

# Parametric T-spline Face Morphable Model for Detailed Fitting in Shape Subspace

Weilong Peng<sup>1</sup>, Zhiyong Feng<sup>2</sup>, Chao Xu<sup>2</sup>, Yong Su<sup>1</sup>

<sup>1</sup>School of Computer Science and Technology, <sup>2</sup>School of Computer Software  
 Tianjin University, Tianjin, China

{wlpeng, zyfeng✉, xuchao, suyong}@tju.edu.cn

## Abstract

*Pre-learnt subspace methods, e.g., 3DMMs, are significant exploration for the synthesis of 3D faces by assuming that faces are in a linear class. However, the human face is in a nonlinear manifold, and a new test are always not in the pre-learnt subspace accurately because of the disparity brought by ethnicity, age, gender, etc. In the paper, we propose a parametric T-spline morphable model (T-splineMM) for 3D face representation, which has great advantages of fitting data from an unknown source accurately. In the model, we describe a face by  $C^2$  T-spline surface, and divide the face surface into several shape units (SUs), according to facial action coding system (FACS), on T-mesh instead of on the surface directly. A fitting algorithm is proposed to optimize coefficients of T-spline control point components along pre-learnt identity and expression subspaces, as well as to optimize the details in refinement progress. As any pre-learnt subspace is not complete to handle the variety and details of faces and expressions, it covers a limited span of morphing. SUs division and detail refinement make the model fitting the facial muscle deformation in a larger span of morphing subspace. We conduct experiments on face scan data, kinect data as well as the space-time data to test the performance of detail fitting, robustness to missing data and noise, and to demonstrate the effectiveness of our model. Convincing results are illustrated to demonstrate the effectiveness of our model compared with the popular methods.*

## 1. Introduction

3D Morphable Models (3DMMs), being originally formulated by Blanz and Vetter [4], are powerful 3D statistical models of human face shape and texture, and now have been widely applied in numerous areas, such as computer vision, human behavioral analysis, computer graphics [5, 2, 1]. By assuming that human faces are in a linear class, 3DMMs are

generally constructed by performing typical Principal Component Analysis (PCA) on a training set of facial meshes and textures, and any new face shape as well as its texture can be represented by the linear combination of components (or basis).

Generally, 3DMMs is used as a tool to encode any 3D face in a low dimensional feature space, a compact representation that makes tractable many 3D facial analysis problems such as recognition and expression normalization [5, 11, 29]. And 3DMMs owning rich shape priors can be leveraged in fitting algorithms to reconstruct good 3D representations of faces from data-deficient sources like in-the-wild 2D landmarks [24] or reconstruct noisy 3D depth scan data [7] accurately.

In fact, not only identity fitting but also expression fitting is a very important performance index of the model. Recently, Booth et al build 3DMM from 10,000 high-quality neutral 3D facial scans considering different ages, genders, and ethnic backgrounds [7], and report the stat-of-art fitting results. Cao et al propose an expressions model containing 19 expressions of 150 individuals [10], which can fit the image expression and perform good-looking animation. One problem is that pre-learnt subspace is not complete due to the variety and uncertain details of faces and expressions, resulting in a limited performance of fitting details. Therefore, a defective fitting result always happens when a new test is not exactly in the pre-learnt face subspace. Another problem is that fitting result mixes identity information and expression information when the identity is unknown.

Motivated by the linear subspace idea of 3DMMs, as well as amendable limitations, in the paper we propose a parametric T-spline morphable model (T-splineMM) based on a pre-learnt face subspace for 3D face representation. To enlarge the representation span, shape units (SUs) division is done on a T-mesh according to the facial action coding system (FACS) [12]. Being compared with segment dividing proposed by Blanz et al [4] on the surface directly, the SUs dividing is based on parametric space of T-splines shape, and contribute to continuous transitions between ad-

jacent subregions. To handle incorrect fitting, we also propose a T-splineMM fitting algorithm for 3D data which can handle the missing data, approach the details, and separate the identity part and expression part. To illustrate the strength of our approach, we perform experiments on face scan data, kinect data as well as space-time 3D expressions. Both qualitative and quantitative experiments are conducted and compared with the state-of-the-art method.

In summary, this paper has made three contributions.

- A parametric T-spline surface morphable model (T-splineMM) is proposed based on SUs division on T-mesh and pre-learnt face subspaces of both identity and expression. And it can implement a larger span of morphing beyond the prior statistical data.
- T-splineMM fitting algorithms are proposed to not only fits 3D data robustly to missing data, noise, ethnicity and expression, etc., but also separate the identity part from expression in detail.

## 2. Related Work

**Morphing technology** Face morphing technologies, including 2D image morphing [16] and 3D morphing [27], have been used frequently in computer animation, movie and facial image processing. While 2D image morphing techniques are unable to correctly handle changes in illumination and visibility, 3D morphing technologies create shape that is independent of the viewing and lighting parameters. 3D shape morphing techniques can be divided into two categories: volume-based methods and surface-based method. Volume-based methods like level set approaches [27, 13] implement nonlinear and "smooth" interpolation produce between an object and a target subject, and the target always can be fitted accurately. But they are time- and memory-consuming. Surface-based methods are typically faster to compute and require less memory because they operate on a lower-dimensional, but they are influenced by the correspondence problem and the path problem. 3DMMs solve the correspondence registration by considering the similarity of face, and implement face morphing based on pre-learnt face subspace. But the morphing span is limited by prior information.

**3D Morphable models** To our knowledge, public and available 3DMMs of human face include Basel Face Model (BFM) from University of Basel [19], an expression model from FaceWarehouse by Cao et al [10], and a fused 3DMM integrating 3D face shapes of different subjects in neutral expression [9] and different expressions [8, 6]. The reconstruction of 3DMM usually consists of two main steps: establishing group-wise dense correspondence between a training set of facial meshes, and then performing some kind of statistical analysis such as PCA on the registered data to produce a low-dimensional model. To a great extent, fitting performance of 3DMMs relies on the completeness

of the registered training data. And the more training data the better. Blanz and Vetter built model for the facial meshes of 200 subjects of similar ethnicity and age [4]. James Booth et al did that for 10,000 high-quality 3D facial scans considering different ages, genders, and ethnic backgrounds [7]. Fitting facial expression is also very important in 3DMM application, e.g. [29]. Chu et al extend 3DMM to contain expressions as the offset [11], but only consider 6 universal expressions including angry, happy, fear, disgust, sad and surprise. And FaceWarehouse presented by Cao et al [10] contains 19 expressions for learning. In fact, a high-quality 3D facial scan has about 20,000-50,000 points. Although only 80 components contain 96% of the variance of about 10,000 neutral faces in [7], some details must be lost for 3D fitting. In another aspect, the category number of expression is not limited to 19 because of expression variety. That is the reason why a new test, sometimes, cannot be represented by 3DMMs accurately.

