image formation model can be expressed as [6]:

$$I(\mathbf{m}) = (E * PSF)(\mathbf{x}) = \int_{\mathbf{X}} E(\mathbf{x}).PSF(\mathbf{x} - \mathbf{m})d\mathbf{x}, \qquad (8)$$

where$E(.)$ is the continuous irradiance light-field that would reach the image plane under the pinhole model, $PSF(.)$ is the point spread function of the virtual camera, and the integral is taken over $\mathbf{x} = (x, y) \in \mathbf{R}^2$ which is the continuous pixel coordinates on the image plane.

PSF can further be decomposed into two components:

$$PSF(\mathbf{x}) = (w * a)(\mathbf{x}), \qquad (9)$$

where w models the optical blur and a models the spatial integration performed by the virtual CCD sensors. In the simplest case, the optical blur can be neglected ($w(\mathbf{x}) = \delta(\mathbf{x})$). Furthermore, we assume that the CCD elements are square with side length s. Hence, $a(\mathbf{x})$ takes the form:

$$a(\mathbf{x}) = \begin{cases} \frac{1}{A} & \text{if } |x| < \frac{s}{2} \ \& \ |y| < \frac{s}{2} \\ 0 & \text{o.w.} \end{cases} \qquad (10)$$

where $A = s^2$ is the area of each pixel.

The synthesised image used by the conventional fitting algorithms (eq. 7) is actually the irradiance, $E(\mathbf{x})$, sampled at the pixel centre locations of the image plane. Such algorithms neglect the effect of the convolution with the camera point spread function, effectively assuming $PSF(\mathbf{x}) = \delta(\mathbf{x})$. For a high resolution image, this assumption can be justified since the pixels can be assumed to have an infinitesimally small size ($A \rightarrow 0$). However, as the resolution of the image decreases, this assumption becomes less and less valid since the effect of the spatial integration becomes more and more significant making the fitting criterion (eq. 3) increasingly sub-optimal. Thus, for low-resolution inputs, the image formation model used to synthesise an image from the 3D morphable model must consider the spatial integration in order to synthesise a more realistic image.

In order to consider the effects of spatial integration in the imaging model, we need to consider the continuous irradiance field, $E(\mathbf{x})$, that will reach the image plane as opposed to the conventional imaging models which only sample this irradiance field at the location of the pixel centres.

The continuous irradiance field at the image plane can be obtained in two steps. First, each model vertex is projected to the image plane and its illuminated texture value is computed. The projection of the vertices is done through the projection P which includes a 3D rigid transform (rotation and translation) to map each point of the model's surface from object-centred coordinates to a position relative to the camera in world-coordinates, followed by a weak perspective projection which projects this point to the image plane. The illuminated texture values are computed using Phong's reflection model. This gives the irradiance value at certain points over the image plane which correspond to the projected model vertices. The second step for calculating the continuous irradiance field is to compute the irradiance over the rest of image plane which can be performed by interpolating between the known irradiance values using the triangle structure of the model. However, for the purpose of synthesising a low-resolution image the variations of texture values within each triangle are negligible. Therefore, we assign a single texture value, $\hat{T}(k)$, to each triangle defined as the average texture of its vertices. Thus, the triangle structure of the model together with the irradiance values assigned to each triangle yield a continuous, and piece-wise constant irradiance field over the whole face area of the image plane.

Given this piece-wise constant irradiance field, the imaging model of Equation 8 can then be estimated by a weighted average of the illuminated texture values of each

triangle whithin a given pixel:

$$I^{mod}(\mathbf{m}) = \frac{1}{A}\sum_{k\in\mathcal{K}} \hat{T}(k).W(k,\mathbf{m}) \tag{11}$$

where $\mathcal{K}$ is the set of triangles which after projection to the image plane overlap with pixel $\mathbf{m}$, and $W(k,\mathbf{m})$ is the area of overlap between pixel $\mathbf{m}$ and the k^{th} triangle after it is projected to the image plane.

Equation 11 defines an imaging model for synthesising an image from the model considering the spatial integration over the low-resolution pixels. We use this model to formulate a suitable criterion for fitting a 3D morphable model on LR images.

4. Fitting Algorithm for Low-Resolution

In addition to the sub-optimality of the cost function for low-resolution images, the optimisation methods used for optimising the cost function in the fitting algorithms of [7] and [4] is not suitable for the low-resolution case. Assuming that the contribution of all pixels of the image to the overall cost are redundant, these methods used a stochastic optimisation method which only evaluates the cost over a small number of vertices in each iteration. The reason for using the stochastic optimisation was to gain efficiency and avoid local minima at the cost of limited convergence properties (such as convergence radius).

