3D Bilinear Face Model Fitting from Multiple Cameras

Christophe Ecabert, Hua Gao and Jean-Philippe Thiran [1]
[1] Signal Procesing Laboratory (LTS5), Ecole Polytechnique Fédérale de Lausanne, Switzerland

Abstract— **3D facial analysis attracts much interest recently due to the fact that it provides solutions for mitigating confounding factors in 2D image analysis, such as pose, illumination. On the other hand, it also provides enriched representation with more discriminative depth information for applications such as expression or identity analysis. In this paper, we investigate 3D face reconstruction based on sets of extracted facial features in a multi-view camera setup. The reconstruction is done using Bilinear Face Models where identity and expression are modelled independently in different modes. A novel algorithm based on full-perspective projection model is introduced. We validate our reconstruction method on synthetic data in this study. Experiments show that the reconstruction performance is significantly improved with multi-view inputs in terms of point-to-point error, normal error, as well as errors in model coefficients. We also show that the reconstruction method based on full-perspective projection model produces superior reconstruction accuracy comparing to weak-perspective model in multi-view reconstruction. The robustness of reconstruction against noise in the feature data is discussed and show the proposed method is applicable on real data.**

I. INTRODUCTION

Over the past years, scientists in computer vision field have developed a growing interest in modelling the face with 3D data. The motivation behind this interest is to improve facial analysis tool performance (*i.e. face recognition*) for non-trivial case such as non-frontal images, bad illumination condition. In this sense, Vetter *et al.* have presented the concept of 3D Morphable Model (3DMM), a statistical model where the 3D shape and the 2D texture are included [2]. It is an extension of the well known Active Appearance Model (AAM) [3] working with 3D shape scans instead of 2D. Such generative model is widely used in different type of application ranging from face recognition [5] to facial expression recognition and expression analysis [6]. The 3D face structure is reconstructed by matching the 2D image texture with the one generated by the model, which is also called analysis by synthesis similar to AAM fitting [7]. The optimisation uses a modified gradient descent algorithm and it is very slow; convergence time has been improved by the use of image features and specular highlights by Romdhani *et al.* [8]. For accurate surface reconstruction, 3DMM needs dense mesh and texture leading to large computational time and may not be suited for all systems. The processing time goes from 30 seconds [8] up to 5 minutes [5] producing highly accurate shape reconstruction and texture, given a reasonable initialisation.This method has a significant processing time therefore it can only be considered for application targeting animation or offline analysis.

The efficiency of 3DMM fitting has been improved using featured-based and model parameter regularisation method introduced in [11] by dropping the texture and reconstructing the 3D object through the object-image correspondence. Later that work was extended to a multi-view framework with weak-perspective projection model in [12]. Vlasic *et al.* goes even further by introducing Multilinear Morphable Models [4] that embed multiple face attributes such as identities, expressions and visemes (*i.e. speech-related mouth articulations*) into one single model. Multilinear models are powerful analysis tool because each attributes are modelled as its own, this decoupling gives precise information through model's parameters unlike linear one (*i.e. information about mixture of expression and identity for instance*).

In this work, we aim at reconstructing 3D human faces with variations in identity and expression. For this, a Bilinear 3D Morphable Model is applied where the identity and the expression are the two separate modes. We investigated different camera projection models for situation where cameras are calibrated or not in a single or multiple view setup. Experimental results show that multiple view system with a full-perspective projection model helps to improve the reconstruction accuracy especially for the expression.

The contribution of this work is three-fold: (a) We propose a solution for fitting a Bilinear 3D Morphable Model to multi-view image features with a coordinate-descent optimisation framework under weak-perspective projection model. This extends the work in [12] and [4] for recovering multiple modes of facial variations in a multi-view setup. (b) The full-perspective projection model is suggested for better reconstruction accuracy in multiple cameras setup. We present an iterative algorithm for accurate pose recovery with an effective initialisation based on back-projection. (c) We apply an adaptive feature data selection method for selecting plausible 2D feature points for reconstruction. This removes the points that are invisible in a 2D image to a specific camera due to self-occlusion, and thus are inconsistent to the corresponding vertices in the 3D model.

The rest of the paper is structured as follows : Section II provides details on how to build a Bilinear Face Model and fit it on images with one or multiple views with two types of projection model. In Section III we present the experiments, and the results will be discussed and finally Section IV will conclude this study.

II. METHODS

A. Notation

Through all this paper the following notation is used : Italic capital letter $\mathscr{T}$ denotes tensor, bold capital letter $\mathbf{M}$ denotes matrix, bold lower-case letter $\mathbf{v}$ denotes vector and italic lower-case letter s stands for scalar value. Operation on tensor such as mode-*n* product is denoted by $\mathscr{T} \times_n \mathbf{M}$ or $\mathscr{T} \times_n \mathbf{v}$.

B. Bilinear Face Models

Most of the Morphable Models used nowadays are linear. Building a model that is able to handle different type of variation such as identity and expression increase the complexity and leads to a more complicated model. This training issue can be avoided by moving toward multilinear 3D Morphable Model assuming that the identity and expression can separately parametrise the geometric variations [4].

The Facewarehouse database [1] recently introduced by Cao *et al.* includes 3D blendshapes of different identities in various expressions. The database holds a total of 47 expressions such as smiling, mouth-opening, winking, etc (Fig.1). Those blendshapes are available for 150 individuals aged between seven and eighty years old. The Bilinear Face Model used in this work is built using this set of blendshapes. All the face mesh from this database share the same topology, therefore they can be arranged into a rank-three tensor $\mathscr{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, where the first mode d_1 holds the vertex variation (*i.e. shape*), the second d_2, the identity and the last one d_3, the expression. To extract meaningfully information from the training data, the tensor is decomposed using the *N-mode Singular Value Decomposition* (*N-SVD*) [9]. The decomposition of the tensor $\mathscr{T}$ produces a *core* tensor $\mathscr{C}$ and N singular matrices $\mathbf{U}_n$ (*i.e.* $N = 3$) that rotate the mode space and can be expressed with the *N-mode* product as shown in (1).