**Surface representation.** Surface types involved by surface deforming technique include discrete type like polygonal mesh, and continuous type like explicit and implicit surface model [18]. Mesh-based methods are typically faster to compute when they operate on a lower dimensional representation of an object. Parametric surface model features its smoothness and continuity during morphing change, such as B-spline evolution [14], B-spline modeling [21] and T-spline level set [27, 13]. Spline surfaces require fewer number of control points to describe a complete surface, in contrast to discrete surface points. Generally, T-splines surfaces use fewer parameters than B-splines and 3D mesh to describe a same shape by considering the different complexity of local shape [25, 22]. Although the mesh-based representation can also be smoothed by an operation like Laplacian Editing [26][23], the precision of mesh is controlled by a template. Therefore, with a high compression ratio, low storage and good continuity, parametric T-spline surface will be a useful tool to model face morphing accurately if we combine the control point adjustment with the idea of representation in linear face class.

## 3. T-spline face morphable model

In this section, firstly we define the parametric T-spline surface of face, and describe SUs division on T-mesh in section 3.1. Secondly, T-splineMM is formulated based on pre-learnt face subspaces in section 3.2. Finally, a 3D fitting method is given by using our morphable model in section 3.3.

### 3.1. Definition of T-spline face

T-splines, also being called Point based Splines (PB-splines), are generalizations of tensor product B-splines [25], for which some order has been imposed on the control points by means of control grid (or T-mesh). The

equation for a T-spline surface is

$$\mathbf{S}(s, t) = \frac{\sum_{i=1}^n \mathbf{P}_i B_i(s, t)}{\sum_{i=1}^n B_i(s, t)}, \quad (s, t) \in Dom$$

where  $\mathbf{P}_i$  are control points, and  $Dom$  is the domain of  $(s, t)$ .  $B_i(s, t)$  are basis functions given by

$$B_i(s, t) = N_{i0}^3(s)N_{i0}^3(t)$$

where  $N_{i0}^3(s)$  and  $N_{i0}^3(t)$  are certain cubic B-splines, whose associated knot vectors  $s_i = [s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}]$  and  $t_i = [t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}]$  are determined by the T-mesh, as seen in Fig.1 (d).

In order to simplify the notion, we use  $\mathbf{u}$  to represent the point  $\mathbf{u} = (s, t)$  in parameter domain, and gather the control points (in a suitable ordering) in a column vector  $\mathbf{P}$ . The T-spline basis functions form another column vector  $\mathbf{b} = [b_1, b_2, \dots, b_n]^T$ , where

$$b_i(\mathbf{u}) = B_i(\mathbf{u}) / \sum_{i=1}^n B_i(\mathbf{u}), \quad i = 1, 2, \dots, n.$$

T-spline equation is redefined as

$$\mathbf{S}(\mathbf{u}; \mathbf{P}) = \mathbf{b}(\mathbf{u})^T \mathbf{P}, \quad \mathbf{u} \in Dom. \quad (1)$$

Because human faces are similar in topology and geometry, T-mesh can be fixed (thus  $\mathbf{b}$  is fixed), and then the surface morphing is determined by only changing control points  $\mathbf{P}$ .

Since a T-spline function is piecewise rational, the T-spline parametric surfaces are piecewise algebraic surfaces. Moreover, if no singular points are present, they inherit the order of differentiability of the basis functions, i.e., they are  $C^2$  in the cubic case.

### 3.1.1 T-mesh construction

T-mesh imposed on PB-spline is defined in the parameter domain, and serves two purposes. First, it reflects the topology of PB-spline control points in parameter space, and provides a friendly user interface of adjusting shape. Second, it deduces the knot vectors  $s_i$  and  $t_i$  for each basis function.

Firstly, surface parameterization is necessary in order to map 3D surface points onto complex (or 2D) plane. According to Riemann mapping theorem, we use conformal map to find a parameterization, or an efficient 3D-to-2D mapping, in terms of both angular and edge distortion. Every surface point  $(x, y, z)$  is mapped to a complex points  $s + i \cdot t$  of the complex plane. Secondly, space partition is carried on complex plane by using quad-tree division, which restricts the point numbers in all region is equal or lesser than a certain threshold. The edges of all rectangles form the T-mesh, and the junction point of edges are called the knots.

By collecting all  $n$  knot in the T-mesh and abstracting the real and imaginary part, we get  $n$  parameter points

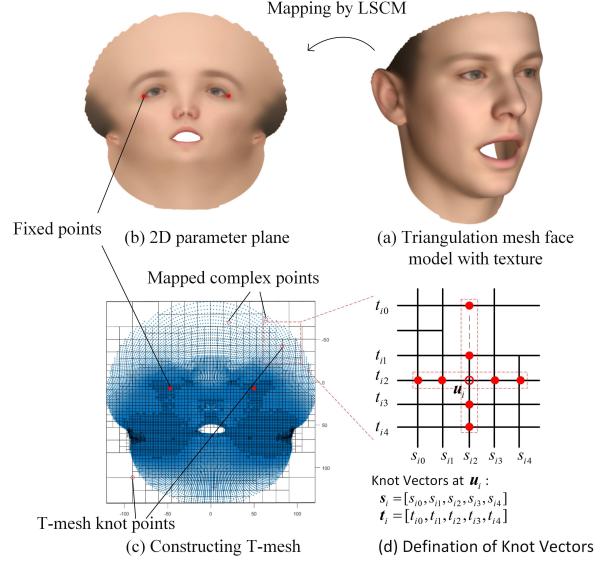


Figure 1. Diagrams of conformal mapping and T-mesh construction: triangulation mesh model is mapping to 2D parameter plane by LSCM with two fixed points at outer corners of eyes; T-mesh is constructed by quad-tree division based on the mapped complex points and T-mesh parameter points are obtained at the knots; knot vectors  $s_i = [s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}]$  and  $t_i = [t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}]$  for each knot  $\mathbf{u}_i$  are deduced from the T-mesh.