For a low-resolution input however, the initial assumption of redundant contribution from image pixels is no longer valid. In fact, due to the lack of information in a low-resolution image it is crucial to make sure all available information is used. Hence, we do not use a stochastic optimisation algorithm. Instead, we deal with the problem of local minima by using a multi-feature fitting strategy similar to that of [10]. Due to using multiple features the overall cost function in this framework is smoother and a stochastic optimisation is not necessary to avoid local minima ([8]). Levenberg-Marquadt optimisation can therefore be used in order to optimise the cost function.

We use the landmarks, image edges, and pixel colour values as features in the MFF framework. A cost function is associatedd with each of these features. Furthermore, we use two cost functions to account for the shape and texture priors. The landmark cost function and the prior costs in our method are similar to the conventional MFF algorithm (See [8] for more details). However, for the edge cost and pixel colour features, we use novel cost functions described in the following.

4.1. Edge Cost Function

The aim of the edge cost function is to maximise the likelihood $p(f^e(I)|\alpha,\rho)$. Romdhani and Vetter used a deterministic edge detector (Canny) in order to obtain the edge map of the input image. The distance transform of this edge map was then used as a cost surface to evaluate the edge cost function defined as:

$$-\ln p(f^e(I)|\alpha,\rho) \propto C^e = \mathbf{e}^{e^T}.\mathbf{e}^e \tag{12}$$
$$\text{with} \quad \mathbf{e}^e_i(\alpha,\rho) = \|\mathbf{q}^e_i - \mathbf{p}_i\|$$

where $\mathbf{q}^e_i$ is the 2D coordinates of the i^{th} edge point of the input image, and $\mathbf{p}_i$ is the 2D location of its corresponding model edge point. However, using a single edge detector with a fixed set of parameters (eg. thresholds) doesn't always recover suitable edges in real-world images without tuning its parameters. Noting this, in [1], Amberg *et al.*constructed a smooth silhouette cost surface by integrating the information obtained from multiple edge detectors with different thresholds.

We use a similar approach and extend it into multiple resolutions of the input image. Given an input image, we first construct its Gaussian pyramid with l levels. If the input image is HR, we place it at the lowest level (highest resolution) of the pyramid and construct the higher levels (lower resolutions) by blurring and downsampling the original input. On the other hand, if the input is LR, we place it at a higher level of the pyramid -based on its resolution- and construct the lower levels by bilinear interpolation. In the next step, we use the Canny edge detector with a range of different thresholds, $\{th_1, th_2, \dots th_n\}$, to obtain multiple edge maps for each image in the pyramid and for each edge map we compute its distance transform. Finally, the edge cost surface, S, is constructed by integrating all the available distance transforms as: $S = \frac{1}{nl}\sum_{i=1}^{l}\sum_{j=1}^{n}\frac{D_i^{th_j}}{D_i^{th_j}+k}$, where $D_i^{th_j}$ is the distance transform of the i^{th} level of the pyramid obtained with the j^{th} threshold, and k is a smoothing constant and its value is approximately $\frac{1}{20}$th of the size of the input face. The cost surface constructed in this manner has the desirable property of having strong minima at locations where the edge is consistant over the range of resolutions and thresholds while still retaining weaker minima at the locations of weaker edges which are only detected by fewer combinations of resolution and threshold.

4.2. Colour Cost Function

As was argued in Section 3 the conventional pixel colour cost becomes sub-optimal in low resolutions. By replacing the conventional imaging model with our low-resolution imaging model (eq. 11), we improve the pixel colour cost function.

The pixel colour cost function aims to maximize the likelihood of the input image given the model, projection, and illumination parameters: $p(I^{inp}|\theta)$, where $\theta = \{\alpha,\beta,\rho,\gamma\}$, or equivalently minimising the (negative) log-likelihood $-\ln p(I^{inp}|\theta)$. By assuming that the image pixels are

affected by independent, identically distributed Gaussian noise, the pixel colour cost function can be expressed as:

$$-\ln p(I^{inp}|\theta) \propto \frac{1}{2}\sum_{\mathbf{m}}[I^{inp}(m) - I^{mod}((m))]^2 \quad (13)$$

where $I^{mod}(\mathbf{m})$ is given by equation 11. Unlike conventional fitting algorithms in which the pixel cost function is summed over the model vertices or polygons, in our formulation of the pixel colour cost function (eq. 13) the sum is taken over the pixels of the LR input image.

