



# Statistical Modeling of Craniofacial Shape and Texture

Hang Dai<sup>1,2</sup> · Nick Pears<sup>1</sup> · William Smith<sup>1</sup> · Christian Duncan<sup>3</sup>

Received: 3 December 2018 / Accepted: 28 October 2019 / Published online: 9 November 2019  
© The Author(s) 2019

## Abstract

We present a fully-automatic statistical 3D shape modeling approach and apply it to a large dataset of 3D images, the *Headspace dataset*, thus generating the first public shape-and-texture 3D morphable model (3DMM) of the full human head. Our approach is the first to employ a template that adapts to the dataset subject before dense morphing. This is fully automatic and achieved using 2D facial landmarking, projection to 3D shape, and mesh editing. In dense template morphing, we improve on the well-known Coherent Point Drift algorithm, by incorporating iterative data-sampling and alignment. Our evaluations demonstrate that our method has better performance in correspondence accuracy and modeling ability when compared with other competing algorithms. We propose a texture map refinement scheme to build high quality texture maps and texture model. We present several applications that include the first clinical use of craniofacial 3DMMs in the assessment of different types of surgical intervention applied to a craniosynostosis patient group.

**Keywords** 3D morphable model · Statistical shape model · Craniofacial shape · Shape morphing

## 1 Introduction

Very young children quickly learn to understand the rich shape and texture variation in a certain class of object, such as human faces, cats or chairs, in both 2D and 3D. This ability helps them to recognize the same person, distinguish different kinds of creatures, and sketch unseen samples of the same object class. In machine learning, the process of capturing this prior knowledge is mathematically interpreted as statistical modeling. One such realisation of this is a 3D Morphable Model (3DMM) (Banz and Vetter 1999), a vector space representation of objects, that captures the variation of

shape and texture. Any convex combination of vectors of a set of object class examples generates a real and valid example in this vector space. 3DMMs provide a useful encoding and prior statistical distribution of both shape and texture that can be used as a constraint in analysis problems or generatively in synthesis problems. Modeling shape and texture variability of the human face is of particular interest because of the wide variety of related applications, such as in biometrics (An et al. 2018), affective computing (Garrido et al. 2016) and creative media (Saragih et al. 2011).

Here, we are concerned with 3D statistical shape modeling of craniofacial data, i.e. models of the variability of the full head and face. A full head model opens up new applications and introduces useful constraints that are not afforded by existing 3D face models. In graphics, knowledge of full head shape is necessary for modeling hairstyles onto a correctly proportioned cranium (Petr and Ivana 2015). In ergonomics, predicting the fit of headwear objects such as helmets, spectacles and breathing apparatus requires modeling the fit over the full head region (Harrison and Robinette 2006). In forensics, reconstructing face models from skulls (Madsen et al. 2018) could be further constrained if the relationship between skull and outer face surface were modelled over the whole skull. In craniofacial syndrome diagnosis and surgical planning and evaluation, a full head model is a prerequisite for comparing syndromic or post-surgical head shape to popu-

Communicated by Cristian Sminchisescu.

✉ Nick Pears  
nick.pears@york.ac.uk

Hang Dai  
hang.dai.cs@gmail.com

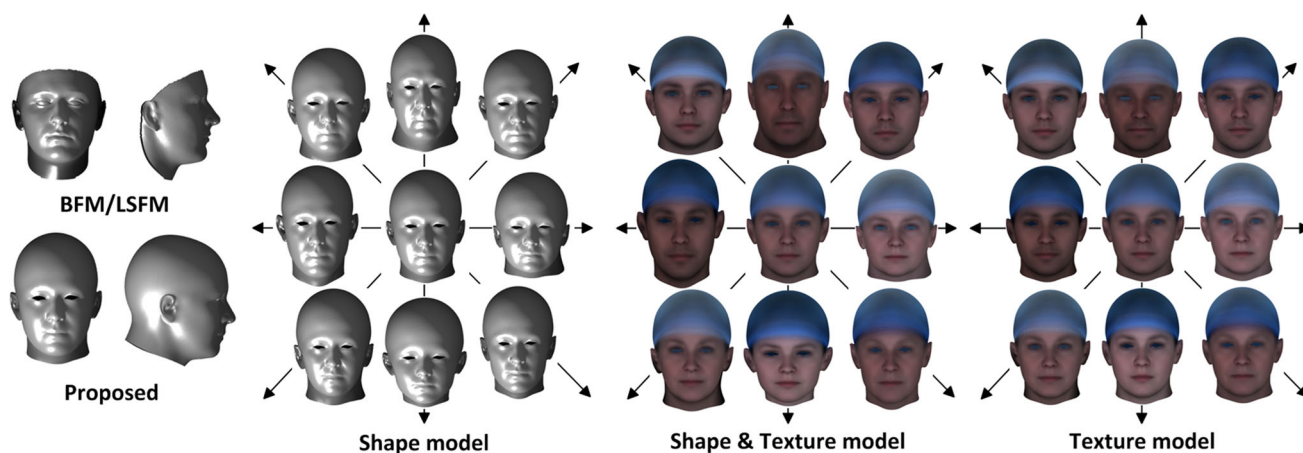
William Smith  
william.smith@york.ac.uk

Christian Duncan  
Christian.Duncan@alderhey.nhs.uk

<sup>1</sup> Department of Computer Science, University of York, York, UK

<sup>2</sup> Present Address: Inception Institute of AI, Abu Dhabi, UAE

<sup>3</sup> Alder Hey Hospital, Liverpool, UK



**Fig. 1** The proposed global Liverpool–York Head Model (LYHM) trained using 1212 subjects (606 male, 606 female) from the Headspace dataset. 1st block: shape compared to the Basel Face Model (BFM) (Paysan et al. 2009) and the Large Scale Face Model (LSFM) (Booth

et al. 2016); 2nd–4th blocks: the central head is the mean and the first four principal modes of variation are shown along four axes, covering  $\pm 3$  standard deviations of shape and texture variation

lation norms and for proposing surgical interventions (Dai et al. 2017a). In face animation, the skull location can be used to stabilize rigid motion of a performance (Beeler and Bradley 2014). Estimating skull location would be considerably simplified with a full head model. In computer vision, a full head model enables prediction or completion of the unobserved back of the head from face images or from silhouettes, which has potential applications in biometrics, and provides a mechanism to address any of the other aforementioned applications.

These rich applications motivate our work, but building full head models brings with it new challenges that are not confronted by face-only models. To capture the rich variability in craniofacial shape variation requires a dataset of 3D scans that covers the whole head and face area and is diverse enough to sample the full space of variation. The first challenge is that cranial shape usually cannot be directly observed (due to hair or headwear) and many scanning systems only cover the ear-to-ear region, so no suitable dataset previously existed. Second, with large-scale data, the model construction pipeline must be fully automatic to avoid costly and unreliable manual intervention. Third, building a 3DMM requires establishment of dense correspondence between all training samples. The cranium and neck region dominate the face in terms of surface area, yet are largely devoid of distinctive features. This makes meaningful correspondence difficult to compute in the cranial area and also risks sacrificing quality of correspondence in the face area, as the cranium dominates.

We present a fully automatic 3DMM training pipeline and apply it to a large dataset of 3D images of the human head, thus generating a shape-and-texture 3D morphable model of the full human head: the Liverpool–York Head

Model (LYHM). The model is named as such to reflect the data collection and model training respectively. As illustrated in Fig. 1, the Basel Face Model (BFM) (Paysan et al. 2009) and the Large Scale Face Model (LSFM) (Booth et al. 2016) are face-only models employing the same template, whereas the LYHM is the first publicly-available full head model. The contributions of this work are as follows:

- We propose a fully automatic pipeline to train 3DMMs that: i. is the first to employ an adaptive template; ii. employs a new morphing algorithm that integrates ideas from Iterative Closest Points (Besl and McKay 1992) and Coherent Point Drift (Myronenko and Song 2010) and iii. employs regularized projection to reduce morphing error.
- We present a high-quality, multi-view texture mapping method and employ it for texture modeling.
- We build both global craniofacial 3DMMs and demographic sub-population 3DMMs from 1212 distinct identities in the Headspace dataset and we make both the 3DMMs [improved from our earlier public release in Dai et al. (2017b)] and the Headspace dataset publicly available (Duncan et al. 2018). To our best knowledge, our models are the first public shape-and-texture craniofacial 3DMMs of the full human head.
- We demonstrate a wide range of applications that demonstrate the power of our 3DMMs, these include: i. the flexibility modes of shape completion, ii. age-based regression of craniofacial shape, and iii. clinical assessment of craniofacial surgery.

## 2 Related Work

The first 3DMM of the human face was presented by Blanz and Vetter (1999). Here, 3D face scans of 200 young adults, evenly split between male and female, were used to construct the model. Dense correspondences were computed using optical flow with an energy term dependent on both shape and texture. Independent shape and texture models were developed, where each was constructed using Principal Component Analysis (PCA). This was achieved in an iterative bootstrapping process, where the expressive power of the model was gradually increased by increasing the number of model components. Later the same authors described how to do face recognition within an analysis-by-synthesis setting (Blanz and Vetter 2003). This was achieved by fitting the 3D face model, along with pose and illumination model parameterisation, to a single facial image.

Many important works had led up to the first 3DMM, at least dating from the *transformation grid* work of Thompson et al. (1917). Other key milestones include the shape-space work of Kendall (1984), work on Thin Plate Splines by Bookstein (1989), the theoretical underpinnings of statistical shape modeling by Dryden et al. (1998), Point Distribution Models (PDMs) by Cootes and Taylor (1995), Active Shape Models (ASMs) (Cootes et al. 1995), and Active Appearance Models (AAMs) Cootes et al. (2001), to name a few.

More recently, the Basel Face Model (BFM) has become the most well-known and widely-used 3DMM of the human face and was developed by Paysan et al. (2009). As with Blanz and Vetter's model, 200 scans were used, but the method of determining corresponding points was improved. Instead of optical flow, a set of hand-labelled feature points is marked on each of the 200 training scans. The corresponding points on a template mesh are known, which is then morphed onto the training scan using under-constrained per-vertex affine transformations, which are constrained by regularisation across neighbouring points (Amberg et al. 2007). The technique is known as optimal-step Non-rigid Iterative Closest Points (NICP).

The BFM was released as both a global model and a part-based model that is learned for four regions: the eyes, nose, mouth and the rest of face. In the part-based version, the manually-defined regions are fitted to the data independently and merged in a post-processing step (ter Haar and Veltkamp 2008; Basso et al. 2007). The part-based model was shown to lead to a higher data accuracy than the global model. De Smet and Van Gool (2010) proposed a method to find the optimal segmentation automatically by clustering the vertices, which is based on features derived from their displacements. In order to address the potential discontinuities at the boundaries of the segments, they smoothly weight the segments to obtain regionalized basis functions for the training data.

A multilinear model has been employed by several authors (Vlasic et al. 2005; Yang et al. 2011; Bolkart and Wuhler 2013; Yang et al. 2012) to capture varying facial expressions. Vlasic et al. (2005) modelled facial shape using a combination of identity and expression variation. Yang et al. (2011) modelled the expression of a face in a different input image of the same subject. A number of PCA shape spaces for each expression are built and combined with a multilinear model. A follow-up work (Bolkart and Wuhler 2013; Yang et al. 2012) used this model for a better description of expressions in videos. When a sequence of 3D meshes is given, Bolkart and Wuhler (2013) fitted a multi-linear model to parameterize a 4D sequence. Later, they demonstrated a direct construction of a multilinear model from a set of meshes using a global optimization of 3DMM parameters along with a group-wise registration over the 3D scans (Bolkart and Wuhler 2015). Another alternative to modeling faces with expression is to blend different shape models with expressions, which was introduced by Salazar et al. (2014) to establish correspondence among faces with expression.