$\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$ ,  $\mathbf{u}_i = (s_i, t_i)$  that corresponds to  $n$  control points  $\{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n\}$ . And according to T-mesh structure we also compute the knot vectors  $s_i = [s_{i0}, s_{i1}, \dots, s_{i4}]$ ,  $t_i = [t_{i0}, t_{i1}, \dots, t_{i4}]$ , and T-spline basis coefficients  $\mathbf{b}(\mathbf{u}_i)$ .

T-mesh describes local characteristics of face geometry well, because local positions with denser mesh show more shape details and can be adjusted by control points more precisely. Particularly, we use Cauchy-Riemann equations based least square conformal mapping (LSCM) [15], to obtain a quasi-conformal parameterization, which is also widely used in applications of texture mapping, model morphing, etc. Particularly, the outer corners of two eyes are as the fixed points in LSCM, as is shown in Fig.1 (b) and (c). And T-mesh should better be constructed from a mouth-open face, to obtain dense grid at mouth contour which can implement finer-grained control over mouth action, as is shown in Fig.1 (c).

### 3.1.2 Shape unit division on T-mesh

FACS [12] defines 32 Action Units (AUs) to describe the basic actions of human face, which are a contraction or relaxation of one or more muscles. Different facial expressions can be created based on different combinations of AUs. For clarification, FACS is an index of facial expressions, but does not actually provide any bio-mechanical in-

Table 1. Shape unit division based on facial muscles and action units

Shape Unit	Muscle NO.	Muscle Name	Descriptions of Relative Actions
A	(1) (2)	Frontalis, pars medialis Frontalis, pars lateralis	Inner Brow Raiser; Outer Brow Raiser
B1,B2	(4) (5) Other	Corrugator supercilii Orbicularis oculi: (a)pars palpebralis and (b)pars orbitalis Levator palpebrae superioris	Brow Lowerer; Upper Lid Raiser; Cheek Raiser; Lid Tightener; Slit; Eyes Closed; Squint; Wink; Lid droop
C	(6) (7)	Depressor supercilii Nasalis	Brow Lowerer; Nose Action
D1,D2	(8) (9) (10) (11) (12) (14) Other	Levator labii superioris alaqueae nasi Zygomaticus: minor and major Risorius Platysma Buccinator Masseter Levator anguli oris	Nose Wrinkler; Upper Lip Raiser; Nasolabial Deepener; Lip stretcher; Dimpler; Lip Corner Puller; Cheek Puffer
E	(13)	Orbicularis oris	Lip Puckerer; Lip Funneler; Lip Tightener; Lip Pressor; Lips part; Lip Suck
F	(15) (16) (17) Others	Depressor labii inferioris Mentalis Depressor anguli oris Temporalis; Internal Pterygoid; Pterygoids, Digastric	Lower Lip Depressor; Chin Raiser; Lip Corner Depressor; Jaw Drop
G	(3) Others	Auricularis Other muscles that control contour	Ear Action; Face contour action.

formation about the degree of muscle activation. Though muscle activation is not given, the main muscles involved in FACS can be used to define shape units (SUs) on the T-mesh.

T-mesh corresponds to the control points that determine the shape of face surface. Then T-mesh division makes it possible that the control points simulate different muscle functions at different locations via different SUs. A local shape is always affected by several neighbour muscles, so we divide 9 areas  $\{A, B1, B2, C, D1, D2, E, F, G\}$  on T-mesh for SU division according to facial map of 17 muscle units as is shown in Fig.2. See Table.1 for detailed muscle names and relative action descriptions covered by SUs. E.g., SU A is controlled by frontalis, pars medialis and lateralis, and can produce actions of inner or outer brow raiser.

Finally, T-spline control points are divided into 9 groups, that is  $\mathbf{P}^j, j = 1, 2, \dots, 9$ , and the relations between them are

$$\mathbf{P} = \sum_{j=1}^9 \mathbf{P}^j, \quad j = 1, 2, \dots, 9.$$

which also means that  $\mathbf{P}^j$  is non-zero at the location of  $j$ -th group of control points while zero at other locations. We can change  $SU_j$  by changing corresponding control points  $\mathbf{P}^j$  to obtain new shape by

$$\mathbf{P}_{new} = \mathbf{P} + \Delta\mathbf{P}^j$$

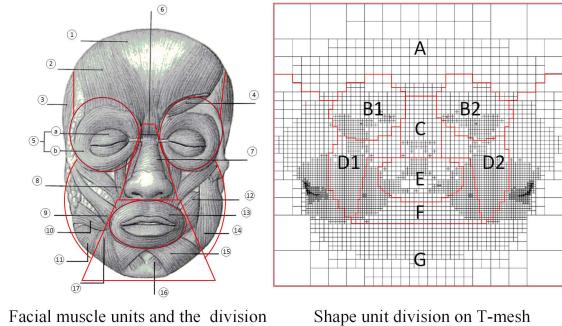


Figure 2. Shape unit division on T-mesh according to Facial muscle units: 9 shape units  $\{A, B1, B2, C, D1, D2, E, F, G\}$  on T-mesh are defined by considering the 17 muscle units and their neighborhood relations that are relative to facial actions; reference Table.1 for detailed definition of SUs, the names of covered muscles and relative actions.

where  $\Delta\mathbf{P}^j$  is the control increment for  $SU_j$ . Different local shapes can be controlled based on different SUs: SU A covers the frontalis area; SU B1 and B2 cover the area of eyes and brows; SU C covers the nose area; SU D1 and D2 cover the cheek area; SU E covers the area of mouth; SU F covers chin area around mouth; SU G covers area including ears, chin and outer cheeks.