As mentioned previously, we do not use a stochastic optimisation method. This means all visible polygons of the model are used in every iteration for evaluating the pixel colour cost function and the sum in equation 13 is taken over all pixels of the LR face. Hence, our algorithm is computationally more expensive than the conventional MFF.

5. Experimental Results

We evaluate our theoretical framework on the the PIE dataset [11]. The LR images are synthesized by down-sampling the original images using differet down-sampling factors (DSF). We experiment using a subset of the the PIE dataset which covers different poses and illuminations. More specifically, we use pictures of all 68 available individuals in 2 poses: *frontal* and *side* (poses 27 and 5, respectively [1]), and 3 illumination conditions (illuminations 01, 02, and 13). In total the subset we use for our experiments contains 408 images covering a broad range of facial shape and texture, as well as different imaging conditions.

Figure 1 shows fitting results of a sample image with DSF ranging from 4 to 12. While the conventional fitting fails to recover detailed texture even for $DSF = 4$, our approach manages to recover a reasonable amount of the detail even in much lower resolution, notably outperforming the conventional method at lower resolutions.

Once the model fitting has been performed, one can use the recovered parameters to render the face in any desired resolution, pose, or illumination. Figure 2 shows an example of the model fitted on a LR image with $DSF = 8$, rendered at the resolution equvalent to $DSF = 2$, in other words, 4 times enlarged. For comparison, we aslo included results of the same rendering when the model was fitted using the conventional MFF algorithm, and bilinear interpolation. Note that the conventional algorithm has completely failed in fitting the model when the input resolution is very low (the input resolution in this figure is equivalent to the third column of Figure 1)

In the next experiment we compare the similarity of the parameters recovered from LR images with those recovered from HR images. We take the model parameters obtained using the conventional fitting on the original HR images ($DSF = 1$) as *ground truth* and measure the similarity of parameters recovered from lower-resolutions to these *ground truth* parameters. Note that this is a plausible choise of ground truth since a)The true 3D parameters for these real images are not known, b)The similarity of the HR parameters to the true 3D parameters is outside the scope of this paper and has been addressed in other works [], and c)In a realistic scenario, these HR parameters are the ones that would be used for most applications (eg. face recognition) where the input is a 2D image.

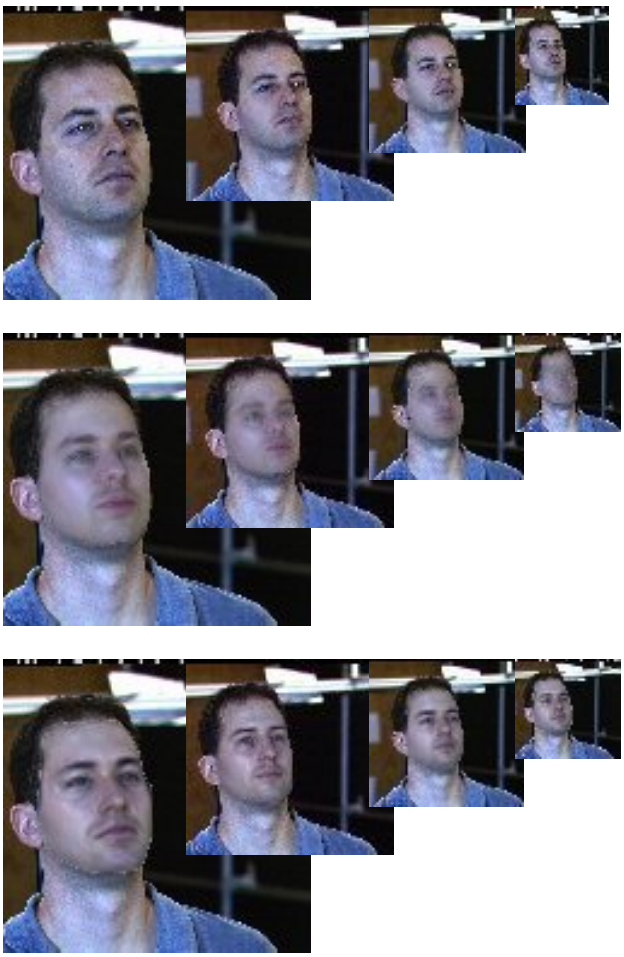

Figure 1. Comparison of model fitting on LR images. Each column shows from left to right images with DSF=4,6,8, and 12 respectively. Top row: original image, middle row: Model fitted using the conventional MFF algorithm. Bottom row: Model fitted using our LR-MFF apprach.