A hierarchical pyramids method was introduced by Golovinskiy et al. (2006) to build a localized model. In order to model the geometric details in a high resolution face mesh, this statistical model is able to describe the varying geometric facial detail. Brunton et al. (2011) described 3D facial shape variation at multiple scales using a wavelet basis. The wavelet basis provided a way to combine small signals in local facial regions that are difficult for PCA to capture.

Recently, Booth et al. (2016) built a Large Scale Facial Model (LSFM), using the NICP template morphing approach, as was used in the BFM, but with error pruning, followed by Generalized Procrustes Analysis (GPA) for alignment, and PCA for the model construction. This 3DMM employs the largest 3D face dataset to date, and is constructed from 9663 distinct facial identities.

Lüthi et al. (2017) model the shape variations with a Gaussian process, which they represent using the leading components of its Karhunen–Loeve expansion. Such Gaussian Process Morphable Models (GPMMs) unify a variety of non-rigid deformation models with B-splines and PCA models as examples. In their follow-on work, they present a novel pipeline for morphable face model construction based on Gaussian processes (Gerig et al. 2017). GPMMs separate problem-specific requirements from the registration algorithm by incorporating domain-specific adaptations as a prior model.

Tran and Liu (2018) proposed a framework to construct a nonlinear 3DMM model from a large set of unconstrained face images, without collecting 3D face scans. Specifically, given a face image as input, a network encoder estimates the projection, shape and texture parameters. Two decoders served as the nonlinear 3DMM to map from the shape and texture parameters to the 3D shape and texture, respectively.

The work presented here builds on our earlier conference publication (Dai et al. 2017b) that introduced the first publicly available 3DMM of the human head. In that paper, we used a hierarchical parts-based approach to shape morphing. Here, we use an adaptive template approach to personalize the template to the subject's facial features before dense morphing. The dense morphing algorithm itself then employs a new algorithm called Iterative Coherent Point Drift (ICPD), combining concepts from the well-known ICP (Besl and McKay 1992; Chen and Medioni 1992) and CPD (Myronenko and Song 2010) algorithms. In Sect. 9.2, we demonstrate that the new morphing results are superior to our earlier approach and we update our LYHM 3DMM public release with a version that uses this improved morphing. Our other earlier work introduced a symmetrized version of the CPD morphing algorithm (Dai et al. 2018a), and we also evaluate against this pipeline here, although symmetry is not a central consideration of this paper. Furthermore, in this paper, we use the constructed 3DMM in the first 3DMM-based clinical assessment of craniofacial surgery.

### 3 Overview of 3DMM Training

Our 3DMM training pipeline has three main functional blocks:

*i. Data preprocessing* We use automatic 2D landmarking and map to 3D using the known 2D-to-3D registration supplied by the 3D camera system. These 3D landmarks can then be used for both pose normalisation and template adaptation (personalization of the template).

*ii. Dense correspondence* A collection of 3D scans are reparametrized into a form where each scan has the same number of points joined into a mesh triangulation that is shared across all scans. This is achieved by non-rigid template deformation. Furthermore, the semantic or anatomical meaning of each point is shared across the collection, as defined by the template mesh. We use the publicly-available *FaceWarehouse* head mesh (Cao et al. 2014) as our template, which has 11510 vertices.

*iii. Alignment and statistical modelling* The collection of scans in dense correspondence are aligned using Generalized Procrustes Analysis (GPA). Then Weighted Principal Component Analysis (WPCA) is applied, generating a 3DMM.

In Sect. 4, we give an overview of our training dataset. Section 5 describes data preprocessing including pose normalisation and 3D facial landmark detection. Section 6 presents the correspondence establishment method using an adaptive template. Similarity alignment and statistical modeling are described in Sect. 7. After dealing with shape modeling, Sect. 8 presents a texture mapping method that can capture the same resolution as the captured image, thus allowing us to train a high resolution texture model.



**Fig. 2** Left: 3dMD's static 5-viewpoint *3dMDhead* system was used to capture the *Headspace* dataset. Each of the 5 viewpoints is a 3D camera system that has an IR projector, IR camera and RGB camera. The system stitches the 5 point clouds into a single 3D output mesh, and provides the texture coordinates for texture mapping. Centre: texture mapped output mesh. Right: 3D point cloud output

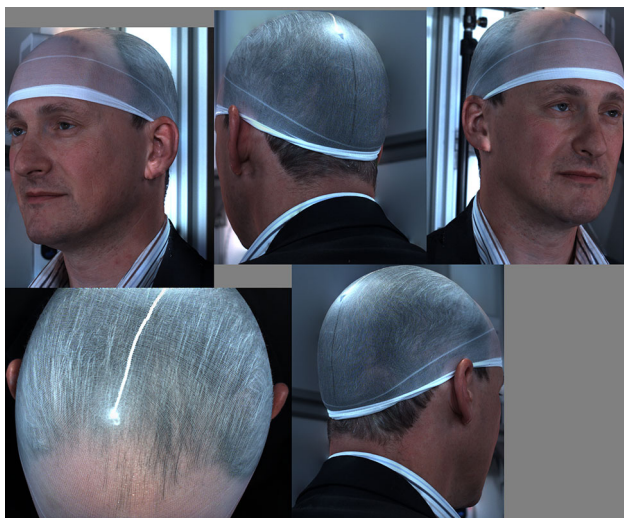


**Fig. 3** *Headspace* dataset example. Left: texture mapped mesh rendered from two viewpoints. Right: monochrome mesh rendered from the same two viewpoints

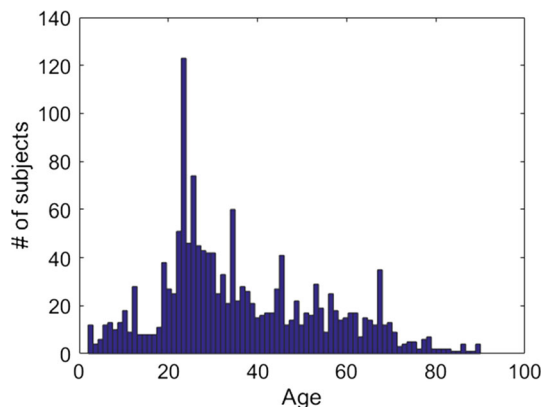
### 4 Overview of the *Headspace* Dataset

The *Headspace* dataset comprises 3D images of the human head for 1519 subjects. The data was collected and annotated by the Alder Hey Children's Hospital (AHCH) Craniofacial Unit (Liverpool, UK), who employed 3dMD Ltd's static 5-view *3dMDhead* scanning system, using the five 3D camera configuration shown in Fig. 2. This dataset has been structured and made available online for research purposes, in a collaboration between AHCH and the Department of Computer Science, University of York. Access to the dataset is via the author's *Headspace* web page (Duncan et al. 2018).

A typical output of this system rendered from different viewpoints, both with and without texture, is shown in Fig. 3. Vertex resolution is variable but typically there are around 180K vertices. All subjects are wearing tight fitting latex caps to reduce the effect of hairstyles. For subjects with relatively low-volume hairstyles, the shape of the cranium is clearly revealed. If this is not the case, we exclude them from the 3DMM training data, filtering on the basis of the *hair bulge* flag in the metadata.



**Fig. 4** Headspace 5-view composite texture image example



**Fig. 5** Age distribution of the subjects in the Headspace dataset. The age range is wide (1–89 years), but the distribution is non-uniform, with a high proportion of subjects in their twenties

The dataset employs the OBJ format, which contains a list of the 3D vertex coordinates, a set of vertex index triples that form the mesh faces, and a set of UV texture coordinates that map each vertex into the supplied 5-view composite image (BMP format). An example is shown in Fig. 4, which allows texture mapping over the whole mesh from a single texture image.

The dataset is supplied with subject-based metadata, capture-based metadata and a set of 3D landmark coordinates extracted using the Zhu-Ramanan mixture-of-trees algorithm (Zhu and Ramanan 2012). The subject information includes: gender, declared ethnic group, age, eye color, hair color, beard descriptor (none, low, medium, high), moustache descriptor (none, low, medium, high), and a spectacles flag. The capture information contains a quality descriptor (free text, such as ‘data spike right upper lip’), a *hair bulge* flag (hair bulge under latex cap distorting the apparent cranial shape), a *cap artefact* flag (cap has a ridge at its apex due to

poor fitting), a *cranial hole* flag (a missing part in the data scan at the cranium) and an *under chin hole* flag (missing part under chin).

The dataset is well-balanced in gender, but not age, which is predominantly 20s, see Fig. 5. However, the age range is wide, from 1 to 89 years. Also it is not well-balanced in declared ethnicity, which is predominantly ‘white’, with 90% ‘white’, 5.3% ‘asian’, 2.7% ‘mixed heritage’, 1% ‘black’ and 1% ‘other’. Eye color is distributed as 33.36% brown, 46.38% blue, 19.89% green and 0.37% other.

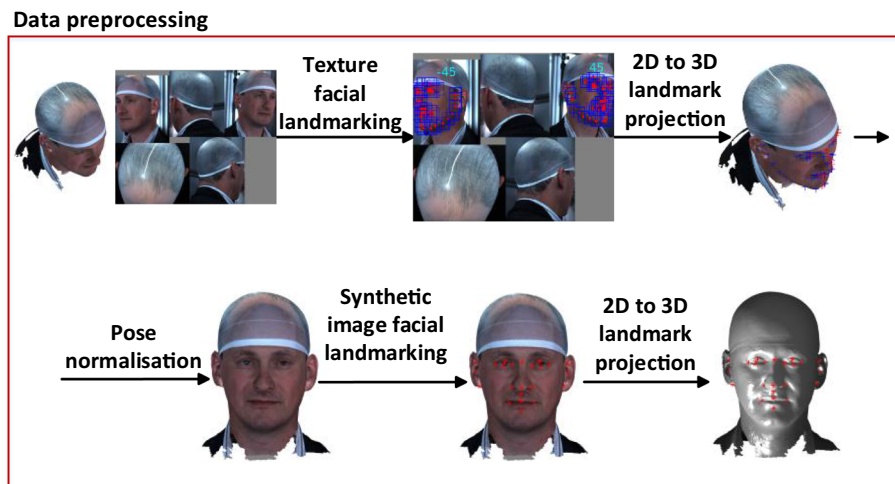
## 5 Data Preprocessing

Preprocessing of the 3D scan serves to place the data in a frontal pose and localise a complete and accurate set of automatic 3D landmark positions, for every 3D image, that corresponds to a set of manually-placed landmarks on the template. Placing manual landmarks on the template is done only once, there is no manual landmarking on a per-subject basis.

Figure 6 illustrates the five stages of preprocessing: i. 2D landmarking, ii. projection to 3D landmarks, iii. pose normalisation, iv. synthetic frontal 2D image landmarking, and v. projection to 3D landmarks. Thus there are two phases of landmarking—the first to approximately normalize pose, and the second to more accurately localize landmarks (now including some around the ears) that correspond to the one-shot manual annotation of the template. This latter stage is employed both for refined template alignment and template adaptation to the subject’s features.

In more detail, the first stage employs the *Mixture of Trees* method of Zhu and Ramanan (2012) to localize 2D facial landmarks on the 5-view composite texture image. Although there are more recent network-based landmarkers (see Wu and Ji (2019) for a review), this method works highly reliably on non-frontal poses, as captured in our 5-view composite 2D images (see Fig. 4), where there are typically two facial views at around  $\pm 45^\circ$  to the frontal view. In all cases, at least one view was successfully landmarked (in 99% of cases both views are landmarked), so all 1212 3DMM training images could be pose normalized. Our framework may be used with any other landmarker than can handle pose variations.

The mixture of trees that we use has 13 landmark tree models (‘components’) for 13 different yaw angles of the head. For each subject, two face detections are found, corresponding to the left and right side of the face. The detected 2D landmarks are then projected to 3D using the OBJ texture coordinates in the data. Given that we know where all 3D landmarks should be for a frontal pose, it is possible to do standard 3D pose alignment in a scale-normalized setting (Dai et al. 2017b).