### 3.2. T-splines morphable model

By referring to the 3DMM [4] assumption that a face shape can be linear combination of shape bases, we build the T-spline morphable model:

$$\mathbf{S}(\mathbf{u}) = \mathbf{b}(\mathbf{u})(\bar{\mathbf{P}} + \sum_{i=1}^{N_{id}} \alpha_i \ddot{\mathbf{P}}_{id}(i) + \sum_{i=1}^{N_{ex}} \beta_i \ddot{\mathbf{P}}_{ex}(i)), \mathbf{u} \in Dom. \quad (2)$$

where  $\bar{\mathbf{P}}$  is the *mean* control points,  $\ddot{\mathbf{P}}_{id}(i), i = 1, 2, \dots, N_{id}$  are identity *bases* for control points, and  $\ddot{\mathbf{P}}_{ex}(i), i = 1, 2, \dots, N_{ex}$  are expression *bases* for control points. Then  $\mathbf{b}(\mathbf{u})\bar{\mathbf{P}}$ ,  $\mathbf{b}(\mathbf{u})\ddot{\mathbf{P}}_{id}(i)$  and  $\mathbf{b}(\mathbf{u})\ddot{\mathbf{P}}_{ex}(i)$  are mean shape, identity basis, and expression basis of face. Coefficients  $\alpha_i$  and  $\beta_i$  are used to control corresponding shape components.

However, a pre-learnt facial shape subspace cannot bring a accurate representation because of face variety of human identities and expressions, so that it covers a limited span of morphing. Therefore, based on the SUs predefined according to T-mesh, we can simulate face deformation in larger span of morphing by:

$$\begin{aligned} \mathbf{S}(\mathbf{u}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\xi}_{id}, \boldsymbol{\xi}_{ex}) &= \mathbf{b}(\mathbf{u})(\bar{\mathbf{P}} + \sum_{i=1}^{N_{id}} \sum_{j=1}^9 \alpha_i^j \ddot{\mathbf{P}}_{id}^j(i) \\ &\quad + \boldsymbol{\xi}_{id} + \sum_{i=1}^{N_{ex}} \sum_{j=1}^9 \beta_i^j \ddot{\mathbf{P}}_{ex}^j(i) + \boldsymbol{\xi}_{ex}). \end{aligned} \quad (3)$$

where  $\ddot{\mathbf{P}}_{id}^j(i), i = 1, 2, \dots, N_{id}$  and  $\ddot{\mathbf{P}}_{ex}^j(i), i = 1, 2, \dots, N_{ex}$  are respectively identity and expression *basis* control points for SU<sub>j</sub>, and coefficients  $\{\alpha_i^j\}$  and  $\{\beta_i^j\}$  are stacked in  $\boldsymbol{\alpha}$  and  $\boldsymbol{\beta}$  respectively. Particularly,  $\boldsymbol{\xi}_{id}$  and  $\boldsymbol{\xi}_{ex}$  are identity and expression details that face subspace cannot supply.

The new surface morphable model by Equ.3, which is called T-splineMM, has  $9 * (N_{id} + N_{ex})$  degrees of freedom for shape, while the model by Equ.2 has only  $N_{id} + N_{ex}$ . In fact, T-splineMM considers the local morphing in the domain of T-spline control points, and then the span of morphing space is largely broadened in continuous shape space. And the details parameters  $\boldsymbol{\xi}_{id}$  and  $\boldsymbol{\xi}_{ex}$  beyond the representation of face subspace, and contribute to detailed 3D fitting.

#### Building model

Because T-spline morphing is based on face shape subspace, any pre-learnt 3DMM model can be utilized to build our model. To implement a morphing with enough large of span, we build T-splineMM based on fusing the identity bases from BFM model [19] and the expressional bases from FaceWarehouse [10]. The former are trained on neutral face scans of 100 female and 100 male persons between 8 and 62 years old. And the latter are trained on face scans of 150 persons each with 20 expressions. Firstly, we use Nonrigid ICP(Iterative Closest Point) [3] to find corresponding points and merge the two models. The merged 3DMM is given by  $S = \bar{S} + B_{id}\alpha_{id} + B_{ex}\alpha_{ex}$ . Secondly, we reversely calculate the *mean* control points and *basis* control points  $\{\bar{\mathbf{P}}, \ddot{\mathbf{P}}_{id}(i), \ddot{\mathbf{P}}_{ex}(i)\}$  from T-splineMM from  $\bar{S}, B_{id}, B_{ex}$ . Particularly, facial T-mesh is constructed based on an mouth-open face simulated by BFM.

### 3.3. 3D fitting

First, we define the T-spline distance between source S and target T within domain D as

$$\int_D (\mathbf{S}(\mathbf{u}) - \mathbf{T}(\mathbf{u}))^2 dD,$$

where  $v^2$  here stands for the traditional  $L_2$ -norm of a  $3 \times 1$  vector  $v$ . Then surface fitting is to compute the control points  $\mathbf{P}$  of surface S by minimizing the T-spline surface distance. For the practical task (e.g., point cloud) a discrete version is more appropriate, so we minimize

$$E_f(\mathbf{P}) = \frac{A(D)}{N_0} \sum_{j=1}^{N_0} w(\mathbf{u}_j)(\mathbf{S}(\mathbf{u}_j) - \mathbf{T}(\mathbf{u}_j))^2, \quad (4)$$

where  $w(\mathbf{u}_j) = \begin{cases} 1, & \text{if } T(\mathbf{u}_j) \text{ exists,} \\ 0, & \text{if } T(\mathbf{u}_j) \text{ is absent.} \end{cases}$ , and  $\mathbf{u}_{j=1:N_0}$  is a sequence of sampling parameter points, which are uniformly distributed in the T-spline function domain D.  $A(D)$  is the area of the domain. Because  $w(\mathbf{u}_j)$  indicates the correspondence between source and target, the minimization problem becomes a step-by-step least-square approximation by updating  $w(\mathbf{u}_j), j = 1, \dots, N_0$ . Particularly, a s-moothness term  $E_s$  is necessary to handle the missing data, e.g.,

$$E_s(\mathbf{P}) = \sum_{j=1}^{N_0} (\Delta \mathbf{S}(\mathbf{u}_j) - \Delta \bar{\mathbf{S}}_1(\mathbf{u}_j))^2 + \zeta \cdot \Delta^2 \mathbf{S}(\mathbf{u}_j), \quad (5)$$

in which  $\Delta$  is Laplace operator that makes  $\Delta \mathbf{S} = \frac{\partial^2 \mathbf{S}}{\partial u^2} + \frac{\partial^2 \mathbf{S}}{\partial v^2}$ , and  $\bar{\mathbf{S}}_1$  is a reference neutral face.  $\Delta$  and  $\Delta^2$  makes a smooth surface and continuous evolution during optimization. Combining Equation(4) Equation(5), we can get wonderful T-spline fitting by step-by-step minimizing

$$E_T(\mathbf{P}) = \lambda_1 \cdot E_f(\mathbf{P}) + \lambda_2 \cdot E_s(\mathbf{P}). \quad (6)$$