$$\mathscr{T} = \mathscr{C} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_n \mathbf{U}_n \tag{1}$$

The singular matrix $\mathbf{U}_n$ is defined as the left singular matrix of the tensor $\mathscr{T}$ unfolded along the n^{th} mode. Hence computing the *N-SVD* of $\mathscr{T}$ yields to the computation of N matrix Singular Value Decomposition (*SVD*), one for each mode unfolded. Similarly to regular *SVD*, the variance inside the core tensor $\mathscr{C}$ is concentrated into one corner therefore dimension can be reduced by truncating the singular matrices without loosing much information (*i.e.* data compression). Knowing the N singular matrices, the core tensor is defined as the multiplication of the data tensor by each matrices along the corresponding mode :

$$\mathscr{C}_{reduced} \approx \mathscr{T} \times_1 \check{\mathbf{U}}_1^\top \times_2 \check{\mathbf{U}}_2^\top \cdots \times_n \check{\mathbf{U}}_n^\top . \tag{2}$$

The 3D surface needs a certain amounts of vertices to be precise enough therefore it does not make sense to truncate the data along this direction. Only dimension linked to other variation will be reduced (*i.e. identity or expression*) yielding to the following decomposition (3) where $\mathscr{C}_r = \mathscr{C} \times_1 \mathbf{U}_{vert}$.

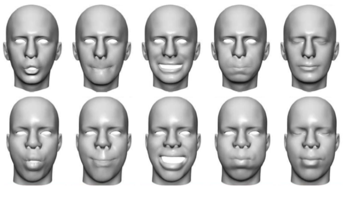


Fig. 1: FaceWarehouse expression examples

$$\mathscr{T} \approx \mathscr{C}_r \times_2 \check{\mathbf{U}}_{id} \times_3 \check{\mathbf{U}}_{expr} \tag{3}$$

This truncation gives generally good approximation but it is not optimal, using alternate least square method gives better approximation by refining $\check{\mathbf{U}}_n$ and $\mathscr{C}_{reduced}$ [9][10]. Multiplying the *core* tensor with singular matrices regenerates the original dataset (*i.e. set of faces*) therefore using the *N-mode* product and the corresponding i^{th} row of each singular matrix will generate one specific face from the training set. Moreover using a linear combination of rows from each $\check{\mathbf{U}}_n$ it is possible to generate an arbitrary face $\mathbf{f}$ as in (4) where $\mathbf{w}_{id}^\top$ and $\mathbf{w}_{expr}^\top$ are column vector of identity and expression weights.

$$\mathbf{f} = \mathscr{C}_r \times_2 \mathbf{w}_{id}^\top \times_3 \mathbf{w}_{expr}^\top \tag{4}$$

The *core* tensor $\mathscr{C}_r$ is what is called *Bilinear Face Model*.

C. Fitting Method

The feature-points based Morphable Model fitting method [11] is based on two key assumptions. The first one states that there is a direct correspondence between vertices (*i.e. Surface's 3D points*) and the features coming from the images through the projection model, and secondly it is possible to recover the model coefficient only using a subset of feature points (*i.e. sparse measurement*) [12]. Given a sparse measurement vector $\mathbf{r} \in \mathbb{R}^{2f}$ composed of f facial landmarks, (5) defines the relationship between the reconstructed 3D surface and the facial landmarks. Where $\mathbf{P}$ represents the sparse vertices selection going from N to f vertices ($\mathbf{P} : \mathbb{R}^{3N} \to \mathbb{R}^{3f}$) and $\mathbf{L}$ models the projection of this subset onto the image plane [11].

$$\mathbf{r} = \mathbf{LPf} = \mathbf{LP}\left(\mathscr{C}_r \times_2 \mathbf{w}_{id}^\top \times_3 \mathbf{w}_{expr}^\top\right) . \tag{5}$$

The selection operator $\mathbf{P}$ that selects a sparse subset of vertices will be explicitly defined in Section II-D and Section II-E gives more details on how the projection operator $\mathbf{L}$ can be estimated.

It is not possible to find a linear combination of shapes from training set that match exactly (5), therefore the error function defined in (6) is minimized to get the closest solution.

$$E\left(\mathbf{w}_{id}, \mathbf{w}_{expr}\right) = \left\|\mathbf{LP}\left(\mathscr{C}_r \times_2 \mathbf{w}_{id}^\top \times_3 \mathbf{w}_{expr}^\top\right) - \mathbf{r}\right\|^2 \tag{6}$$

This type of optimisation problem where two variables need to be estimated can be solved using coordinate-descent [4]. The recovery of the identity and expression weights is done considering one variation at a time and the other being fixed, this way the multilinear problem ends up in a linear one in a form of : $\underset{\mathbf{w}}{\arg\min} \ \|\mathbf{Qw} - \mathbf{y}\|^2$. However directly minimizing this problem will not produce perceptually correct result as shown in [11]. The reason is that it will only minimise the re-projection error between the 3D model and the 2D features without constraining $\mathbf{w}$ to be in the span of the 3D Morphable Model solution. A parameter regularisation based on statistical approach is added to ensure that the parameter $\mathbf{w}$ lays in the span of solution and represents a face. Therefore to recover the identity $\mathbf{w}_{id}$ and expression $\mathbf{w}_{expr}$ the optimisation problem (6) becomes :

$$E\left(\mathbf{w}_{id}\right) = \left\|\mathbf{LPM}_{expr}\mathbf{w}_{id}^{\top} - \mathbf{r}\right\|^2 + \eta_{id}\left\|\mathbf{w}_{id}\right\|^2, \quad (7)$$

$$E\left(\mathbf{w}_{expr}\right) = \left\|\mathbf{LPM}_{id}\mathbf{w}_{expr}^{\top} - \mathbf{r}\right\|^2 + \eta_{expr}\left\|\mathbf{w}_{expr}\right\|^2, \quad (8)$$