**Fig. 6** Data preprocessing with two-stage landmarking. Top left: system input is a texture-mapped 3D shape and a composite 2D image of the five camera views. Top-middle: Zhu–Ramanan landmarker (Zhu and Ramanan 2012) applied to the composite 5-view image. Top-right: 2D landmarks projected to 3D surface using texture coordinates.

Bottom-left: pose-normalisation of 3D shape using 3D landmarks, then rendered to a frontal view. Bottom centre: synthetic frontal image is more accurately landmarked, and now includes landmarks around the ears. Bottom-right: new set of landmarks projected to 3D surface, again using texture coordinates

We automatically learn how to orientate each of the detected trees to frontal pose, based on their 3D structure. To do this, we apply Generalized Procrustes Analysis (GPA) to each collection of 3D tree components and find the nearest-to-mean tree shape in a scale-normalized setting. We do not have any clear semantic meaning of the landmarks in the nearest-to-mean tree and therefore we don't know their relative target positions when normalising to a frontal pose. Therefore, we apply a 3D face landmarker Creusot et al. (2013) to the 3D data of the nearest-to-mean tree shape, which generates a set of 14 near-symmetric landmarks, each with clear semantic meaning. This landmark set is easily frontal-pose normalized. Here, we find the alignment that moves the symmetry plane of these 14 landmarks to the Y–Z plane and positions the nasion directly above the subnasale to normalize the tilt angle. To complete the training phase, the mean 3D tree points for each of the 13 trees are then carried into this frontal pose using the same rotation, and are used as reference points for the frontal pose normalisation of the 3D trees.

In around 1% of the dataset, only one tree is detected and that is used for pose normalisation and, in the rest, 2–3 images are detected. In the cases where 3 trees are detected, the lowest scoring tree is always false positive and can be discarded. For the remaining two trees, a weighted combination of the two rotations is computed using quaternions, where the weighting is based on the mean Euclidean error to the mean tree, in the appropriate tree component.

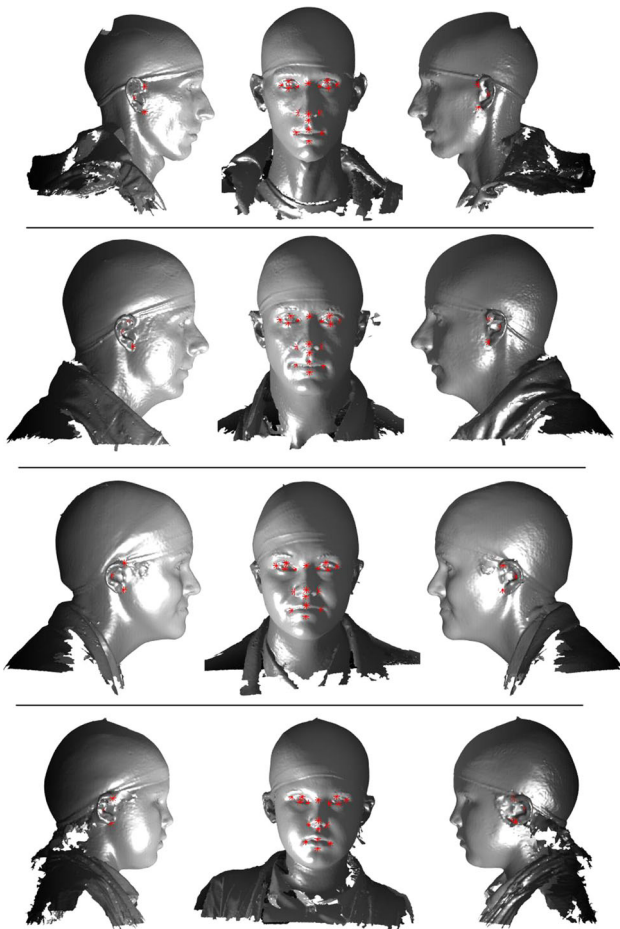
After we have rotated the 3D image to a frontal view and generated a synthetic frontal 2D image, we wish to generate a set of landmarks that are accurate and correspond to the

set marked up on the template. This is the set related to the central tree ( $0^\circ$  yaw) in the mixture, and we subsample 17 of these 68 landmarks around the eyes (indices 28, 37, 39, 40, 42, 43, 44, 46, 47) nose base (indices 31, 32, 34, 36) and mouth (indices 49, 52, 55, 58). After these 2D facial landmarks are extracted, they are again projected onto 3D mesh.

We needed to augment this set of landmarks with some landmarks around the ear region to better control template adaptation, otherwise the final template around the ear region will be poor for many subjects. The work of Zhou and Zafeiriou (2017) shows that an Active Appearance Model (AAM) (Cootes et al. 2001) with SIFT features (Lowe 2004) has excellent performance in ear landmark detection. However, we just need three non-colinear ear landmarks for ear alignment, which is a simpler task than that solved in Zhou and Zafeiriou (2017). For each subject, we generate two synthetic images in profile view by rotating  $90^\circ$  and  $-90^\circ$  yaw. We use an AAM with SIFT features to detect the three ear landmarks on each synthetic image and then project the 2D landmarks onto 3D mesh. The detected facial landmarks and ear landmarks are shown in Fig. 7.

## 6 Correspondence Establishment

We employ template morphing as a means of establishing correspondence across our 3DMM training dataset. However, very low error non-rigid shape morphing over a diverse set of target shapes is still a challenging problem. The true underlying shape transformation of the template to the data is



**Fig. 7** Landmark detection results: 1st column—3 right ear landmarks; 2nd column—17 facial landmarks; 3rd column—3 left ear landmarks

very different for different head shapes and we require a technique that permits an accurate mapping between target points (corresponding landmarks), while regulating the deformation of the remaining mesh. The use of Gaussian Process models in morphing is a leading recent approach (Gerig et al. 2017), whereas the use of the Laplace–Beltrami operator in *As Rigid As Possible* shape regulation (Sorkine and Alexa 2007) is a leading traditional approach. These were natural choices for us to evaluate within our pipeline—in particular they are employed to adapt, or personalize, the template to each individual subject, before dense morphing proceeds.

Now we present a new fully-automatic non-rigid 3D shape registration pipeline by integrating several powerful ideas from computer vision and graphics. These include Iterative Closest Points (ICP) (Besl and McKay 1992), Coherent Point Drift (CPD) (Myronenko and Song 2010), and mesh editing using the Laplace–Beltrami (LB) operator (Sorkine and Alexa 2007). As mentioned, we also provide comparisons of the latter approach with the use of Gaussian Process Morphable Models (GPMMs) (Gerig et al. 2017).

Our contributions include: i. an adaptive shape template method to accelerate the convergence of registration algorithms *and* achieve a better final shape correspondence and ii. a new iterative registration method that combines ICP with CPD to achieve a more stable and accurate correspondence establishment than standard CPD alone. We call this approach *Iterative Coherent Point Drift* (ICPD). These two processing stages are illustrated in Fig. 8, and are presented in Sects. 6.1 and 6.2 respectively.

Figure 9 is a qualitative illustration of a typical result where our method achieves a more accurate correspondence than standard CPD. Note that the landmarks in our method are almost exactly the same position as their corresponding ground-truth points on the 3D scan. Even though standard CPD-affine is aided by Laplace–Beltrami Regularized Projection (LBRP, a component of our new pipeline), the result shows a *squeezed* face around the eye and mouth regions and the landmarks are far away from their corresponding ground-truth positions.

## 6.1 Template Adaptation

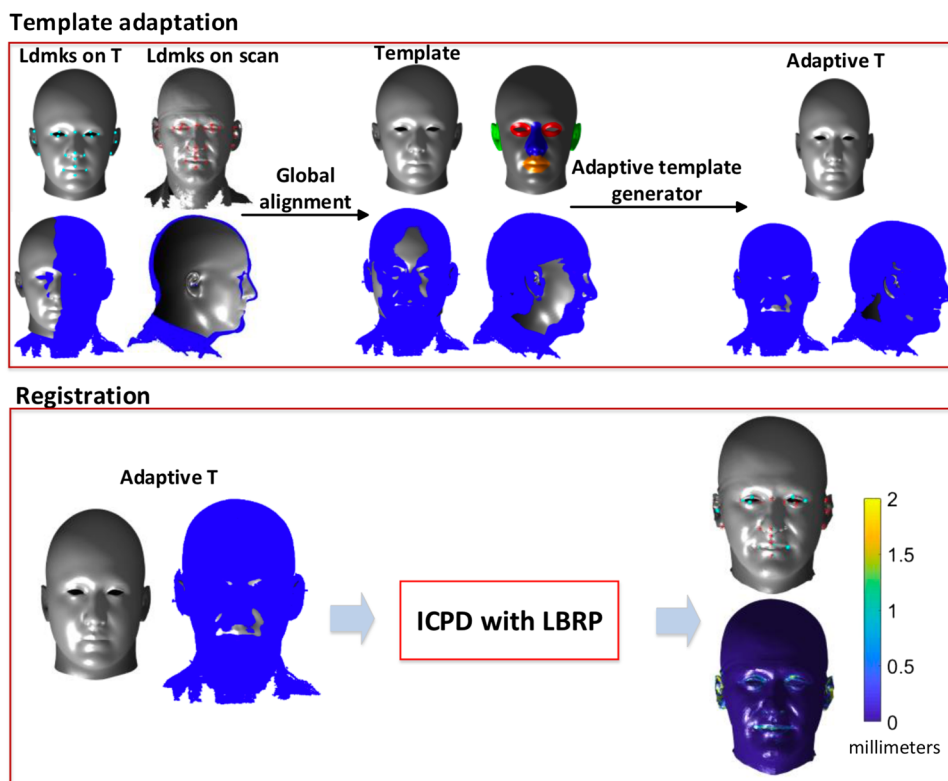
As shown in Fig. 8, template adaptation consists of two sub-stages: i. global alignment followed by ii. dynamically adapting the template shape to the data. For global alignment, we manually select the same landmarks on the template as we automatically extract on the data i.e. using the 17 landmarks sampled from the zero yaw angle tree component from Zhu and Ramanan (2012), augmented with three additional landmarks per ear. Note that this needs to be done *once only* and so doesn't impact on the autonomy of the *online* operation of the framework. Then we align rigidly (without scaling) from the 3D landmarks on 3D data to the same landmarks on the template. The rigid transformation matrix is used for the data alignment to the template.

The template is then *adapted* to better align with the scan. A better template helps the later registration converge faster and gives more accurate correspondence at the beginning and end of registration. A good template has the same size and position of local facial parts (e.g. eyes, nose, mouth and ears) as the scan. This cannot be achieved by mesh alignment alone. We evaluate two methods to give a better template that is adapted to the 3D scan: i. Laplace–Beltrami mesh editing; ii. Template estimation via posterior GPMMs. For both methods, three ingredients are needed: landmarks on 3D data, the corresponding landmarks on template, and the original template.

### 6.1.1 Laplace–Beltrami Mesh Manipulation

We decompose the template into several facial parts: eyes, nose, mouth, left ear and right ear. We rigidly align landmarks on each part separately to their corresponding landmarks

**Fig. 8** Template morphing framework using an adaptive template. Top left: A set of corresponding landmarks have been found on both the template and the 3D scan data. Top centre: corresponding landmarks are used for global least-squares alignment. Top right: the template is adapted so that the colored facial parts match up between the template and the data, and all other parts of the template move as-rigidly-as-possible. The blue overlays represent the 3D scan data and are included to show how well the template (grey surface with smooth neckline) and the scan match. Bottom-left: The adapted template is densely morphed and registered using our ICPD process followed by our LBRP process. Bottom-right: the color-mapped nearest-point vertex registration errors are in millimeters



on 3D data. These rigid transformation matrices are used for aligning the decomposed parts to 3D data. The rigidly transformed facial parts tell the original template where it should be. We treat this as a mesh manipulation problem. We use Laplace–Beltrami mesh editing to manipulate the original template towards the rigidly transformed facial parts, as follows: i. the *facial parts* (fp) of the original template are manipulated towards their target positions—these are rigidly transformed facial parts; ii. all other parts of the original template are moved *As Rigidly As Possible* (ARAP) (Sorkine and Alexa 2007).