We measure the similarity in parameter space in terms of normalised correlation (NC) between the parameter vectors of the HR and LR image. We compare both shape (α) and texture (β) similarity in the parameter space. Figure 3 compares the similarity scores for shape and texture vectors obtained using our algorithm with those obtained using conventional MFF. It is clear from Figure 3 that our algorithm performs significantly better that the con-

[1] See [11] for details

(a) Original image at DSF=2

(b) Bilinear interpolation of DSF=8 to DSF=2

(c) Model Fitted using proposed LR-MFF on DSF=8. Rendered at the resolution of DSF =2

(d) Model Fitted using conventional MFF on DSF=8. Rendered at the resolution of DSF =2

Figure 2. Enlarging the face after fitting the model.

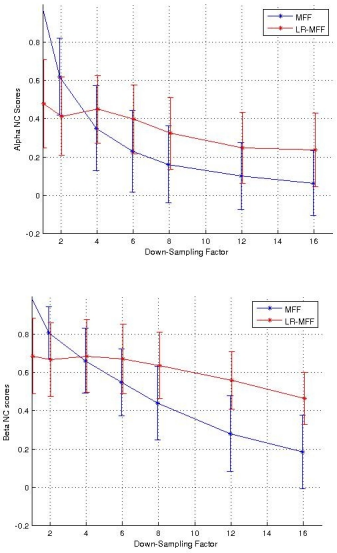

Figure 3. Similarity of the model parameters in the parameter space. 3(a): Shape similarity, 3(b): Texture similarity

ventional MFF in low resolutions. Note that in higher resolutions ($DSF < 4$), the conventional MFF outperforms our method. This is an expected observation since our algorithm is specifically designed for low resolutions where multiple polygons are projected to the same image pixel. In HR images where this assumption does not hold our algorithm performs worse than the conventional MFF due to its higher ambiguity. However, since during the early stages of the fitting it can easily be confirmed wether the input image is HR or LR (by consdiering the estimated focal length and distance to camera, or by measuring the average area of the projected polygons), it is straighforward to propose a hybrid algorithm which uses the conventional MFF for HR inputs and switches to LR-MFF if the input is LR.

Finally, we use our fitting algorithm for face recognition in low resolution. We Fit the morphable model on the frontal HR images ($DSF = 2$) with ambient illumination only (illumination 00 [2]) and take the optimised parameters as gallery. We then use both algorithms for fitting on all images in our subset and use the optimised parameters as probe. Similar to [10], in addition to the global model fitting, we fit 4 segments of the face, namely, the eyes, nose, mouth, and the rest of the face, separately in order to increase the descriptiveness of the model. The parameters obtained by the segmented fitting are stacked together with the global parameters to form a descriptive vectore. The most commonly used way of forming an identitiy vector from the model parameters is to stack all shape and texture parameters of both global and segmented models in one vector (Equation 14), each normalised by their respective standard deviation. This vector is then used it as the *Identity Vector* of the given face for identification (eg. see [10]).

$$V = [\frac{\alpha_1^g}{\sigma_{S,1}}, \ldots, \frac{\alpha_{n_\alpha}^g}{\sigma_{S,n_\alpha}}, \frac{\alpha_1^{s1}}{\sigma_{S,1}}, \ldots, \frac{\alpha_{n_\alpha}^{s1}}{\sigma_{S,n_\alpha}}, \ldots, \frac{\alpha_{n_\alpha}^{s4}}{\sigma_{S,n_\alpha}}, \ldots, \frac{\beta_1^g}{\sigma_{T,1}}, \ldots, \frac{\beta_{n_\beta}^g}{\sigma_{T,n_\beta}}, \frac{\beta_1^{s1}}{\sigma_{T,1}}, \ldots, \frac{\beta_{n_\beta}^{s1}}{\sigma_{T,n_\beta}}, \ldots, \frac{\beta_{n_\beta}^{s4}}{\sigma_{T,n_\beta}}] \quad (14)$$

where superscript g denotes the global model and superscripts $s1$ to $s4$ denote the 4 segments of the model, $\sigma_{S,i}$ and $\sigma_{T,i}$ denote the standards deviation of the i^{th} shape and texture parameters respectively, and n_α and n_β denote the number of α and β parameters, respectively.

Normalised correlation between the gallery and probe identity vectors is then used as a similarity score to identify the face.