where $\mathbf{M}_{expr} = \mathscr{C}_r \times_3 \mathbf{w}_{expr}^{\top}$ and $\mathbf{M}_{id} = \mathscr{C}_r \times_2 \mathbf{w}_{id}^{\top}$. This problem stands for single image but can be easily extended to support multiple images. The coefficients of the Bilinear Face Model are not dependant on the viewpoint because they are characterising 3D object's shape and not its projection on the image plane. When looking at the object from a different angle, the only variation in the optimisation problem is the facial landmark location $\mathbf{r}$, the projection operator $\mathbf{L}$ and the sparse selection $\mathbf{P}$. Therefore if they are known it becomes easy to extend previous problem to multiple view with the cost function :

$$E\left(\mathbf{w}\right) = \frac{1}{K}\sum_{i=1}^{K}\left\|\mathbf{Q}_i\mathbf{w}^{\top} - \mathbf{r}_i\right\|^2 + \eta\left\|\mathbf{w}\right\|^2, \quad (9)$$

where $\mathbf{w}$ stands for $\mathbf{w}_{id}$ or $\mathbf{w}_{expr}$, $\mathbf{Q}_i$ for $\mathbf{L}_i\mathbf{P}_i\mathbf{M}_{expr}$ or $\mathbf{L}_i\mathbf{P}_i\mathbf{M}_{id}$ and η for η_{id} or η_{expr}. The coefficient $\frac{1}{K}$ ensures that all views have the same weight and balance the regularization [12]. The optimal solution $\mathbf{w}^*$ that minimises (9) can be found when its derivate with respect to $\mathbf{w}$ is null :

$$\begin{aligned} \frac{\partial E\left(\mathbf{w}\right)}{\partial \mathbf{w}} &= \frac{1}{K}\sum_{i=1}^{K}\left(2\mathbf{Q}_i^{\top}\mathbf{Q}_i\mathbf{w} - 2\mathbf{Q}_i^{\top}\mathbf{r}_i\right) + 2\eta\mathbf{w} = 0, \\ \mathbf{w}^* &= \left(\sum_{i=1}^{K}\left(\mathbf{Q}_i^{\top}\mathbf{Q}_i\right) + K\mathbf{I}\eta\right)^{+}\sum_{i=1}^{K}\left(\mathbf{Q}_i^{\top}\mathbf{r}_i\right). \end{aligned} \quad (10)$$

The operator $(\mathbf{A})^{+}$ denotes the pseudo-inversion of the matrix $\mathbf{A}$.

Solving (9) for $\mathbf{w}_{id}$ and $\mathbf{w}_{expr}$ into an iterative framework ensures that the face model parameters get closer to the optimal solution.

D. Landmarks Selection

The facial features used to reconstruct the 3D surface are automatically detected from the image with the help of a face tracker (*i.e. based on the supervised descent method* [16]). In this work, we assume that the face tracker provides a set of 68 facial landmarks such as eyes and mouth contour, chin, nose and eyebrows. Using such tool, some distortion can be introduced in particular when some part of the face is occluded. For instance when the face has a half-profile pose, part of the facial landmarks are not visible and even though they are hidden, the tracker places them on the edge of the face (*i.e. Fig. 2, red dots*). Therefore the given position does not match the real position and using this information in the reconstruction process will lead the introduction of distortion since it is assumed direct correspondence between 3D vertices and 2D facial landmarks. From the set of 2D points provided by the face tracker only landmarks corresponding to visible vertices from the cameras are used in the reconstruction to avoid to introduce such errors. This subselection of f features can be defined with the help of the 3D surface by determining whether or not a vertex is hidden behind it. Therefore the selection operator $\mathbf{P} \in \mathbb{R}^{3f \times 3n}$ that extracts the vertices corresponding to the facial landmarks can be define with the 3D surface and the previous pose estimation.

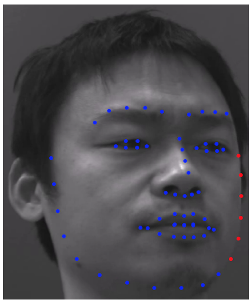


Fig. 2: Face landmarks given by the tracker, where blue dots are correctly matching the corresponding vertices and red do not match

E. Pose Estimation

In real world situation the face model is not aligned with the data therefore it is required to determined the rigid transformation that matches both dataset. This transformation is defined by a translation, a rotation and a scaling factor. In this work, two situations has been investigated where the cameras are un-calibrated and calibrated. From this configuration two types of projection models are studied, weak-perspective for un-calibrated case respectively full-perspective.

1) Weak Perspective: The weak-perspective projection model holds when object lays close to the optical axis and the depth variation is small against the distance to the camera. With a small variation in depth, it is assumed that all points stand onto the same plane located at an average distance to the camera. For human faces standing in normal situation in front of a camera, those constrains are fulfilled. The final transformation is given by (11) where $\mathbf{R}$ defines the orientation of the face relative to the camera, $\mathbf{t}$ defines the displacement on the image plane to align the points and s acts as scaling factor.

$$\mathbf{p} = s\mathbf{R}\mathbf{x} + \mathbf{t}. \quad (11)$$

In order to define the operator $\mathbf{L}$ the parameters s, $\mathbf{R}$ and $\mathbf{t}$ have to be estimated. To achieve this, the method presented in [11] has been used. The proposed algorithm uses extra basis added to the linear model to recover the unknown rigid transformation. Unfortunately with Bilinear Face Model and multiple views, it is not possible to recover the rigid and non-rigid parameters at the same time. However this basis technique can be used to recover the rigid transformation. The transformation is recovered by minimizing the projection error between basis and the facial landmarks. Using the 3D shape simple basis can be defined to estimate the transformation. The translation in the space can be retrieved using basis defined with unit vector corresponding to the correct direction : $\mathbf{s}_{tx} = (1,0,0,1,0,0,...,1,0,0)^\top$ and $\mathbf{s}_{ty} = (0,1,0,0,1,0,...,0,1,0)^\top$. The basis used for the scaling and the rotation are based on the 3D surface, typically the mean 3D shape coming form the training set. The rotation is assumed to be for small angles, $\gamma, \theta, \phi \ll 1$, therefore the rotation matrix can be simplified by considering the cosine terms equal to 1 and ignoring the product of sines. Finally the contribution of each angle can be separated from each other and the basis are define as in (12). The scaling factor uses the whole 3D mean shape as basis (13).