#### 3.3.1 Moving control points along shape subspace of SUs

When applying our T-spline morphable model, if we set  $\boldsymbol{\xi}_{id}$  and  $\boldsymbol{\xi}_{ex}$  to  $\mathbf{0}$ , and only compute the coefficients  $\boldsymbol{\alpha}$  and  $\boldsymbol{\beta}$  in Equation (3) instead of control points  $\mathbf{P}$  of Equation (1). Then the fitting is to solve the following minimization

$$E_M(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \lambda_1 \cdot E_f(\boldsymbol{\alpha}, \boldsymbol{\beta}; \boldsymbol{\pi}) + \lambda_2 \cdot E_s(\boldsymbol{\alpha}, \boldsymbol{\beta}), \quad (7)$$

where

$$\begin{cases} E_f(\boldsymbol{\alpha}, \boldsymbol{\beta}; \boldsymbol{\pi}) = \sum_j w(\mathbf{u}_j)(\mathbf{S}(\mathbf{u}_j; \boldsymbol{\alpha}, \boldsymbol{\beta}) - \boldsymbol{\pi} \circ \mathbf{T}(\mathbf{u}_j))^2, \\ E_s(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_j (\Delta \mathbf{S}(\mathbf{u}_j; \boldsymbol{\alpha}, \boldsymbol{\beta}) - \Delta \bar{\mathbf{S}}_1(\mathbf{u}_j))^2 + \zeta \cdot \Delta^2 \mathbf{S}(\mathbf{u}_j; \boldsymbol{\alpha}, \boldsymbol{\beta}), \end{cases}$$

in which model  $\mathbf{S}(\mathbf{u}_j; \alpha, \beta)$  considers parameters  $\{\alpha, \beta\}$  only and ignores  $\{\xi_{id}, \xi_{ex}\}$ . And operator  $\pi$  considers the scale, rotation and transformation between target and source. The optimization procedure goes as following Algorithm 1.

### Algorithm 1

1. Initialize  $\bar{\mathbf{S}}_1$  with  $\mathbf{S}$ ;
2. Estimate  $\pi$  and compute  $\{w(\mathbf{u}_j)\}$  and  $\{\mathbf{T}(\mathbf{u}_j)\}$ , according to  $\bar{\mathbf{S}}_1$  and  $\mathbf{T}$ ;
3. Get  $\{\alpha, \beta\}$  by solving Equ(7);
4. Update  $\bar{\mathbf{S}}_1$  in  $E_s$  with  $\mathbf{S}(\mathbf{u}; \alpha, \beta)$ ;
5. Termination: check whether the stopping criterion is satisfied for  $\{\alpha, \beta\}$ . Continue with step 2 (no) or finish (yes).

During optimization the T-spline control points move along the pre-learnt shape subspace of SUs. It is a little bit like classic 3DMM fitting, but it is more accuracy than 3D-MM. However, it cannot approach to a detailed fitting by Algorithm 1, because a new test is not always exactly in the pre-learnt face subspace even we have enlarged the morphing span via multiple SUs. In another aspect, it cannot guarantee an authentic continuity at the adjacent parts of SUs because shape bases between SUs are almost independent from each other during fitting. Therefore, to approach the details unfulfilled is another important work.

### 3.3.2 Final Refinement

The final refinement is to compute parameters  $\xi_{id}$  and  $\xi_{ex}$  of T-spline morphable model, but a big issue is that identity detail  $\xi_{id}$  and expression detail  $\xi_{ex}$  are mixed with each other. We find that identity shape must be in neutral shape space where face is neutrally mouth-closed and eyes-opened. That is to say, mainly neutral face have similar 2nd-order information at contours of mouth and eyes. Therefore, to separate the identity details, an additional 2nd- order term  $E_c(\xi_{id}; \alpha)$  is considered in the optimization:

$$E_R(\xi_{id}, \xi_{ex}) = \lambda_1 \cdot E_f(\xi_{id}, \xi_{ex}; \alpha, \beta, \pi) + \lambda_2 \cdot E_s(\xi_{id}, \xi_{ex}; \alpha, \beta) + \lambda_3 \cdot E_s(\xi_{id}; \alpha) + \lambda_4 \cdot E_c(\xi_{id}; \alpha), \quad (8)$$

where

$$E_c(\xi_{id}, \alpha) = \sum_{\mathbf{u}_j \in C} (\Delta \mathbf{C}(\mathbf{u}_j; \xi_{id}, \alpha) - \Delta \bar{\mathbf{C}}(\mathbf{u}_j; \xi_{id}, \alpha))^2$$

and  $\Delta \mathbf{C}$  is differential operation on contour line  $\mathbf{C}$  which makes

$$\Delta \mathbf{C}(\mathbf{u}_j; \xi_{id}, \alpha) = 2\mathbf{S}(\mathbf{u}_j; \xi_{id}, \alpha) - \sum_{\mathbf{u}_i \in N(\mathbf{u}_j)} \mathbf{S}(\mathbf{u}_i; \xi_{id}, \alpha).$$

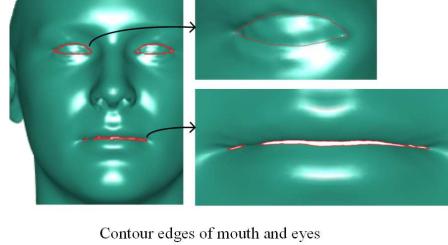


Figure 3. Contours of mouth and eyes labeled by red lines: local contour lines of all neutral faces are assumed to be similar in second order.