Given an  $n$ -vertex 3D data scan,  $\mathbf{Y} \in \mathbb{R}^{n \times 3}$  (shortened to a ‘scan’ from here onwards) and a  $p$ -vertex template  $\mathbf{X} \in \mathbb{R}^{p \times 3}$ , we formulate the problem, which balances the desirable outcomes in (i) and (ii) above, as the following weighted linear least squares problem:

$$\begin{pmatrix} \lambda \mathbf{L}_0 \\ \mathbf{S}_{\mathbf{X}_{\text{fp}}} \end{pmatrix} \mathbf{X}_1 = \begin{pmatrix} \lambda \mathbf{L}_0 \mathbf{X}_0 \\ \mathbf{S}_{\mathbf{Y}_{\text{fp}}} \mathbf{Y} \end{pmatrix}, \tag{1}$$

where  $\mathbf{X}_0$  is the initial (unadapted) template and  $\mathbf{X}_1$  is the adapted template that we are solving for. The upper row of the block matrix structure employs the cotangent approximation to the Laplace Beltrami operator,  $\mathbf{L}_0 \in \mathbb{R}^{p \times p}$ , computed from  $\mathbf{X}_0$ , to retain the overall template mesh shape (the ARAP constraint), while the lower row enforces matching of the vertices defining the facial parts. The selection matrix  $\mathbf{S}_{\mathbf{X}_{\text{fp}}} \in [0, 1]^{l \times p}$  selects  $l$  vertices from the template that

belong to the facial parts (fp) and  $\mathbf{S}_{\mathbf{Y}_{\text{fp}}} \in [0, 1]^{l \times n}$  selects the corresponding vertices of the facial parts from the scan.

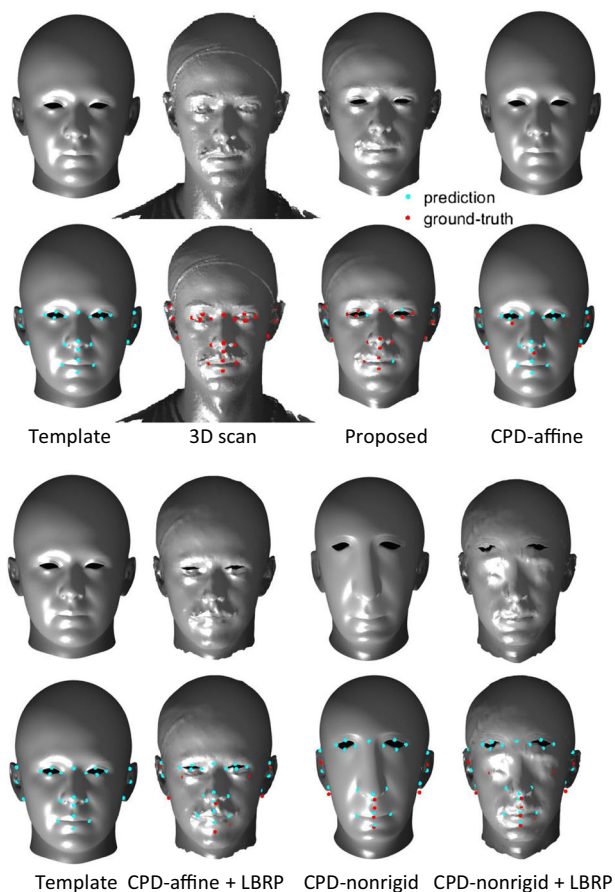
The parameter  $\lambda$  weights the relative influence of the position and regularisation constraints, effectively determining the ‘stiffness’ of the mesh manipulation. As  $\lambda \rightarrow 0$ , the facial parts of the original template are manipulated exactly to the rigidly transformed facial parts, but the template mesh shape is not retained. As  $\lambda \rightarrow \infty$ , the adapted template will retain the shape of the original template,  $\mathbf{X}_0$ , but will not be well-adapted to the subject’s face and head shape. A suitable  $\lambda$  value is chosen to give a good trade off.

### 6.1.2 Template Adaption Using Gaussian Process Posterior Models

Gaussian Process Morphable Models (GPMMs) allow a more general formulation of 3D shape deformation than PCA-based statistical shape models (Lüthi et al. 2017). Firstly, they operate on a continuous rather than discrete domain (which can of course be sampled, as required). Secondly, they allow a wider range of covariance formulations, such as those that don’t require training data. Therefore, we can use them to train statistical shape models and we now exploit GPMMs as an alternative to using the ARAP constraint for template adaptation.

Our aim is to employ the posterior Gaussian Process formulation to solve a regression problem. We present and later





**Fig. 9** Proposed method compared with other CPD-based methods. Ground truth points on 3D scan are shown in red, and the corresponding template points are shown in cyan. Top two rows, from left to right, show the template, the scan, the proposed method and the CPD-affine method. The first row shows the head shapes without landmarks and the second row shows the head shapes with landmarks. Bottom two rows, from left to right, show the template again (for convenience), the CPD-affine with LBRP method, the CPD-nonrigid method and the CPD-nonrigid with LBRP method

evaluate this approach to give a comparison with the ARAP approach. As before, the aim is to infer the full shape from a set of landmark positions on the shape. Given partial observations, such GPMMs are able to determine the potential full shape. They show the probable range of motion of all the vertices in the shape, when the landmarks are fixed to their target positions.

In a GPMM, let  $\{\mathbf{x}_1, \dots, \mathbf{x}_l\} \in \mathbb{R}^{l \times 3}$  be a fixed set of  $l$  vertices on the template mesh, and assume that there is a regression function  $f_0 \rightarrow \mathbb{R}^{p \times 3}$  that generates a displacement vector field  $\delta \mathbf{x}_i \in \mathbb{R}^{p \times 3}$  according to

$$\delta \mathbf{x}_i = f_0(\mathbf{x}_i) + \epsilon_i, \quad (i = 1, \dots, p). \tag{2}$$

where  $\epsilon_i$  is independent Gaussian noise,  $\epsilon_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_3)$ . The regression problem is to infer the function  $f_0$  at the input points  $\{\mathbf{x}_1, \dots, \mathbf{x}_l\}$ . The possible deformation field  $\delta \mathbf{x}_i$  is mod-

elled using a Gaussian process model  $\mathcal{GP}(\mu, k)$  that models the shape variations of a given shape family. Here  $\mu$  is the mean function of the Gaussian process, which is set to zero, as we are encoding the deformation relative to our template,  $\mathbf{X}_0$ , and we do not want any bias deformation, and  $k$  is the Gaussian process covariance function (kernel). We used the default settings in the SCALISMO implementation of the Open Framework (Gerig et al. 2017), which employs a Gaussian kernel.

The landmarks on the original template are the fixed set of input 3D points. The same landmarks on 3D data are the target positions of the fixed set of input 3D points. Given a GPMM,  $\mathcal{GP}(\mu, k)$ , that models the shape variations of a shape family, the adapted template is given as:

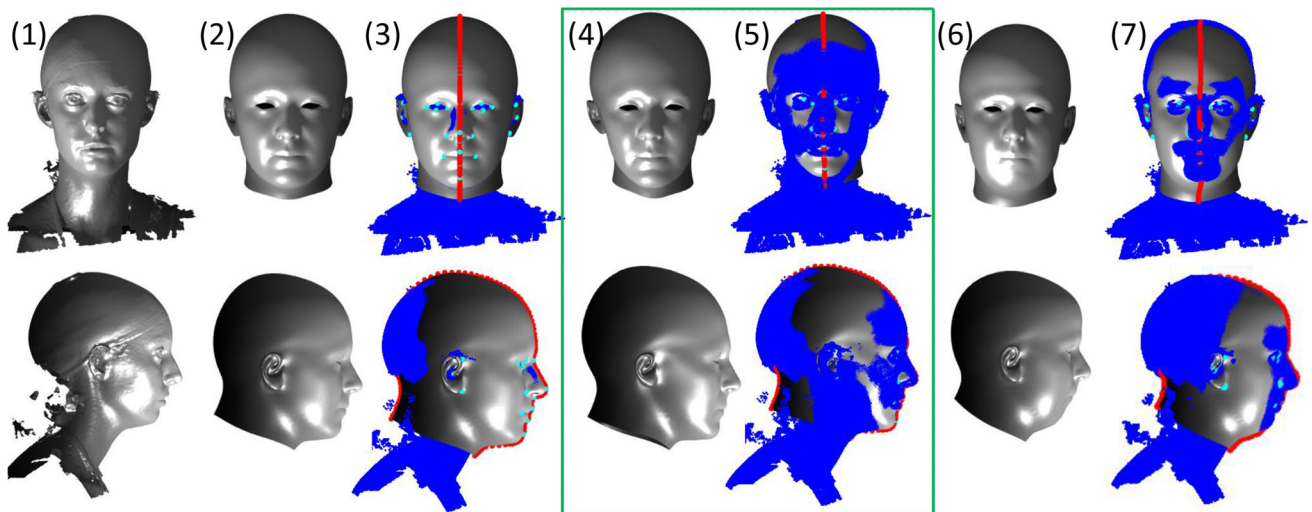
$$\mathbf{X}_1 = \mathbf{X}_0 + \Delta \mathbf{X} \tag{3}$$

where  $\Delta \mathbf{X}$  is constructed from  $\delta \mathbf{x}_i$ , ( $i = 1, \dots, p$ ). The mean template estimate by this method is shown in Fig. 10 (6) and (7).

### 6.2 Iterative Coherent Point Drift

After template alignment and shape adaptation, the task is to further deform and align the template to the target 3D data scan. Here, we employ a new shape morphing algorithm that integrates ideas from Iterative Closest Points (ICP) (Besl and McKay 1992) and Coherent Point Drift (CPD) (Myronenko and Song 2010). ICP in itself is only a rigid alignment scheme and, although CPD offers non-rigid morphing, we have found that it is often unstable when the template and data are highly imbalanced in the number of points; in particular, our data has significantly more points than our template. To counter this, we use the template to iteratively sample the data via nearest neighbours.

The pseudocode of Iterative Coherent Point Drift (ICPD) with an adaptive template is given in Algorithm 1. At each iteration,  $i$ , our deformed template is given by  $\mathbf{X}_i$ , with  $i = 0$  and  $i = 1$  used for the initial undeformed template and adapted template respectively. The first three lines are related to aligning and adapting the template, as discussed in the previous section, after which follows the main ICPD loop. Within this loop, we iterate affine and non-rigid CPD with nearest point correspondence computation preceding each stage. These *cpdAffine* and *cpdNonrigid* functions are from the original online code package of CPD. The *knnsearch(X,Y)* function is a kNN search with  $k = 1$  and finds the nearest neighbour in the data,  $\mathbf{Y}$ , for each point in the current ( $i$ th) deformation of the template  $\mathbf{X}_i$ . The function *f<sub>diff</sub>* finds the number of elements that are different between old correspondence priors  $\mathbf{id}\mathbf{x}_{old}$  and new correspondence priors  $\mathbf{id}\mathbf{x}_{new}$ . Affine CPD is fast and employed for alignment and matching of scale and aspect ratio, then the template resamples the



**Fig. 10** Template adaptation, with the top row showing a frontal view and the bottom row showing a side view. The blue overlays are the 3D scans (raw data), the cyan points are the template landmarks and the red line is the template symmetry contour. From left to right: (1) 3D scan;

(2) template with global rigid alignment; (3) 2 compared with the scan; (4) adapted template using LB mesh editing; (5) 4 compared with the scan; (6) adapted template using GP posterior model; (7) 6 compared with the scan in (7)

data and we apply non-rigid CPD to non-rigidly morph the template. This loop executes until the indices of the sampled data points do not change very much over the main loop.