$$\begin{aligned} \mathbf{s}_\gamma &= (-\overline{y}_1, \overline{x}_1, 0, ..., -\overline{y}_n, \overline{x}_n, 0)^\top, \\ \mathbf{s}_\theta &= (0 - \overline{z}_1, \overline{y}_1, 0, ..., 0, -\overline{z}_n, \overline{y}_n)^\top, \\ \mathbf{s}_\phi &= (\overline{z}_1, 0, -\overline{x}_1, ..., \overline{z}_n, 0, -\overline{x}_n)^\top, \end{aligned} \tag{12}$$

$$\mathbf{s}_s = (\overline{x}_1, \overline{y}_1, \overline{z}_1, ..., \overline{x}_n, \overline{y}_n, \overline{z}_n)^\top. \tag{13}$$

Using the basis, the rigid transformation can be recovered for each view by minimising the error between the basis projection and the facial landmarks as defined in (14)

$$E(\mathbf{w}_p^{(i)}) = \left\| \mathbf{L}_i \mathbf{P}_i \mathbf{S} \mathbf{w}_p^{(i)} - \mathbf{r}_i \right\|^2, \tag{14}$$

where $\mathbf{S} = \left[\; \mathbf{s}_s \mid \mathbf{s}_\gamma \mid \mathbf{s}_\theta \mid \mathbf{s}_\phi \mid \mathbf{s}_{tx} \mid \mathbf{s}_{ty} \; \right]$ and $\mathbf{w}_p^{(i)} = \left[\; s \mid s \cdot \sin\gamma \mid s \cdot \sin\theta \mid s \cdot \sin\phi \mid t_x \mid t_y \; \right]$ is the rigid parameters for the i^{th} view. To have a better estimation of the parameters the minimisation is included in a multi-pass approach where the operator $\mathbf{L}_i$ is updated at each iteration to have a better update at the next pass. once the rigid transformation is estimated for each view, the projection operator $\mathbf{L}_i \in \mathbb{R}^{2f \times 3f}$ is define as block diagonal matrix where each block is define as

$$\mathbf{B} \in \mathbb{R}^{2\times 3} = s_i \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{R}_i, \tag{15}$$

where s_i and $\mathbf{R}_i$ correspond to the scaling factor and the object orientation for the i^{th} view. The translation factor $\mathbf{t}_i$ is directly applied on the input vector $\mathbf{r}_i$. The scaling factor s_i englobes the effect of the object's distance to the camera as well as the physical size of the object. The size of the head and the position can vary between each individual and images, therefore the re-projection error range will increase and the regularisation factor will not have the same effect. To overcome this issue, the input vector $\mathbf{r}_i$ needs to be normalized to cancel the effect of the scaling factor. The input normalization factor (20) is defined with the face model's projection (16), the projection's center of gravity (17), the input's center of gravity (18) and the centred landmarks (19) :

$$\mathbf{p}_i = \mathbf{L}_i \mathbf{P} \left(\mathscr{C}_r \times_2 \mathbf{w}_{id_{n-1}}^\top \times_3 \mathbf{w}_{expr_{n-1}}^\top \right), \tag{16}$$

$$\overline{\mathbf{p}}_i = (\overline{\mathbf{p}}_x, \overline{\mathbf{p}}_y, ..., \overline{\mathbf{p}}_x, \overline{\mathbf{p}}_y)^\top, \tag{17}$$

$$\overline{\mathbf{r}}_i = (\overline{\mathbf{r}}_x, \overline{\mathbf{r}}_y, ..., \overline{\mathbf{r}}_x, \overline{\mathbf{r}}_y)^\top, \tag{18}$$

$$\mathbf{r}_i^{'} = \mathbf{r}_i - \overline{\mathbf{r}}_i, \tag{19}$$

$$\alpha_i = \frac{\left\| \mathbf{p}_i - \overline{\mathbf{p}}_i \right\|}{\left\| \mathbf{r}_i^{'} \right\|}. \tag{20}$$

The cost function slightly changes as shown in (21). For the initialisation at the first pass, the 3D surface used is the mean shape coming from the training set, then it uses the shapes reconstructed at the previous iteration.

$$E(\mathbf{w}) = \frac{1}{K} \sum_{i=1}^{K} \left\| \mathbf{Q}_i \mathbf{w}^\top - \alpha_i \cdot \mathbf{r}_i^{'} + \overline{\mathbf{p}}_i \right\|^2 + \eta \left\| \mathbf{w} \right\|^2, \tag{21}$$

Algorithm 1 summarises all the operation performed by the reconstruction pipeline when the weak-perspective projection model is used. If the face tracker provides a rough estimation of the object's pose $\hat{\mathbf{R}}_0$ it can be used as initialisation.

2) Full Perspective: If we consider the full-perspective projection model, the coordinates on the image plane of the 3D points are defined by (22) when using homogene coordinates.

$$\lambda \mathbf{p} = \mathbf{K} \left(\mathbf{R}_{cam} \left(s \mathbf{R}_{obj} \mathbf{x} + \mathbf{T}_{obj} \right) + \mathbf{T}_{cam} \right) \tag{22}$$

The position onto the image plane depends on the characteristics of the camera (*i.e. focal length, image center*) called *intrinsic* parameters $\mathbf{K}$ and the location and orientation of the camera called *extrinsic* parameters (*i.e.* $\mathbf{R}_{cam}, \mathbf{T}_{cam}$). The *intrinsic* and *extrinsic* parameters can be recovered by calibrating the cameras.