Contour line  $\mathbf{C}$  is a parameter point set  $C$ , and the adjacencies between parameter points are defined by  $\{N(\mathbf{u}_i) | \mathbf{u}_i \in C\}$ . And  $\mathbf{C}$  and  $\bar{\mathbf{C}}$  are the contours of  $\mathbf{S}$  and  $\bar{\mathbf{S}}$  respectively, covering the points around eyes and mouth seen in Fig.3.

In above equations, parameters  $\{\pi, \alpha, \beta\}$  are known parameters computed from former steps. The four terms in Equation (8) play different roles in the optimization:

- $E_f(\xi_{id}, \xi_{ex}; \alpha, \beta, \pi)$  is used to fit  $S(\mathbf{u}; \alpha, \beta, \xi_{id}, \xi_{ex})$  that considering both identity parameters  $\{\alpha, \xi_{id}\}$  and expression parameters  $\{\beta, \xi_{ex}\}$ .

- $E_s(\xi_{id}, \xi_{ex}; \alpha, \beta)$  is used to smooth  $S(\mathbf{u}; \alpha, \beta, \xi_{id}, \xi_{ex})$ . Here the smoothing is based on the reference  $\bar{\mathbf{S}}_1$ .

- $E_s(\xi_{id}; \alpha)$  is used to smooth  $S(\mathbf{u}; \alpha, \xi_{id})$  that considering identity part  $\{\alpha, \xi_{id}\}$  only. Here the smoothing is based on a reference  $\bar{\mathbf{S}}_2$ .

- $E_c(\xi_{id}; \alpha)$  restrains the evolution of  $S(\mathbf{u}; \xi_{id}, \alpha)$  in neutral identity space.

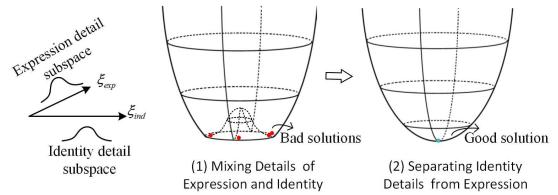


Figure 4. Good solution VS. bad solutions for  $\xi_{id}$  and  $\xi_{ex}$  in expression detail subspace and identity detail subspace respectively: 1) red points indicates that the expression detail and the identity detail are mixed with each other; 2) green point indicates that the expression detail and the identity detail are separated from each other.

The refinement is like to find a optimal solution of  $\xi_{id}$  and  $\xi_{ex}$  in identity detail subspace and expression details subspace respectively as is shown in Fig.4. However, the details of identity and expression are mixed with each other. If only  $E_f$  and  $E_s$  are used, the identity is easily affected by expression part, resulting in bad solution. When term  $E_c$  introduced, contour information in  $\mathbf{S}(\mathbf{u}; \alpha, \xi_{id})$  satisfies the constraint of a neutral face mainly, e.g. the mouth are closed and eye are neutrally opened. By this, identity and

expression details can be separated well. The refinement procedure goes as the following Algorithm 2.

### Algorithm 2

1. Initialize  $\bar{\mathbf{S}}_1$  with  $\mathbf{S}(\mathbf{u}; \alpha, \beta)$ , and  $\bar{\mathbf{S}}_2$  with  $\mathbf{S}(\mathbf{u}; \alpha)$ .
2. Estimate  $\pi$  and compute  $\{w(\mathbf{u}_j)\}$  and  $\{\mathbf{T}(\mathbf{u}_j)\}$ , according to  $\bar{\mathbf{S}}_1$  and  $\mathbf{T}$ ;
3. Get  $\{\xi_{id}, \xi_{ex}\}$  by solving Equ(8);
4. Update  $\bar{\mathbf{S}}_1$  with  $\mathbf{S}(\mathbf{u}; \alpha, \beta, \xi_{id}, \xi_{ex})$ , and  $\bar{\mathbf{S}}_2$  with  $\mathbf{S}(\mathbf{u}; \alpha, \xi_{id})$ ;
5. Termination: check whether the stopping criterion is satisfied for  $\{\xi_{id}, \xi_{ex}\}$ . Continue with step 2 (no) or finish (yes).

### 3.3.3 Updating $\{\mathbf{T}(\mathbf{u}_j)\}$ and $\{w(\mathbf{u}_j)\}$

Point  $\mathbf{T}(\mathbf{u}_j)$  on the target  $\mathbf{T}$  is the correspondence point of  $\mathbf{S}(\mathbf{u}_j)$  on the model  $\mathbf{S}$  being fitted. And  $\{w(\mathbf{u}_j)\}$  are the identifiers of correspondence, which indicates the correspondence and missing data. Nonrigid ICP [3] is a dense registration method that can find a mapping from each point in the template onto the target. It can handle the model transformation with an active stiffness between template and target, e.g., transformation between two different neutral. However, it cannot directly handle the deformation with expression. Therefore, firstly we apply a Laplacian surface editing [26] on  $\mathbf{S}$  by adapting the landmark constraint, so that the deformed  $\mathbf{S}$  has the expression in  $\mathbf{T}$ . Afterwards, we apply Nonrigid ICP to compute the correspondence. At last, for  $j = 1, 2, \dots, N_0$  we define:

$$\mathbf{T}(\mathbf{u}_j) = \arg \min_{\mathbf{t}} \{dist(\mathbf{S}_{deformed}(\mathbf{u}_j), \mathbf{t}) | \mathbf{t} \in \mathbf{T}\},$$

and

$$w(\mathbf{u}_j) = \begin{cases} 1, & \text{if } dist(\mathbf{S}_{deformed}(\mathbf{u}_j), \mathbf{T}(\mathbf{u}_j)) < \varepsilon, \\ 0, & \text{otherwise.} \end{cases}$$

In above equations,  $dist(*, *)$  is Euclidean distance between two points, and  $\mathbf{S}_{deformed}$  is the deformed shape of  $\mathbf{S}$  after using Laplacian editing and Nonrigid ICP, and  $\mathbf{t} \in \mathbf{T}$  means that  $t$  is a point on target  $\mathbf{T}$ . We say  $\mathbf{T}(\mathbf{u}_j)$  is missing when  $w(\mathbf{u}_j) = 0$ .