**Algorithm 1** Iterative CPD (ICPD) using an adaptive template. The functions *cpdRigid*, *cpdAffine* and *cpdNonrigid* are from the public CPD code of Myronenko and Song (2010). The *adaptive* function employs Equation 1 and *knnsearch* is a *k*-NN search with  $k = 1$ .

```

1:  $[\mathbf{R}, \mathbf{t}, s] \leftarrow \text{cpdRigid}(\mathbf{S}_{Y_{fp}}\mathbf{Y}, \mathbf{S}_{X_{fp}}\mathbf{X}_0)$ 
2:  $\mathbf{X}_0 \leftarrow s\mathbf{X}_0\mathbf{R} + \text{repmat}(\mathbf{t}, [\text{size}(\mathbf{X}_0, 1), 1])$ 
3:  $\mathbf{X}_1 \leftarrow \text{adaptive}(\mathbf{X}_0, \mathbf{Y}, \mathbf{S}_{X_{fp}}, \mathbf{S}_{Y_{fp}})$ 
4:  $\text{flag} = 0$ 
5:  $\text{idx}_{new} \leftarrow \text{knnsearch}(\mathbf{Y}, \mathbf{X}_1)$ 
6: while  $\text{flag} = 0$  do
7:    $\text{idx}_{old} = \text{idx}_{new}$ 
8:    $\mathbf{X}_i \leftarrow \text{cpdAffine}(\mathbf{Y}(\text{idx}_{old}, :), \mathbf{X}_i)$ 
9:    $\text{idx} \leftarrow \text{knnsearch}(\mathbf{Y}, \mathbf{X}_i)$ 
10:   $\mathbf{X}_i \leftarrow \text{cpdNonrigid}(\mathbf{Y}(\text{idx}, :), \mathbf{X}_i)$ 
11:   $\text{idx}_{new} \leftarrow \text{knnsearch}(\mathbf{Y}, \mathbf{X}_i)$ 
12:  if  $f_{diff}(\text{idx}_{old}, \text{idx}_{new}) < N$  then
13:     $\text{flag} = 1$ 
14:  end if
15: end while

```

The qualitative output of ICPD is very smooth, a feature inherited from standard CPD. A subsequent regularized point projection process is required to capture the target shape detail, and this is described next.

### 6.3 Laplace–Beltrami Regularized Projection

When ICPD has deformed the template close to the data, point projection is required to eliminate any shape distance error in a direction normal to the data’s surface. Such a point projection process is potentially fragile. If the data is incomplete or noisy, then projecting vertices from the deformed template to their nearest vertex or surface position on the data may cause large artefacts. Again, we overcome this by treating the projection operation as a mesh editing problem with three ingredients. First, position constraints are provided by those vertices with mutual nearest neighbours between the deformed template and data. Using mutual nearest neighbours reduces sensitivity to missing data. Second, local position constraints are provided by the automatically detected landmarks on the data. Third, regularisation constraints are provided by the LB operator which acts to retain the local structure of the mesh. We call this process Laplace–Beltrami Regularized Projection (LBRP), as shown in the registration framework in Fig. 8.

In the same problem formulation as our template adaptation, we write the point projection problem as a linear system of equations. The vertices of the data scan are stored in the matrix  $\mathbf{Y} \in \mathbb{R}^{n \times 3}$  and the template, after  $i$  deformations, is stored in the matrix  $\mathbf{X}_i \in \mathbb{R}^{p \times 3}$ . We define the selection matrices  $\mathbf{S}_{X_{nn}} \in [0, 1]^{m \times p}$  and  $\mathbf{S}_{Y_{nn}} \in [0, 1]^{m \times n}$  as those that select the  $m$  vertices with mutual nearest neighbours from deformed template and scan respectively, and these selection matrices may also select landmark-based cor-

respondences. This linear system can be written as:

$$\begin{pmatrix} \lambda \mathbf{L}_i \\ \mathbf{S}_{X_{nn}} \end{pmatrix} \mathbf{X}_{i+1} = \begin{pmatrix} \lambda \mathbf{L}_i \mathbf{X}_i \\ \mathbf{S}_{Y_{nn}} \mathbf{Y} \end{pmatrix} \quad (4)$$

where  $\mathbf{L}_i \in \mathbb{R}^{p \times p}$  is the cotangent Laplacian approximation to the LB operator based on  $\mathbf{X}_i$  and  $\mathbf{X}_{i+1} \in \mathbb{R}^{p \times 3}$  are the projected vertex positions that we wish to solve for to obtain the final template deformation. The parameter  $\lambda$  weights the relative influence of the position and regularisation constraints, effectively determining the ‘stiffness’ of the projection. As  $\lambda \rightarrow 0$ , the projection tends towards nearest neighbour projection. As  $\lambda \rightarrow \infty$ , the deformed template will only be allowed to rigidly transform.

### 7 Alignment and Statistical Modeling

We use Generalized Procrustes Analysis (GPA) to align our deformed templates before applying statistical modelling using PCA. This generates a 3DMM as a linear basis of shapes, allowing for the generation of novel shape instances. We may use all of the full head template vertices for this modelling, or any subset. For example, later we select the cranial vertices when we build a specialized 3DMM to analyse a cranial medical condition.

In many applications, vertex resolution is not uniform across the mesh. For example, we may use more vertices to express detail around facial features of high curvature. However, standard PCA attributes the same weight to all points in its covariance analysis, biasing the capture of shape variance to those regions of high resolution. Whether or not this is desirable is application dependent. Here, to normalize against the effect of varying surface-sampling resolution, we employ Weighted PCA (WPCA) in our statistical modelling.

#### 7.1 3DMM Training and Fitting Using Weighted PCA

To train a 3DMM using PCA, we need to represent the collection of deformed, GPA-aligned templates,  $\mathbf{X}_j \in \mathbb{R}^{p \times 3}$ ,  $j \in \{1 \dots N\}$ , in shape space. This is achieved using the reshaping operation

$$\mathbf{x}_j = \text{vec}(\mathbf{X}_j) = [x_1, y_1, z_1, \dots, z_p]^T \in \mathbb{R}^{3p}. \quad (5)$$

Each vector,  $\mathbf{x}_j^T$ , is stacked into the  $j$ -th row of a data matrix, and the column-wise mean is subtracted to give the mean-centered data matrix  $\mathbf{X}_D \in \mathbb{R}^{N \times 3p}$  for the  $N = 1212$  Headspace subjects and  $p = 11510$  template vertices. We also note that the mean shape in shape space is given as  $\bar{\mathbf{x}} = \text{vec}(\bar{\mathbf{X}})$ .

Standard PCA performs an eigendecomposition of the covariance matrix associated with the set of training exam-

ples,  $\Sigma = \mathbf{X}_D^T \mathbf{X}_D$ . In our case, we have a small number of training data observations ( $N$ ) compared with the number of features, or dimensions ( $3p$ ), hence we would need to apply SVD to  $\mathbf{X}_D$ , as  $\Sigma$  is not full rank. However, a more efficient alternative is to employ snapshot PCA, which computes the eigenvectors of the Gram matrix  $\mathbf{G} = \mathbf{X}_D \mathbf{X}_D^T$ . This is significantly smaller than the covariance matrix and the desired covariance eigenvectors can be computed by premultiplying the Gram eigenvectors by  $\mathbf{X}_D^T$ . That is, the desired principal components are linear combination of original zero-mean data using weights that are the eigenvectors of the Gram matrix.

Additionally, to compute a Weighted PCA (WPCA), we define a sparse symmetric mass matrix,  $\mathbf{M}$ , where the diagonal elements are based on the means of all the faces connected to that vertex (8 for non mesh boundary vertices), and the symmetric off-diagonal elements ( $i, j$ ) and ( $j, i$ ) are non-zero where there is a mesh edge. These elements are based on the means of the two faces connected to that edge. We compute the weighted Gram matrix

$$\mathbf{G}_M = \mathbf{X}_D \mathbf{M} \mathbf{X}_D^T \quad (6)$$

and its eigendecomposition,  $[\mathbf{U}, \Lambda]$ , representing  $N - 1$  eigenvectors and eigenvalues respectively. The required 3DMM eigenvectors (principal components) are then given as  $\mathbf{V} = \mathbf{X}_D^T \mathbf{U}$ .

Given the mean vectorial template shape  $\bar{\mathbf{x}} \in \mathbb{R}^{3p}$ , the basis matrix  $\mathbf{V} \in \mathbb{R}^{3p \times k}$  for  $k < N$  principal components, and some morphed template to be represented by the model,  $\mathbf{x} \in \mathbb{R}^{3p}$ , we least-squares align the morphed template to the mean template shape and compute the shape parameters  $\alpha \in \mathbb{R}^k$  as:

$$\alpha = \Lambda^{-\frac{1}{2}} \mathbf{V}^T (\mathbf{x} - \bar{\mathbf{x}}) \quad (7)$$

Using the matrix,  $\Lambda^{-\frac{1}{2}} \in \mathbb{R}^{k \times k}$ , allows the shape parameter vector to be interpreted as the number of standard deviations along each eigenvector. The 3DMM reconstruction,  $\mathbf{x}$  is then given as:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{V} \Lambda^{\frac{1}{2}} \alpha \quad (8)$$

which is then reshaped into the form  $\mathbf{X} \in \mathbb{R}^{p \times 3}$ .

#### 7.2 Flexibility Modes of Partial Data Reconstruction

In the cases of incomplete scan data, the morphed template will also be incomplete. This is straightforward to detect in our morphing algorithm, as those vertices will not have a mutual nearest neighbour with the scan that it is warped to. Despite this, the 3DMM can both infer the missing parts and

estimate the variation in shape of these missing parts when the variation in the partial shape is minimized.

Given partial data, we can divide the shape components into two parts: one for the missing data,  $\mathbf{x}_a$ , and the other for (partial) present data,  $\mathbf{x}_b$ . Without loss of generality, this is achieved by permutation of blocks of 3 variables (one block per mesh vertex) such that  $\mathbf{x} = (\mathbf{x}_a^T, \mathbf{x}_b^T)^T$  and  $\mathbf{V}_{a,b}$  are the associated partitioned eigenvectors.

For reconstruction of missing data, we first compute shape parameters from the partial template as:

$$\boldsymbol{\alpha} = \Lambda^{-\frac{1}{2}} (\mathbf{V}_b^T \mathbf{V}_b)^{-1} \mathbf{V}_b^T (\mathbf{x}_b - \bar{\mathbf{x}}_b) \quad (9)$$

The pseudoinverse of  $\mathbf{V}_b$  is employed, as this matrix is not orthogonal after partitioning it from the eigenvector matrix for the full shape. We then reconstruct the missing part using this shape parameter vector to give:

$$\mathbf{x}_a = \bar{\mathbf{x}}_a + \mathbf{V}_a (\mathbf{V}_b^T \mathbf{V}_b)^{-1} \mathbf{V}_b^T (\mathbf{x}_b - \bar{\mathbf{x}}_b). \quad (10)$$

It is possible to compute the variability of the reconstructed missing part,  $\mathbf{x}_a$ , with the present part,  $\mathbf{x}_b$ , entirely fixed, only if the covariance matrix of that present part is invertible, which in turn requires the number of model training examples to be greater than the number of variables associated with the present part, which is often not the case.

To circumvent this problem, Albrecht et al. (2008) allow a small amount of variance in the present part in order to explore the remaining flexibility in the missing part. This can be formulated as a generalized eigenvalue problem, the solution of which yields a set of generalized eigenvectors that describe the variation in the overall shape and these are called flexibility modes.