In this case the rigid transformation is the same as earlier, it is composed of a scaling factor s, a rotation matrix $\mathbf{R}_{obj}$ and a translation vector $\mathbf{T}_{obj}$ (*i.e. orientation and position are absolute values*). The estimation of the unknown rigid transformation is based on the method presented in [13]. Their approach uses the information from each view at the same time and combine them to recover the pose, location and scale of the object. The calibration of each camera allows the consideration of the network of cameras as a generalized one and move from each referential (*i.e. one for each view*) to a global reference system. The camera's extrinsic parameter $\mathbf{T}_i = (\mathbf{R}_{cam}^{(i)}, \mathbf{T}_{cam}^{(i)})$ is used to fuse all information into a common reference system. Therefore it becomes possible to find the pose $\mathbf{T} = (s, \mathbf{R}_{obj}, \mathbf{T}_{obj})$ of the object by minimising the error between 3D object's

Algorithm 1 Calculate $\mathbf{w}_{id}, \mathbf{w}_{expr}$ using weak-perspective

```
for $it = 0$ to $it = M$ do
    if $it = 0$ then
        $\mathbf{t}_i = \overline{\mathbf{r}}_i$ (18), $s_i = \left\| \mathbf{r}_i - \overline{\mathbf{r}}_i \right\| / \left\| \overline{\mathbf{f}} \right\|$, $\mathbf{R}_i = \hat{\mathbf{R}}_0^{(i)}$ or $\mathbf{I}$ if no initial estimation
        Initialise $\mathbf{S}$ basis with meanshape, (12)(13)
        Initialise $\mathbf{L}_i$ and $\mathbf{P}_i$.
    else
        Update shape (4) and $\mathbf{S}$ basis with new shape $\mathbf{f}$
    end if
    Estimate pose (14), update $\mathbf{L}_i$, $\mathbf{P}_i$ and normalise $\mathbf{r}_i$, (15)(20)
    Estimate $\mathbf{w}_{id}^\top$, $\mathbf{w}_{expr}^\top$, (21)
end for
Update shape (4)
```

projection (*i.e. the 3D surface*) and the facial landmarks. The problem is define in (23), where only the valid points $\mathbf{P}_j$ (*i.e. vertex that are visible from the viewpoint i*) included in the vertices set corresponding to the facial landmarks $\mathbb{P}$ are projected onto the corresponding image plane using the full-perspective projection model.

$$T^* = \arg\min_{T} \sum_{i=1}^{K} \sum_{\mathbf{P}_j \in \mathbb{P}} \left[\mathbf{p}_j^i - proj(\mathbf{T}_i \mathbf{T} \mathbf{P}_j)\right]^2, \qquad (23)$$

The solution to this type of minimisation problem can be found using the Levenberg-Marquardt iterative solver [14].

Once the rigid transformation is known, the projection operator $\mathbf{L}_i$ can be defined for full-perspective projection model. The u and v coordinates of the points are given by the relation $u = f_x \cdot x/z$ and $u = f_y \cdot y/z$ which is not linear. However if the z coordinate is known, the computation of the image point coordinates become simpler and linear. A good estimation of this z value is given by the reconstructed surface at the previous iteration and for the initialisation the mean shape is used. With the previous surface reconstruction and the current rigid transformation, the new vertices position can be defined by (24) then a scaling factor can be computed for each vertex : $\beta_x^{(j)} = f_x / \mathbf{P}_z^{(j)}$ and $\beta_y^{(j)} = f_y / \mathbf{P}_z^{(j)}$.

$$\mathbf{P}^{(j)} = \left(\mathbf{R}_{cam}\left(s\mathbf{I}\mathbf{R}_{obj}\mathbf{x}^{(j)} + \mathbf{T}_{obj}\right) + \mathbf{T}_{cam}\right) \qquad (24)$$

Once those scaling factors are defined the projection operator $\mathbf{L}_i$ is defined as block diagonal matrix where each block is define as in (25) and a translation vector define as in (26) is also included into the optimisation process.

$$\mathbf{B}_i^{(j)} = \begin{bmatrix} s \cdot \beta_x^{(j)} & 0 & 0 \\ 0 & s \cdot \beta_y^{(j)} & 0 \end{bmatrix} \mathbf{R}_{cam}^i \mathbf{R}_{obj} \qquad (25)$$

$$\mathbf{t}_i^{(j)} = \begin{bmatrix} \beta_x^{(j)} & 0 & 0 \\ 0 & \beta_y^{(j)} & 0 \end{bmatrix} \left(\mathbf{R}_{cam}^i \mathbf{T}_{obj} + \mathbf{T}_{cam}^i\right) \qquad (26)$$

The cost function define in (9) is modified to included the changes to use the full-perspective projection model. However the effect of the head size (*i.e. the coefficient s*) will also have an impact on the regularisation parameters, therefore the input needs to be normalised to get ride of the effect. In case of single-view application, the input normalisation by ratio describe previously can be used without any modification. However this will introduced distortion since scaling 3D object will not lead to the same transformation onto the image plane. When working with multiple cameras configuration, the solution to avoid to introduce such deformations is to back-project the landmarks (*i.e. recover the 3D position of the landmarks from two distinct viewpoints*) and the normalise the size of the face using the scale estimated previously and then projecting back those points. Finally the cost function is :

$$E(\mathbf{w}) = \frac{1}{K} \sum_{i=1}^{K} \left\|\mathbf{Q}_i \mathbf{w}^\top - (\mathbf{r}_i'' - \mathbf{t}_i)\right\|^2 + \eta \left\|\mathbf{w}\right\|^2. \qquad (27)$$