## 4. Experiments

In this section, to show the effectiveness of SUs and detail refinement in fitting algorithm, our model is tested on both face scan data [20] and KinectfaceDB [17], and the results are compared with classical 3DMM. And to show the performance of detailed fitting for facial expression, we test our model on space-time faces [28] to implement identity separating from expression by face parsing in shape subspace.

### 4.1. Results on Scan data

The face scan data we used are high-quality neutral scan faces along with BFM model from the website [20] of University of Basel. As they have been properly registered, they are very appropriate for comparison between 3DMM and our T-splineMM model on fitting performance. As all the faces to be fitted are neutral, only identity parameters are considered. For comparisons, both T-splineMM and 3DMM are built on only BFM model. We test 4 methods include 3DMM based on global face (3DMM\_Glb), 3DMM based on segments (3DMM\_Segs), T-splineMM without refinement (T-spline\_MM\_SUs), and T-splineMM with refinement (T-spline\_MM\_Ref). Particularly, we test T-splineMM\_SUs and T-splineMM\_Ref to verify the performance of SUs and detail refinement respectively.

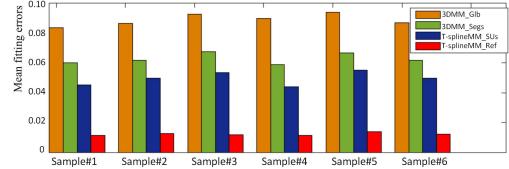


Figure 5. Error Comparisons of four models on 6 samples

Errors of fitting are computed and normalized according to the distance of eye corners. The result and error comparison for 6 samples is shown in Fig.5. Our T-splineMM methods show better performance than 3DMM method on all examples. Table.2 shows the mean errors of 4 models for all samples. 3DMM\_Glb obtains the highest error over 8%. The mean error of 4.99% for T-splineMM\_SUs is about 1.5% lower than that of 6.16% for T-splineMM\_Segs, and T-splineMM\_Ref obtains the lowest mean error 1.21%. So SUs in T-spline model are more effective to improve precision than segments in 3DMM, also seen in Fig.6(b) and (c). And the refinement process makes a result close to the ground truth, as is shown in Fig.6(d).

Table 2. Mean fitting errors of 4 methods

Method	3DMM_Glb	3DMM_Segs
Mean Error	8.68%	6.16%
Method	T-splineMM_SUs	T-splineMM_Ref
Mean Error	4.99%	1.21%

### 4.2. Results on Kinect data

Kinect face data [17] contains low-quality 3D point cloud of faces, and it exists large noise and missing data in them. An important reason we use this database is that it covers several kinds of ethnicities such as Caucasian, East Asian, India, etc. This assures the shape subspace in the test set is less overlapped with the pre-learnt model.

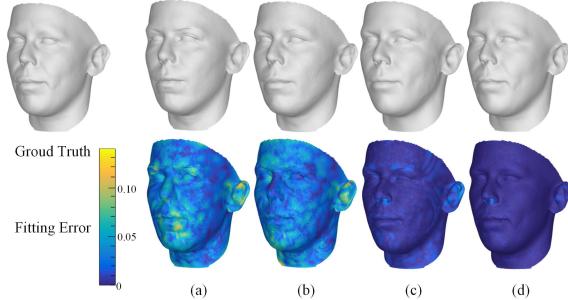


Figure 6. Fitting results and errors on one scan sample by four models: global 3DMM (a), segments based 3DMM (b), T-splineMM without refinement (c) and T-splineMM with refinement (d).

Both 3DMM and T-splineMM here are built on the pre-learnt BFM. We utilize 3DMM\_Segs for 3DMM fitting and T-splineMM\_SU for T-splineMM fitting. Examples of fitting results and comparison are showed in Fig.7. The left results are generated by 3DMM fitting while the right ones are of T-splineMM. The latter results are clearly more accurate than the former ones. On one aspect, it illustrates that T-spline SU can enlarge the morphing span of pre-learnt shape subspace greatly. And another, T-splineMM proves to be robust to missing data, so as to generate smooth and complete results.

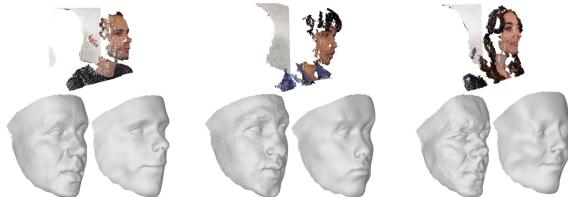


Figure 7. Fitting results on Kinect face data: the left is generated by 3DMM and the right one is by T-splineMM.

#### 4.3. Results on Space-time faces

Space-time faces [28] contains well-reconstructed temporal human face data with expression changing. Continuous frames of 3D expressions provide us useful data for expression analysis. The identity of the faces is East Asia, not overlapped with BFM model in ethnicity. We conduct an experiment of identity separating from expression on 10 samples.

Different from former experiments where only BFM model is used as the pre-learnt model, both 3DMM and T-splineMM here also consider facial expression part by merging the expression model [10] as the pre-learnt expression subspace. For comparison, we implement the 3DMM expression fitting by iteratively optimizing expression and

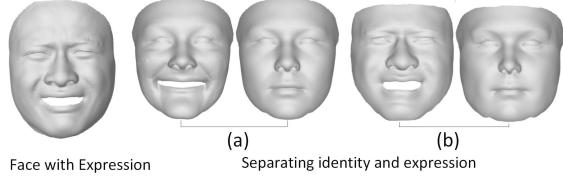


Figure 8. Separating identity from expression data by 3DMM (a) and T-splineMM (b): the left results are the expression reconstruction of fitting, and the right are the identity reconstruction.

identity coefficients. The fitting errors for identity and expression are computed and the statistics of errors are given in Table.3. 3DMM has much larger mean error on expression fitting than T-splineMM as the latter one approaches the ground truth. Then the separated identity is more close to the true neutral face with a mean error of 8.50% by our method, in contrast with the error 10.69% by 3DMM. Comparison on results of one sample is showed in Fig.8. Obviously, T-splineMM method shows a better performance on task of separating identity from expression.

Table 3. Mean fitting errors for expression and identity.