We define scaled, partitioned principal components as follows:

$$\mathbf{Q}_{a,b} = \mathbf{V}_{a,b} \Lambda_{a,b}^{\frac{1}{2}} \quad (11)$$

Then, for some  $k$ -dimensional shape parameter vector  $\boldsymbol{\alpha} \in \mathbb{R}^k$  (we use  $k = 100$ ), the deformations in the ‘variable’ (i.e. missing) and ‘near-fixed’ (i.e. present) parts of the template are given by  $\mathbf{Q}_a \boldsymbol{\alpha}$  and  $\mathbf{Q}_b \boldsymbol{\alpha}$  respectively. To maximize the change in shape of the variable shape parts, Albrecht et al. (2008) formulate the constrained maximization problem:

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^k} (\boldsymbol{\alpha}^T \mathbf{Q}_a^T \mathbf{Q}_a \boldsymbol{\alpha}), \text{ st. } \boldsymbol{\alpha}^T \mathbf{Q}_b^T \mathbf{Q}_b \boldsymbol{\alpha} = c \quad (12)$$

where  $c$  quantifies the amount of change allowed in the fixed shape components. Using a Lagrangian multiplier,  $\mu$ , and differentiating w.r.t.  $\boldsymbol{\alpha}$  transforms the constrained maximization to the generalized eigenvalue problem:

$$\mathbf{Q}_a^T \mathbf{Q}_a \boldsymbol{\alpha} = \mu \mathbf{Q}_b^T \mathbf{Q}_b \boldsymbol{\alpha} \quad (13)$$

The solution then yields a set of  $k$  generalized eigenvectors  $\{\boldsymbol{\alpha}_1 \dots \boldsymbol{\alpha}_k\}$ , called flexibility modes (Albrecht et al. 2008), ordered by their generalized eigenvalues,  $\{\mu_1, \dots, \mu_k\}$ , from high to low. The generalized eigenvectors are scaled by  $\sqrt{l}$  which ensures that the present, ‘near-fixed’ shape part has a squared Euclidean norm of unity, i.e.  $\|\mathbf{Q}_b^T \boldsymbol{\alpha}_i\|^2 = 1$ . The ‘variable’ part is then scaled by  $\mu_i$  in terms of its squared Euclidean norm.

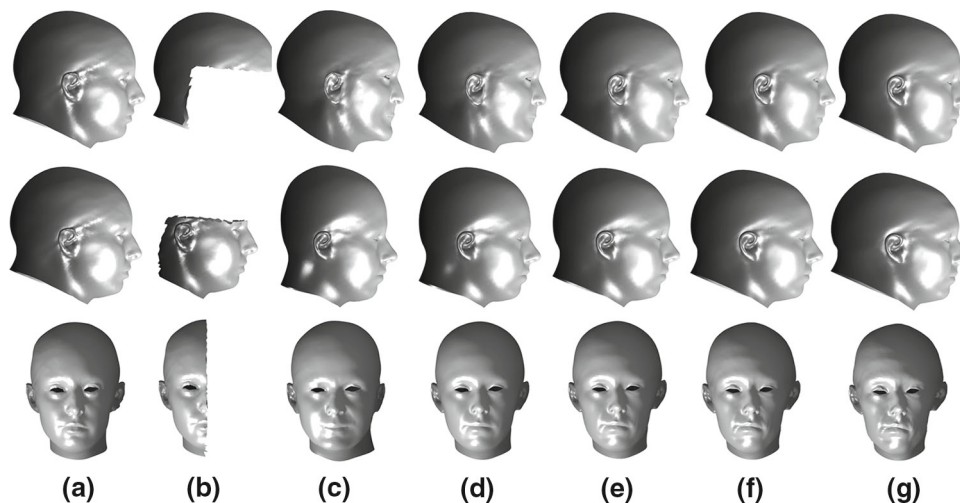
In the following, we explore flexibility modes associated with our global 3DMM. We choose scale factors ( $\pm 2.2\sqrt{l}$ ) to illustrate the flexibility modes of: i. a missing face, ii. a missing cranium, and iii. a missing half head, at one side of the sagittal symmetry plane. In the first row of Fig. 11, we largely fix the shape of the cranium and reconstruct the full head from that shape, while permitting the shape of the face to vary. Showing the remaining flexibility, when one shape part is highly constrained provides more insight into the statistical properties of the shape. Here we found that most variation occurs over the chin region, which may have a wide range of forms (shapes and sizes) for a given cranium. Perhaps this is unsurprising, as the jaw is distant from the cranium and is a separate bone. However, to our knowledge, this is the first time that flexible reconstruction has been performed using a 3DMM of the head.

The second row of Fig. 11 is the reconstruction from the face only, and we note that the principal variation in reconstructed cranium is from high/short to low/long. This offers a way to augment existing face models with a cranium. Finally, in the third row of Fig. 11, reconstruction from one side of the sagittal symmetry plane demonstrates asymmetrical variation of the head. This application can aid shape-from-shading using 2D profile images to predict the 3D shape of the self-occluded half of the face.

## 8 High Resolution Texture Mapping

It is preferable to store texture information in a UV space texture map, where resolution is unconstrained, rather than store only per-vertex colors, where resolution is limited by mesh resolution. To do so requires the texture information from each data scan to be transformed into a standard UV texture space for which the embedding of the template is known. As we use the FaceWarehouse mesh, we also use their vertex embedding into UV space (Cao et al. 2014). The key to obtaining a high quality texture map is embedding all the pixels in one triangular mesh face from the texture image to its corresponding mesh face in the UV template (see Fig. 12 (1)). Compared to a per-vertex color-texture map, a pixel embedding texture map employs all the pixels in each template mesh face, thus capturing more texture detail.

After template morphing, the deformed template has the same number of points joined into a triangulation that is



**Fig. 11** The flexibility of reconstruction from incomplete data using the global 3DMM built from 1212 Headspace subjects, and using 100 model parameters. The three rows show three different data crops—top: remove face; centre: remove cranium; bottom: remove half head. The first two columns show: **a** the template fitted to a scan; **b** the manually

cropped template. The rightmost five columns show reconstructions. **e** The 3DMM-based reconstruction of the missing data. **c, d** The first flexibility component, scaled by  $\pm 2.2\sqrt{I}$ . **f, g** The second flexibility component, scaled by  $\pm 2.2\sqrt{I}$

shared across all scans. Thus in UV coordinates, UV faces of the morphed template are shared with the template. Given the morphed vertex positions  $\mathbf{X} \in \mathbb{R}^{p \times 3}$  from the template morphing stage, we first compute the UV coordinates for each point of the morphed template in original texture image:

$$[\mathbf{u}, \mathbf{v}] = g(\mathbf{S}_{Ynn}\mathbf{Y}) \tag{14}$$

where  $\mathbf{S}_{Ynn} \in \mathbb{R}^{p \times n}$  is the selection matrix that selects the  $p$  vertices with nearest neighbours from the morphed template  $\mathbf{X}$  to the data scan  $\mathbf{Y}$  and  $g$  is the UV coordinates mapping from the data mesh to the texture image, which is implicit in the 3D scan OBJ file. The UV coordinates mapping from texture image to 3D scan mesh is a surjection but not an injection. Thus vertices and faces from the 3D scan mesh may have several sets of UV coordinates in the texture image, depending on the number of capture viewpoints. This creates a problem when the three vertices from a mesh face do not lie in the same viewpoint within our 5-view composite texture image. To overcome this problem, we have a two-stage approach. First, we note that when the three sets of UV coordinates of a mesh face straddle more than one viewpoint, then the mesh face area will be large. Therefore, our first stage is to select the combination of UV coordinates belonging to each mesh face, that minimizes the mesh face area:

$$i^* = \underset{i}{\operatorname{argmin}} h([\mathbf{u}_{i1}, \mathbf{v}_{i1}], [\mathbf{u}_{i2}, \mathbf{v}_{i2}], [\mathbf{u}_{i3}, \mathbf{v}_{i3}]) \tag{15}$$

where  $h$  is the mesh face area computation and its arguments are the three sets of UV coordinates defining the texture coordinates of the mesh face, and the variable  $i$  covers all sets of texture coordinates for a given mesh face.

Minimization of face area does not guarantee that all UV coordinate combinations belong to the same viewpoint (see Fig. 12-2). To overcome this, a second stage that employs affine transformations is used to refine the UV coordinates.

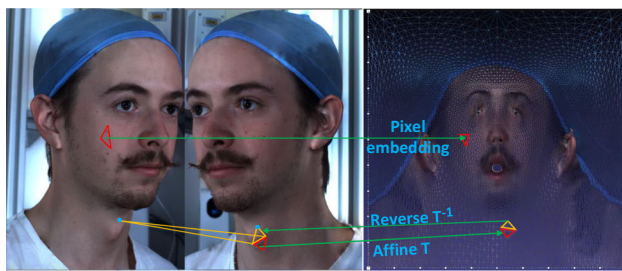
If the UV coordinates in one mesh face are placed in different views, we compute the affine transformation  $\mathbf{T}$  from its adjacent mesh face (the one with a common edge in the same viewpoint) to the corresponding face in the template UV faces. Then this corresponding face is transformed by  $\mathbf{T}^{-1}$  to find the vertex position in the common viewpoint (see blue point in Fig. 12 (1)). The outcome of affine transformation refinement is shown in Fig. 12 (3). As shown in Fig. 13, the quality of the texture map improves compared to the per-vertex approach, such that the freckles can be seen in the rendering.

In the following two sections, we evaluate the quality of our shape correspondences and then the quality of our trained shape models.

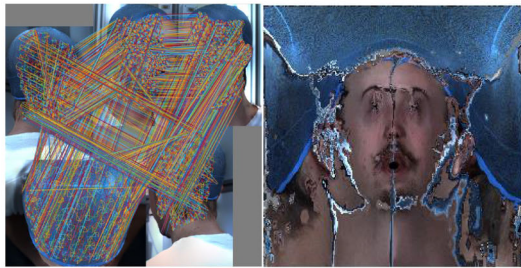
!t

### 9 Evaluation of Correspondences

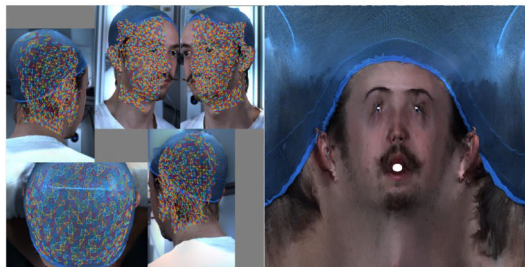
We evaluated the proposed template morphing algorithms using both the 100 subjects with neutral expression from the BU3D dataset (Yin et al. 2008) and 1212 subjects from the Headspace dataset (Duncan et al. 2018). Section 9.1 presents the validation of the template adaptation



(1) Pixel embedding and affine transformation refinement



(2) Before



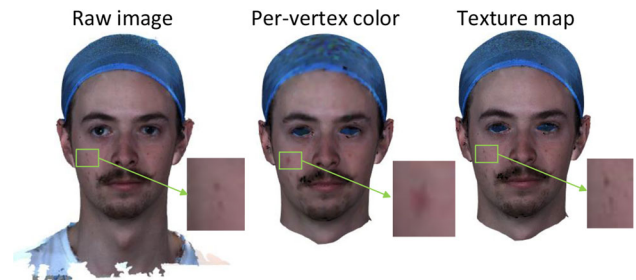
(3) After

**Fig. 12** Texture mapping process: (1) Pixel embedding from the raw image (left) to the UV embedding (right), (2) UV coordinates of mesh faces before affine refinement—here we are unable to use mesh faces that straddle viewpoints. Left: the lines join vertices that belong to the same mesh face but straddle multiple viewpoints. Right: the seams are due to the loss of the mesh faces that straddle multiple viewpoints. (3) Left: after affine refinement, all mesh face UV coordinates lie within a single viewpoint. Right: all mesh faces can be used so there are no longer any seams

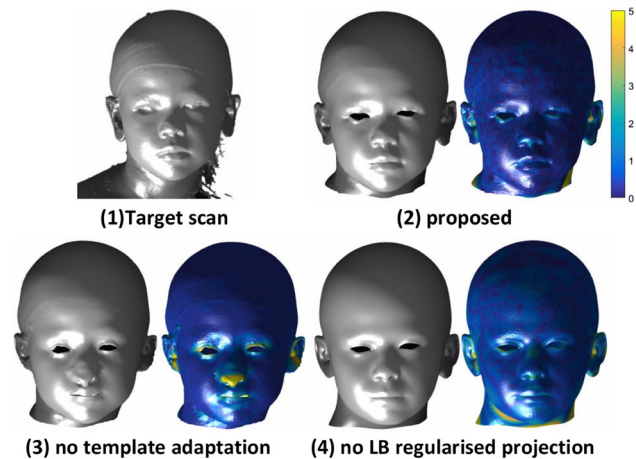
approach. In Sect. 9.2, we compare three registration methods: i. the proposed adaptive template and ICPD based approach, ii. our recent symmetric deformation algorithm (Dai et al. 2018b), and iii. our earlier hierarchical parts-based morphing method (Dai et al. 2017b). In Sect. 9.3, we present the qualitative and quantitative evaluation of the proposed correspondence establishment with competing template morphing approaches from other groups.