Algorithm 2 Calculate $\mathbf{w}_{id}, \mathbf{w}_{expr}$ using full-perspective

```
for $it = 0$ to $it = M$ do
    if $it = 0$ then
        if #View $= 0$ then
            $\mathbf{T}_{obj} = \mathbf{0}$, $s = \|\mathbf{r}_i - \bar{\mathbf{r}}_i\| / \|\bar{\mathbf{f}}\|$, $\mathbf{R} = \hat{\mathbf{R}}_0$ or $\mathbf{I}$ if no initial estimation
        else
            $\mathbf{P} = Backproj(\mathbf{r}_i)$, $\mathbf{T}_{obj} = \overline{\mathbf{P}}$ and $s = \|\mathbf{P}\| / \|\bar{\mathbf{f}}\|$
        end if
    else
        Update shape (4) and $\mathbf{S}$ basis with new shape $\mathbf{f}$
    end if
    Update $\mathbf{P}_i$ and estimate pose, (23)
    Update $\mathbf{L}_i$ (25)
    if #View $= 0$ then
        Normalise $\mathbf{r}_i$, (20)
    else
        Normalise $\mathbf{r}_i$ using back-projection
    end if
    Estimate $\mathbf{w}_{id}^\top$, $\mathbf{w}_{expr}^\top$, (21)
end for
Update shape (4)
```

where $\mathbf{r}_i'' = \alpha_i \mathbf{r}_i'$ (20) for single camera or $\mathbf{r}_i'' = \left(BackProj(\mathbf{r}_i) - \mathbf{T}_{obj}\right)/s + \mathbf{T}_{obj}$ for multiple cameras setup.

Algorithm 2 gives a compact summary of the different steps included in the reconstruction process using the full-perspective projection model. To have a better pose estimation, the iterative optimiser is initialised too the solution. Using the information provided by the back-projection of the facial landmarks the position and the scale can be roughly approximated.

III. Results

A. Data and Experimental Setup

To assess the quality of the reconstructed face using a Bilinear Face Model, synthetic data have been used. The blendshapes used come from the FaceWarehouse database therefore they are part of the model training set. The facial expression database holds a total of 47 different expression composed of one neutral and 46 FACS blendshapes (*i.e. shape including combination of action units*) builded using facial rigging algorithm presented in [15].

The Bilinear Face Model is built using those blendshapes and has the following characteristics, 8260 vertices, 50 identity coefficients and 25 expression coefficients.

The image formation process is simulated using the full-perspective projection model with real calibration information in order to simulate realistic situation. The calibration data come from a camera rig where three cameras are in triangle on the structure. The triangle's top side include two cameras horizontally placed at 50 centimetres apart both aiming at the same point in front of them. The last one stands 45 centimetres below the two others and right in the middle of them. The configurations tested are single-view where only the camera in the middle is used, two views using cameras from both side and three views where all of them are used.

For real world application case, all the facial landmarks are provided by a face tracker. To simulate the same behaviour on synthetic data, zero mean gaussian noise is added to the vertices projection. The standard deviation is adjusted according to the eye distance to ensure that the same amount of noise is added in every case, the range goes from 0% up to 16%.

The testing set is composed of unique *identity* and *expression* pairs randomly picked blendshapes from the FaceWarehouse database. The orientation of the face is uniformly distributed between $[-30°, 30°]$ for every angle, the size is picked in the range between $[90, 110]$, the position is uniformly selected from a volume standing where all cameras are in focus with a size of $[-10, 10] \times [-10, 10] \times [-10, 10]$ centimetres.

Fig. 3 shows the evolution of identity and expression parameter update over the iteration, at the beginning both projection model give significant parameters update (*i.e. identity and expression*). However the gain after four iterations becomes much smaller therefore doing more iteration does not bring a significant improvement therefore the maximum number of iteration for the reconstruction pipeline is set to 4.

The reconstruction quality is analysed with three quality estimators, the average normalised point-to-point error per vertex $E_{pts} = \frac{1}{N}\sum_{i=1}^{N} \left\| \mathbf{x}^{(i)} - \mathbf{x}^{*(i)} \right\| / \left\| \mathbf{x}^{*} - \overline{\mathbf{x}}^{*} \right\|$, the average normal difference defined as $E_n = \frac{1}{K}\sum_{i=1}^{K} 1 - \left| \mathbf{n}_{rec}^{(i)} \cdot \mathbf{n}_{org}^{(i)} \right|$ and the norm of the identity and expression weights error $\|\mathbf{w} - \mathbf{w}^{*}\|$ where $\mathbf{w}$ is the estimated parameter for the identity or expression and $\mathbf{w}^{*}$ is the corresponding ground truth extracted from singular matrices $\mathbf{U}_{id/expr}$. The region of interest is the face and not the whole head therefore only the vertices on this particular region are considered in the error computation. Furthermore those regions (*i.e. ears, neck and forehead*) do not have any constrain during the fitting and can be discarded from the reconstruction quality estimation. When the surface reconstruction is based on the full-perspective model the rigid transform error is computed as well.

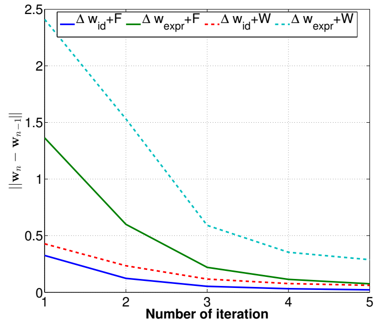
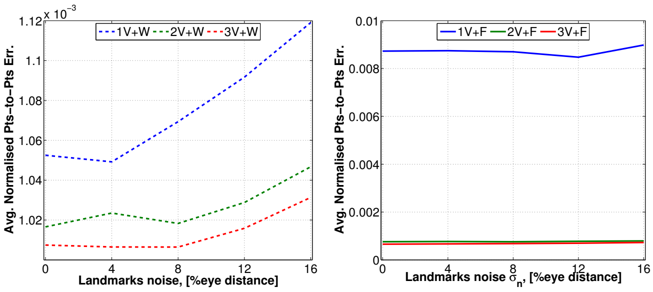


Fig. 3: Evolution of the parameters update over the iteration for weak-perspective (W) and full-perspective (F) projection model

B. Shape Estimation

The experiments have been performed for static image, given a specific identity/expression pairs the 3D surface is reconstructed with one of the configuration explained previously and quality estimators are computed. Moreover the reconstruction pipeline does not include any kind of tracking between each surface reconstruction.