Method	3DMM	T-splineMM
Mean Errors (Ex)	12.64%	0.84%
Mean Errors (Id)	10.69%	8.50%

#### 5. Conclusion

We presented a parametric T-spline morphable model (T-splineMM) for 3D face representation based on pre-learnt identity and expression subspace. Facial SU is defined on T-mesh to enhance the representation performance of T-spline face. An T-splineMM fitting algorithm is proposed to approach the details of both identity and expression. In fact, we solve a problem of incomplete subspace based on two key contributions: local SU definition on T-mesh and refinement solution in fitting algorithm, both of which bring a good performance on various facial deformation. Experimental results on scan data, kinect data and space-time data are presented to demonstrate the effectiveness and robustness of our model to noise, missing data, expression and ethnicity, etc. In the future, T-splineMM will be extended to 2D facial images fitting, or will be introduced into CAD/CAM system as a subspace based free-form surface modeling technology.

#### 6. Acknowledgments

This work is partly supported by National Natural Science Foundation of China (No.61373035 and No.61304262) and National Key Technology R&D Program (No.2015BAH52F00).

## References

- [1] O. Aldrian and W. A. Smith. Inverse rendering of faces with a 3d morphable model. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 35(5):1080–93, 2013. 1
- [2] B. Amberg, R. Knothe, and T. Vetter. Expression invariant 3d face recognition with a morphable model. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 1–6, 2008. 1
- [3] B. Amberg, S. Romdhani, and T. Vetter. Optimal step non-rigid icp algorithms for surface registration. In *IEEE Computer Society Conference on Computer Vision & Pattern Recognition IEEE Computer Society Conference on Cvpr*, pages 1–8, 2007. 5, 7
- [4] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Annual Conference on Computer Graphics*, pages 187–194, 1999. 1, 2, 5
- [5] V. Blanz and T. Vetter. Face Recognition Based on Fitting a 3D Morphable Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1063–1074, 2003. 1
- [6] T. Bolkart and S. Wuhrer. 3d faces in motion: Fully automatic registration and statistical analysis. *Computer Vision & Image Understanding*, 131:100–115, 2015. 2
- [7] J. Booth, A. Roussos, S. Zafeiriou, A. Ponniah, and D. Dunaway. A 3d morphable model learnt from 10,000 faces. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2016. 1, 2
- [8] A. Brunton, T. Bolkart, and S. Wuhrer. *Multilinear Wavelets: A Statistical Shape Space for Human Faces*. Springer International Publishing, 2014. 2
- [9] A. Brunton, A. Salazar, T. Bolkart, and S. Wuhrer. Review of statistical shape spaces for 3d data with comparative analysis for human faces. *Computer Vision & Image Understanding*, 128(11):1C17, 2014. 2
- [10] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Faceware-house: A 3d facial expression database for visual computing. *Visualization & Computer Graphics IEEE Transactions on*, 20(3):413–25, 2014. 1, 2, 5, 8
- [11] B. Chu, S. Romdhani, and L. Chen. 3d-aided face recognition robust to expression and pose variations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1914, 2014. 1, 2
- [12] P. Ekman and W. V. Friesen. Facial action coding system (facs): a technique for the measurement of facial actions. *Rivista Di Psichiatria*, 47(2):126–38, 1978. 1, 3
- [13] R. Feichtinger, H. Yang, and B. Jttler. Particlebased t-spline level set evolution for 3d object reconstruction with range and volume constraints. *Vclav Skala - UNION Agency*, 2008. 2
- [14] M. Gonzlez-Hidalgo, A. Jaume-I-Cap, A. Mir, and G. Nicolau-Bestard. A tool for analytical simulation of b-splines surface deformation. *Computer-Aided Design*, 45(2):168–179, 2013. 2
- [15] B. Levy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *Acm Transactions on Graphics*, 21(3):362–371, 2002. 3
- [16] J. Liao, R. S. Lima, D. Nehab, H. Hoppe, P. V. Sander, and J. Yu. Automating image morphing using structural similarity on a halfway domain. *Acm Transactions on Graphics*, 33(5):1–12, 2014. 2
- [17] R. Min, N. Kose, and J. L. Dugelay. Kinectfacedb: A kinect database for face recognition. *Systems Man & Cybernetics Systems IEEE Transactions on*, 44(11):1534–1548, 2014. 7
- [18] J. Montagnat, H. Delingette, and N. Ayache. A review of deformable surfaces: topology, geometry and deformation. *Image & Vision Computing*, 19(14):1023–1040, 2001. 2
- [19] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In *IEEE International Conference on Advanced Video & Signal Based Surveillance*, pages 296–301, 2009. 2, 5
- [20] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. Well-registered scans data from website of basel face model. <http://faces.cs.unibas.ch/bfm/>, 2009. 7
- [21] W. Peng, C. Xu, and Z. Feng. 3d face modeling based on structure optimization and surface reconstruction with b-spline. *Neurocomputing*, 179(12):228–237, 2016. 2
- [22] L. Piegl and W. Tiller. *The NURBS book*. Springer-Verlag, 1995. 2
- [23] J. Roth, Y. Tong, and X. Liu. Unconstrained 3d face reconstruction. In *IEEE Computer Vision and Pattern Recognition*, pages 2606–2615, 2015. 2
- [24] J. Roth, Y. Tong, and X. Liu. Adaptive 3d face reconstruction from unconstrained photo collections. In *Proc. IEEE Computer Vision and Pattern Recognition*, 2016. 1
- [25] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and t-nurccs. *Acm Transactions on Graphics*, 22(3):477–484, 2002. 2
- [26] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Ssl, and H. P. Seidel. Laplacian surface editing. In *Eurographics/acm SIGGRAPH Symposium on Geometry Processing*, page 54A, 2004. 2, 7
- [27] H. Yang and B. Jttler. 3d shape metamorphosis based on t-spline level sets. *The Visual Computer*, 23(12):1015–1025, 2007. 2
- [28] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Space-time faces: High-resolution capture for modeling and animation. In *Data-Driven 3D Facial Animation*, pages 248–276. Springer, 2007. 7, 8
- [29] X. Zhu, Z. Lei, J. Yan, and D. Yi. High-fidelity pose and expression normalization for face recognition in the wild. pages 787–796, 2015. 1, 2