### 9.1 Ablation Study

In order to validate the effectiveness of each key step in the proposed registration pipeline, we first remove the process of template adaption from the pipeline and evaluate performance. We then replace this and remove LB Regularized Projection from the proposed framework and again evalu-



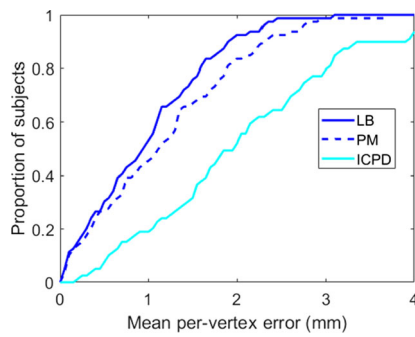
**Fig. 13** Our texture mapping process improves texture detail. The freckles in the texture map (right) match with those in the image (left), but the per-vertex color mapping (centre) cannot capture such detail



**Fig. 14** Ablation study observing the effect when pipeline components are removed: (1) scan; (2) morphed template using proposed full pipeline method; (3) morphed template when template adaptation is removed; (4) morphed template with LB regularised projection. The templates on the right employ error color maps in millimeters

ate performance qualitatively and quantitatively, comparing both modified pipeline cases with the full pipeline. Typical results for a child in the Headspace dataset are shown in Fig. 14. After pure rigid alignment without template adaptation, the nose of the template is still bigger than the target. As can be seen in Fig. 14-3, the nose and ear areas both have a poor deformation result. Without LB Regularized Projection, shown in Fig. 14-4, it fails to capture the shape detail compared with the proposed full pipeline. The adaptive template improves the correspondence accuracy in the local regions, while the LB Regularized Projection helps in decreasing the correspondence error in the surface normal direction.

Using 100 neutral-expression subjects from the BU3D dataset for quantitative validation, we compared the performance of: (i) the proposed ICPD registration, (ii) ICPD with an adaptive template using LB mesh manipulation, and (iii) ICPD with an adaptive template, using a Gaussian Process Posterior Model (PM). The mean per-vertex error is computed between the registration results and their ground-truth. The number of ICPD iterations and computation time is recorded, when using the same computation platform. The

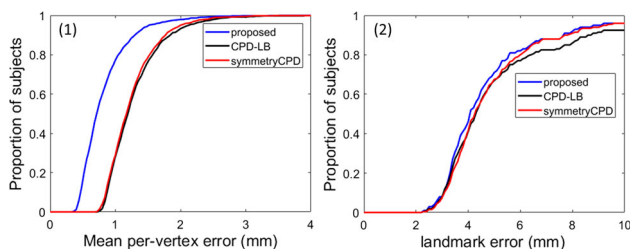


**Fig. 15** Improvement in correspondence and convergence performance when using an adaptive template. The graph shows the proportion of morphs that have their closest point mean per-vertex error less than the abscissa value. (1) ICPD without an adaptive template (cyan); (2) ICPD

**Table.** Convergence of ICPD

	ICPD	LB	PM
ICPD loops	6.47	3.52	3.74
Time (s)	831.35	426.13	434.53

with LB-based adaptive template (blue); (3) ICPD with Gaussian Process PM-based adaptive template (blue dashed). This test is conducted on 100 subjects with neutral expression from the BU3D dataset



**Fig. 16** Correspondence performance for the proposed method compared to *hierarchical parts-based* CPD-LB (Dai et al. 2017b) and symmetry-aware CPD (Dai et al. 2018a). (1) Mean per-vertex closest point error; (2) landmark error. This test is conducted on 100 neutral-expression subjects from the BU3D dataset

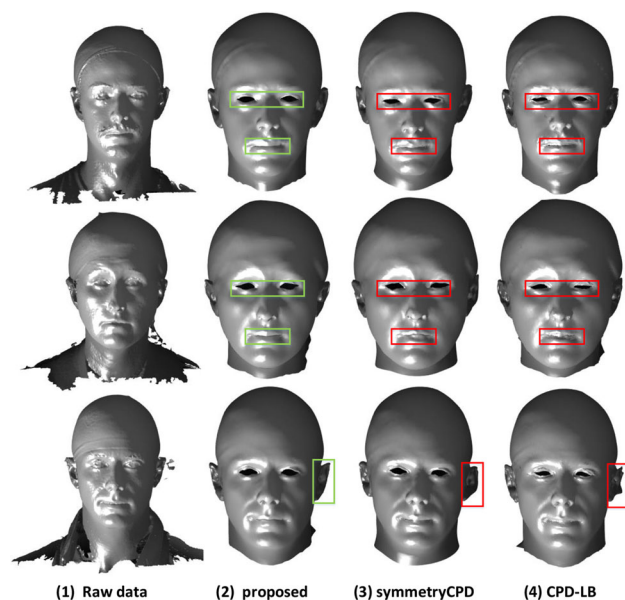
per-vertex error plot in Fig. 15 illustrates that the adaptive template improves the correspondence accuracy of ICPD. The number of ICPD iterations and computation time is significantly decreased by the adaptive template method. In particular, the adaptive template method using LB mesh manipulation has better performance than an adaptive template using a posterior model. Henceforth, we employ the adaptive template approach using LB mesh manipulation.

### 9.2 Comparison with Previous Work

This section compares the proposed method with our previous work. Figure 16 compares: (i) the proposed ICPD with adaptive template, (ii) *hierarchical parts-based* CPD-LB (Dai et al. 2017b), and (iii) symmetry-aware CPD (Dai et al. 2018a).

#### 9.2.1 Quantitative Evaluation

To evaluate the three approaches quantitatively, we use 14 manual facial landmarks over 100 subjects to measure landmark error of each template deformation method. The mean per-vertex error is calculated by the mean nearest point error

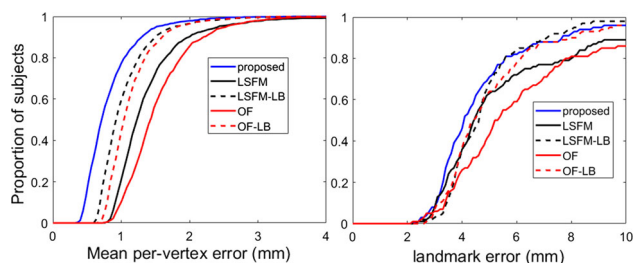


**Fig. 17** Template morphing results for three Headspace subjects (one subject per row). Column (1) the scan. The remaining three columns are the morphed templates, using: (2) the proposed method—ICPD with an adaptive template; (3) symmetry-aware CPD (Dai et al. 2018b); (4) *hierarchical parts-based* CPD-LB (Dai et al. 2017b)

between the morphed template and its corresponding 3D mesh. As can be seen from Fig. 16-1, ICPD with adaptive template is much better than the other two in terms of mean per-vertex error. Fig. 16-2 shows that ICPD with adaptive template achieves slightly better performance in the metric of landmark error.

#### 9.2.2 Qualitative Evaluation

Figure 17 illustrates that the eye region and mouth region can have a small over-fitting problem in morphing when using either *hierarchical parts-based* CPD-LB or symmetry-aware



**Fig. 18** Comparison of the proposed system with the LSFM pipeline (Booth et al. 2016) and the Open framework (OF) (Gerig et al. 2017), using 1212 subjects from the Headspace dataset. Proportion of subjects with less than the abscissa-defined Euclidean distance error. Left: landmark error. Right: per-vertex nearest point error. A higher value on the graph indicates a better performance. Additionally, we augmented both the LSFM and OF methods with LB Regularized Projection to give a more complete and fair comparison

CPD. The third row in Fig. 17 shows that, at least for this example, ICPD with adaptive template gives a better morphing in the ear region where outliers exists in the scan.

### 9.3 Comparison with Other Literature

In this section, we compare the proposed method with two publicly-available registration frameworks, namely the Large-Scale Face Model (LSFM) pipeline (Booth et al. 2016) and Open Framework (OF) (Gerig et al. 2017), using 1212 subjects from the Headspace dataset.

#### 9.3.1 Quantitative Evaluation

Figure 18-left shows cumulative histograms for shape error, where shape error is computed by measuring the nearest point distance from the deformed template to the scan. Note that 99.12% of shape errors from our method are under 2.00mm, which is better than the other two pipelines and their extensions, which appends our suggested LB Regularized Projection (LBRP) to their pipelines. We use 14 manual facial landmarks over 100 subjects to measure landmark error of each template deformation method. As shown in Fig. 18-right, 80% of landmark errors are less than 6mm for our method. Overall, the proposed method outperforms the LSFM and OF pipelines. LB Regularized Projection improves LSFM and OF in both shape error and landmark error.

#### 9.3.2 Qualitative Evaluation

Figure 19 shows a typical example where the proposed method is qualitatively superior to other methods with respect to shape detail capture, and the accuracy of the mouth region. The LSFM pipeline captures shape detail but the mouth region is not close to the 3D scan. The OF pipeline has a

smooth deformation field, thereby failing to capture shape detail. The OF approach requires a point projection stage after the shape registration to reduce shape error.

## 10 Evaluation of 3DMMs

We select 1212 individuals (606 males, 606 females) from the Headspace dataset (Duncan et al. 2018) to build our global 3DMM using our fully-automatic 3DMM training pipeline. Note that the full dataset contains 1519 subjects, but we exclude 307 subjects on the following grounds: i. Poor fitting of the latex cap; ii. Excessive hair bulge under the latex cap; iii. Excessive noise or missing parts in the 3D scan; iv. Declared craniofacial condition/trauma; v. Gender balancing. Subpopulations of these 1212 Headspace subjects are employed to build gender-specific models, LYHM-male, LYHM-female, and four age-specific models (LYHM-age-X).

This evaluation section is structured as follows. Section 10.1 is used for the visualisation of the proposed global models and sub-models on gender and age. The quantitative evaluation metrics employed are compactness, generalisation and specificity and these are presented in Sect. 10.2. The performance of our 3DMMs is compared with both the LSFM pipeline (Booth et al. 2016) and Basel Open Framework (OF) (Gerig et al. 2017). Table 1 in Sect. 10.3 shows the difference in techniques employed in these three pipelines. Section 10.4 discusses how to select the number of model components. Submodels are evaluated in Sect. 10.5 and Sect. 10.6 is used for texture map image quality assessment.