Fig. 4a shows the average normalised point-to-point error for the reconstruction using the weak-perspective projection and it can be seen that adding extra views improves the overall quality of the reconstruction. What appends is that more control points are part of the optimisation which adds more constrains to the surface therefore leads to a smaller average error per vertex. For instance in case of maximum noise it goes from almost 1.12×10^{-3} with one view down to 1.03×10^{-3}. It also shows that the speed at which the relative error increases is slower when using multiple view based reconstruction which means that the system is more robust against noise. Comparing the same reconstruction using the full-perspective projection model instead (Fig. 4b), the first thing to notice is that for a single view the result is not improved as it would be expected, it is even worst. The reason is that the rigid-transformation is not properly estimated even when no noise is added due to poor initialisation and confusion between object's position and scale, more details will be provided in section III-E. However for multiple view the average normalised point-to-point error is consistent and adding extra views decreases it down to 0.7×10^{-3}. However this result can not be directly compared to the weak-perspective reconstruction since not both methods include the rigid-transformation in the reconstruction. Comparing the normal of reconstructed surface and the original one provides information about the surface independently of the vertices location therefore it can be used to tell if there is an improvement between both technics. Fig. 5 shows the normal's difference between each reconstruction methods where the dashed lines stand for the weak-perspective model and the solid one for the full-perspective model. In average the normal's difference when



(a) Weak-perspective projection model

(b) Full-perspective projection model

Fig. 4: Average point-to-point normalised error

using full-perspective model with multiple views is smaller indicating that the topology of the reconstructed surface is closer to the original, therefore its quality is better. Again for single input, it highlights that full-perspective model is not suited for the reconstruction.

C. Model Weights Estimation

The last quality estimator investigated is the error made on the face model parameters itself. This comparison is possible since the ground truth is available form the training set, therefore it can be compared with the coefficients estimated by the reconstruction pipeline.

Fig. 6 confirms again that full-perspective projection model is not well suited for single-view fitting. The left graph shows the error on the identity parameters, and using full-perspective model do not bring a significant improvement. However when looking at the expression coefficients, using such model reduces the error on the coefficient leading to a better reconstruction. Since the projection model is closer to the physical phoneme it is able to better catch small changes related to the expression.

D. Error Distribution

The first line of Fig. 7, from the left to the right, shows the face ground truth (Fig.7a) where each red dot indicates the position of the landmarks used during the synthesis, the normalised point-to-point error (Fig.7b) and the normal difference (Fig.7c) for the weak-perspective projection model. The second line shows the same errors in the same order for the full-perspective projection (Fig.7d,7e). The simulated input is for three cameras where reasonable amount of noise is added (*i.e.* $\sigma_n = 8\%$). Reconstruction with weak-perspective considers projection information from each view independently therefore the alignment error will be stacked in the reconstruction minimization. Moving to calibrated camera allows to consider the rigid transformation estimation in a global system and it provides a much better alignment. Including full-perspective projection model in the reconstruction pipeline as well also lead to a more accurate reconstruction (*i.e. expression more marked*).

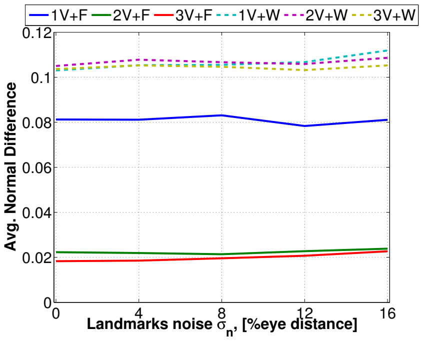
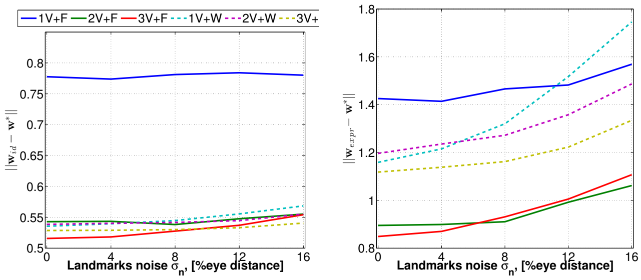


Fig. 5: Average eerror in surface normal for weak-perspective (W) and full-perspective (F) projection model



(a) Identity coefficients error (b) Expression coefficients error

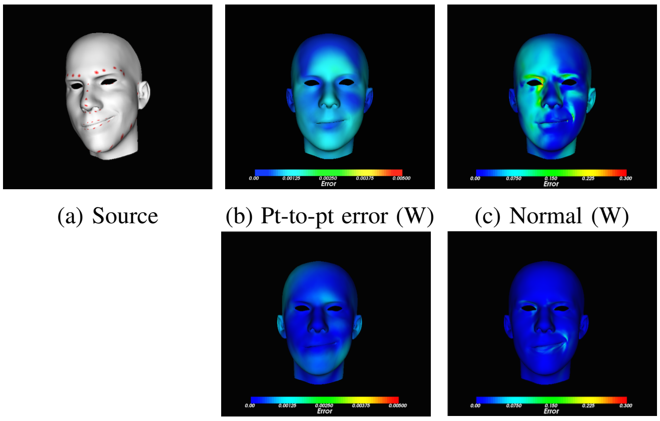
Fig. 6: Error on face model coefficient for weak-perspective (W) and full-perspective (F) projection model

TABLE I: Rigid transformation average error and standard deviation with no noise on landmarks, $\sigma_n = 0\%$ and significant amount of noise $\sigma_n = 12\%$