We present identification results for two different settings corresponding respectively to pose, and illumination variations. Table 1 shows the rank-1 identification results for the pose variation setting where the gallery image is frontal with ambient illumination (pose 27, illumination 01) and the probe image is a non-frontal image with similar ilumination (pose 05, illumination 01). Table 2 shows these results for the illumination variation setting where

[2] see [11] for details

both gallery and probe are frontal but the probe image is subject to side-wise directional light in addition to the ambient light (pose 27, illumination 02). As was expected, our proposed method does not perform as well as the conventional MFF in high resolutions. However, its performance is considerably more stable over the whole range of resolutions and clearly outperforms the conventional MFF in low resolutions in both settings.

Table 1. Rank-1 identification results for the *pose variation* setting over a range of resolutions.

DSF	1	2	4	6	8	12	16
MFF	94	91	88	81	63	32	12
LR-MFF	84	84	85	88	78	66	44

Table 2. Rank-1 identification results for the *illumination variation* setting over a range of resolutions.

DSF	1	2	4	6	8	12	16
MFF	96	100	91	67	26	18	10
LR-MFF	93	93	99	93	90	69	43

6. Conclusions

The problem of fitting a 3D morphable face model on 2D images has been approached by a number of researchers and different algorithms have been proposed in the literature. We investigated this problem under the assumption that the input 2D image has a low resolution. We argued that the objective function used by the common conventional fitting algorithms becomes suboptimal in such a scenario since the image formation model used by these algorithms does not consider the effect of the virtual camera's point spread function implicitly assuming a continuous image. While this assumption has little impact for model fitting on HR images, it renders the fitting criterion invalid as the image resolution decreases.

We proposed a new image formation model given the model parameters that incorporates a camera model and takes into account the spatial integration over pixels which results in a more accurate modelling of the low-resolution image formation process. We showed that the fitting performance can be improved by incorporating this model into the fitting objective function. Experimental results show that the new fitting algorithm outperforms traditional algorithms in low-resolution scenarios in terms of visual quality and similarity of the obtained parameters to the HR parameters. Furthermore, we experimentally confirmed that our fitting is robust enough to be used for face recognition in low resolution with a relatively high performance.

7. Acknowledgement

This work was partially supported by EPSRC project ACASVA (Adaptive cognition for automated sports video annotation) under grant EP/F069421/1.

References

[1] B. Amberg, A. Blake, A. W. Fitzgibbon, S. Romdhani, and T. Vetter. Reconstructing high quality face-surfaces using model based stereo. In *ICCV*, pages 1–8, 2007.

[2] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):1167 – 1183, September 2002.

[3] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In A. Rockwood, editor, *Siggraph 1999, Computer Graphics Proceedings*, pages 187–194, Los Angeles, 1999. Addison Wesley Longman.

[4] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9):1063–1074, September 2003.

[5] G. Dedeoglu, S. Baker, and T. Kanade. Resolution-aware fitting of active appearance models to low-resolution images. In *Proceedings of the 9th European Conference on Computer Vision, ECCV 2006*, volume II, pages 83 – 97. Springer-Verlag, May 2006.

[6] V. S. Nalwa. *A guided tour of computer vision*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1993.

[7] J. T. Rodriguez. *3D Face Modelling for 2D+3D Face Recognition*. PhD thesis, Centre for Vision, Speech, and Signal Processing, University of Surrey, November 2007.

[8] S. Romdhani. *Face Image Analysis Using a Multiple Feature Fitting Strategy*. PhD thesis, Univ. of Basel, January 2005.

[9] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3d morphable model. In *ICCV*, pages 59–66, 2003.

[10] S. Romdhani and T. Vetter. Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *CVPR (2)*, pages 986–993, 2005.

[11] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):1615 – 1618, December 2003.

[12] J. Tena, R. Smith, M. Hamouz, J. Kittler, A. Hilton, and J. Illingworth. 2d face pose normalisation using a 3d morphable model. In *Proceedings of the International Conference on Video and Signal Based Surveillance*, pages 1–6, September 2007.

[13] J. R. Tena, M. Hamouz, A. Hilton, and J. Illingworth. A validation method for dense non-rigid 3d face registration. In *Proceedings of the IEEE International Conference on Advanced Video and Signal-based Surveillance*, November 2006.

[14] T. Vetter and V. Blanz. Estimating coloured 3d face models from single images: An example based approach. In *ECCV (2)*, pages 499–513, 1998.