	$\sigma_n = 0\%$			$\sigma_n = 12\%$		
	1 View $\mu \pm \sigma$	2 Views $\mu \pm \sigma$	3 Views $\mu \pm \sigma$	1 View $\mu \pm \sigma$	2 View $\mu \pm \sigma$	3 View $\mu \pm \sigma$
s	11.4 ± 7.9	5.0 ± 4.0	4.7 ± 3.6	10.9 ± 7.9	5.2 ± 4.4	5.1 ± 4.4
γ	4.4 ± 12.4	1.3 ± 1.4	1.0 ± 1.2	4.8 ± 13.4	1.3 ± 1.4	1.1 ± 1.8
θ	10.3 ± 18.5	4.9 ± 3.9	3.7 ± 2.9	10.0 ± 18.2	4.7 ± 3.9	3.7 ± 3.1
ϕ	7.9 ± 15.4	2.5 ± 3.3	1.9 ± 2.3	7.0 ± 13.6	2.4 ± 3.0	1.9 ± 2.5
t_x	13.3 ± 20.7	2.5 ± 3.0	1.9 ± 2.1	12.7 ± 20.7	2.4 ± 3.0	2.1 ± 2.6
t_y	13.6 ± 20.3	4.6 ± 3.9	3.8 ± 3.0	12.9 ± 20.3	4.8 ± 3.9	3.9 ± 3.1
t_z	50.3 ± 31.9	4.0 ± 3.6	3.6 ± 3.3	49.3 ± 31.3	4.0 ± 3.6	3.8 ± 3.3

E. Rigid Transformation Estimation

When the reconstruction pipeline work with full-perspective projection model during the face model fitting, the estimation of the absolute pose of the 3D object is part of the process. Working with synthetic data provides the ground truth of the transformation and it is possible to define how precise is the estimation. Table I shows the average error and the standard deviation across all views with no noise and significant amount of noise (*i.e.* $\sigma_n = 0\%$ and $\sigma_n = 12\%$ *of the eye distance*). The amount of noise added to the facial landmarks (*i.e. up to* 16% *of the eye distance*) does effect the non-rigid reconstruction but does not have a major impact on the rigid transformation approximation.



(a) Source (b) Pt-to-pt error (W) (c) Normal (W)

(d) Pt-to-pt error (F) (e) Normal (F)

Fig. 7: Error distribution for weak-perspective and full perspective projection model

The values given for the angles are in degrees and the distances are in millimetres. The s value is the error for the scaling factor, $t_{x,y,z}$ are the position error and the γ, θ, ϕ are the error for the roll, pitch and yaw angles. It clearly shows that the single camera model is not well suited to estimate properly the absolute pose and scale of the object. This makes perfect sense since there is a confusion between the scale and the position variation. A change of size or position will have the same impact from the viewpoint of the camera. However as soon as there is a second viewpoint involved in the estimation, the confusion disappears and it becomes possible to properly defined the pose and scale of the object.

IV. CONCLUSIONS

In this work we investigated the reconstruction of 3D face using 2D facial landmarks and Bilinear 3D Morphable Model in a multi-view setup. The modes included in the model are the identity and the expression of the individual. A reconstruction pipeline for fitting Bilinear 3D Face Model in multi-camera images is proposed, in which coordinate-descent optimisation and weak-perspective projection model is applied. It has also be shown how the pipeline can be extended to use the full-perspective projection model in a multi-view setup. Principle for dynamic features selection have been mentioned to select only appropriated landmarks to be include in the reconstruction optmisation. We conduct quantative evaluation on synthetic data for assessing the proposed reconstruction approach. Results show that Bilinear Face Model fitted with multiple views/inputs improves the quality of the reconstructed surface. Moreover using a full-perspective projection model during the fitting process also helped to improve the reconstruction accuracy especially for the expression.

REFERENCES

[1] Cao, Chen and Weng, Yanlin and Zhou, Shun and Tong, Yiying and Zhou, Kun, FaceWarehouse: A 3D Facial Expression Database for Visual Computing, *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, 2014, pp 413–425.

[2] Blanz, Volker and Vetter, Thomas,A Morphable Model for the Synthesis of 3D Faces, *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, 1999, pp 187–194.

[3] Timothy F. Cootes and Gareth J. Edwards and Christopher J. Taylor, "Active Appearance Models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998, pp 484–498.

[4] Vlasic, Daniel and Brand, Matthew and Pfister, Hanspeter and Popović, Jovan, Face Transfer with Multilinear Models, *ACM Trans. Graph.*, vol. 24, 2005, pp426–433.

[5] Blanz, Volker and Vetter, Thomas, "Face Recognition Based on Fitting a 3D Morphable Model", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol 25, 2003, pp 1063–1074.

[6] Ramanathan, Subramanian and Kassim, Ashraf A. and Venkatesh, Y. V. and Wah, Wu Sin, "Human Facial Expression Recognition using a 3D Morphable Model", *ICIP*, 2006, pp 661–664.

[7] Sami Romdhani,Jean-Sebastien Pierrard and Thomas Vetter, "3D Morphable Face Model, a Unified Approach for Analysis and Synthesis of Images".

[8] Sami Romdhani and Thomas 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.

[9] L. De Lathauwer, "Signal Processing based on Multilinear Algebra", 1997, Katholieke Universiteit Leuven, Belgium.

[10] Vmmlib, A vector and matrix math library, http:/vmmlib.sf.net

[11] V. Blanz, A. Mehl, T. Vetter, and H. p. Seidel,"A Statistical Method for Robust 3D Surface Reconstruction From Sparse Data", *Int. Symp. on 3D Data Processing, Visualization and Transmission*, 2004, pp 293–300.

[12] Nathan Faggian, Andrew P. Paplinski, and Jamie Sherrah, "3d Morphable Model Fitting From Multiple Views". *FG*, 2008, pp 1-6.

[13] Alvaro Collet Romea and Siddhartha Srinivasa, "Efficient Multi-view Object Recognition and Full Pose Estimation", *IEEE International Conference on Robotics and Automation* , 2010.

[14] Alvaro Collet Romea, Dimitry Berenson, Siddhartha Srinivasa, and David Ferguson, "Object Recognition and Full Pose Registration from a Single Image for Robotic Manipulation", *EEE International Conference on Robotics and Automation*, 2009.

[15] H. Li, T. Weise, and M. Pauly, "Example-based facial rigging", *ACM Trans. Graph.*, vol. 29, pp. 32:132:6, 2010

[16] Xuehan Xiong and Fernando De la Torre Frade, "Supervised Descent Method and its Applications to Face Alignment," *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013