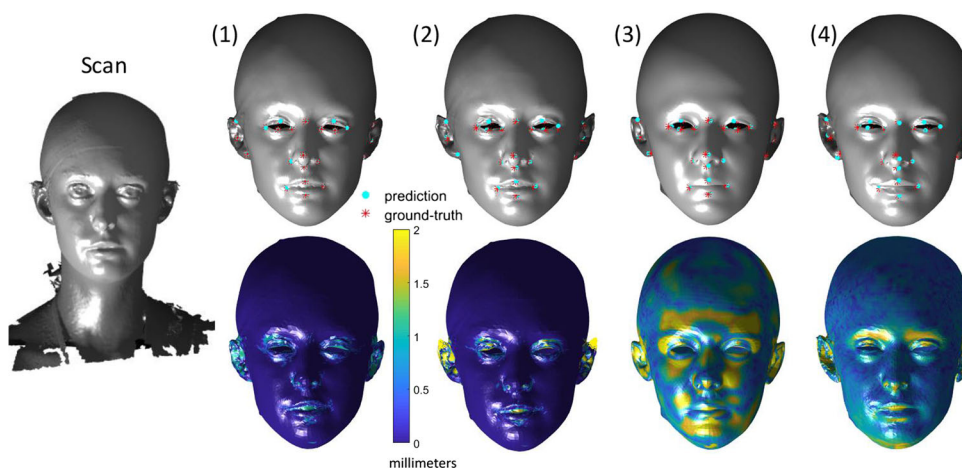
### 10.1 3DMM Visualisation

We present visualisations that provide insight into how different regions of the high-dimensional space of human face/head shape and texture are naturally related to different demographic characteristics. Taking into account the available demographic metadata in the Headspace dataset, we define the following groups: male (all ages) and female (all ages). The dataset is further clustered into four age groups: under 15 years old, 15–30 years old, 31–50 years old, and over 50 years old. The mean and most significant 7 shape components of the 6 demographic-specific models are given in Fig. 20. Likewise, Fig. 21 shows the mean and most significant 7 texture components of the six demographic-specific models visualized on the mean shape. The shape and texture is varied from  $+3\sigma$  to  $-3\sigma$ , where  $\sigma$  is standard deviation.

We apply t-Distributed Stochastic Neighbor Embedding (t-SNE) (Van Der Maaten 2014) to the shape components from all training samples to visualize the manifold of high-dimensional training shapes. By applying t-SNE between the shape parameters and its corresponding demographic data in



**Fig. 19** A comparison of correspondence performance for: (1) the proposed method, (2) the proposed method without template adaptation, (3) the OF method (Lüthi et al. 2017) and (4) the LSFM method (Booth et al. 2018). Left: the 3D scan. Top row: landmark-based correspondence results, with the morphed templates' landmarks colored cyan and the scan's landmarks colored red. Bottom row: the color map of per-vertex closest point error, in millimeters



**Table 1** Comparison of 3DMM construction pipelines

	Initialisation	Dense correspondence	Alignment	Modelling
LSFM	Automatic facial landmarks	NICP with error pruning	GPA	PCA
OF	Manual landmarks needed	GPMM registration	GPA	GP
Proposed	Automatic pose normalisation	ICPD	GPA	WPCA

our dataset, we are able to label samples in this space by their gender, see Fig. 22 (left). We observe that t-SNE has produced a nonlinear 2D embedding that clearly clusters genders, indicating that there are largely separable structural relationships in the context of male and female head shape. We do observe on the right side of the image that male and female are intermingled and these subjects may correspond to children in the dataset. Figure 22 (right) shows the visualisation of an embedding by age. Interestingly, a clear trend of increasing age can be seen across the space, suggesting that head shape also has an age-related structure.

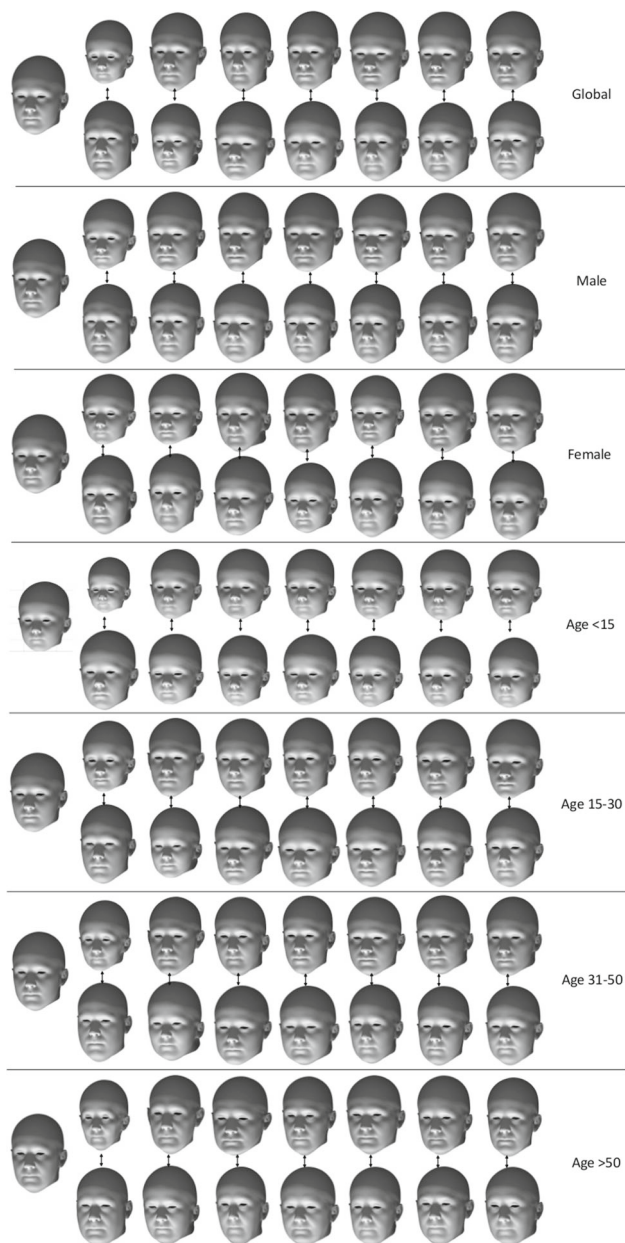
### 10.2 3DMM Quantitative Evaluation Metrics

For quantitative statistical shape model evaluation, Styner et al. (2003) propose three metrics: compactness, generalisation and specificity, as follows: i. *Compactness* of the model describes the number of model parameters (principal components for PCA-based models) required to express some fraction of the variance in the training set. Fewer parameters is better and means that the shape variation is captured more efficiently. ii. *Generalization* measures the capability of the model to represent unseen examples of the class of objects. It can be measured using the leave-one-out strategy, where one example is omitted from the training set and used for reconstruction testing. The accuracy of describing the unseen example is calculated by the mean vertex-to-vertex Euclidean distance error—lower is better for a given number of model components. With an increasing number of model parameters, the generalization error is expected to decrease. iii. *Specificity* measures the ability to gener-

ate shape instances of the class that are similar to those in the training set. In order to assess specificity, a set of shape instances should be randomly sampled from the shape space. Then the Euclidean distance error to the closest training shape is calculated for each shape instance and the average is taken for all the shape instances. The mean Euclidean distance error is expected to increase with increasing number of parameters, as the increasing number of PCA coefficients gives more flexibility to shape reconstruction. It also increases the likelihood of the reconstructed shape instances being away from the real data. For specificity measurement, the lower Euclidean distance error, the closer the model is to the training data data, so the specificity is better.

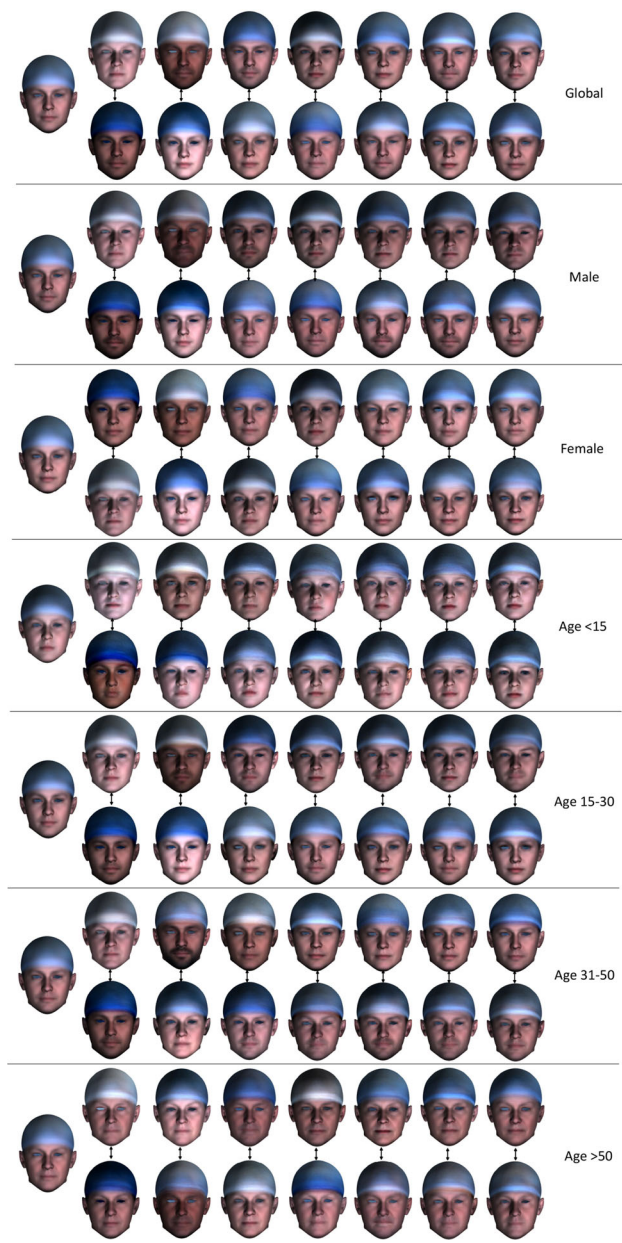
### 10.3 Evaluation of Full Head 3DMMs Using 3DMM Training Pipelines in the Literature

We build full head 3DMMs using the proposed method, the LSFM pipeline (Booth et al. 2016), and the OF pipeline (Gerig et al. 2017), again with 1212 subjects from the Headspace dataset. As can be from Fig. 23-a, when less than 33 components used, LSFM is more compact than the proposed method and OF. Between the first 33 and 79 components, the model constructed by OF is more compact than the other two. When more than 79 components used, the proposed method has better compactness than LSFM (Booth et al. 2016) and OF (Gerig et al. 2017). With the first 56 and the first 146 components used, the 3DMM constructed by the proposed method retains 95% and 98% of the shape variation in the training set.



**Fig. 20** Global shape model and shape sub-models for age and gender demographics: 1st block—global model; 2nd–3rd block—gender-specific models (male and female); 4th–7th block—four age-specific models (< 15, 15–30, 31–50, > 50). In each block, the mean and the seven most significant shape components are shown: + 3SDs (top row of each block) and – 3SDs (bottom row of each block)

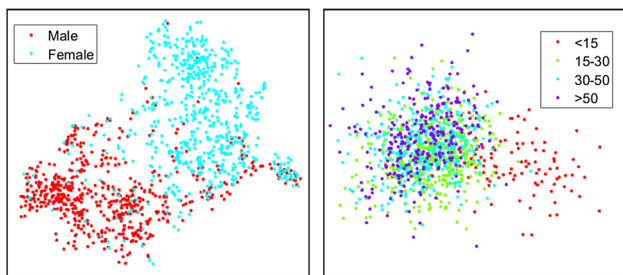
With fewer than 48 model parameters, LSFM is better than the proposed method in terms of generalisation error. With more than 48 components, the proposed method has the lowest generalisation error which, in this case, implies that proposed method has the best performance in describing unseen examples (Fig. 24).



**Fig. 21** Global texture model and texture sub-models for age and gender demographics: 1st block—global model; 2nd–3rd block—gender-specific models (male and female); 4th–7th block—four age-specific models (< 15, 15–30, 31–50, > 50). In each block, the mean and seven most significant texture components are shown: + 3SDs (top row of each block) and – 3SDs (bottom row of each block)

The proposed method has the lowest specificity error, which implies that the proposed method is best at generating instances close to real data.

Overall, with more 79 components used, the proposed pipeline is better than LSFM Booth et al. (2016) and OF Gerig et al. (2017) in terms of compactness. The generalisation error of LSFM decreases faster than the proposed method. However, with more components used, the proposed



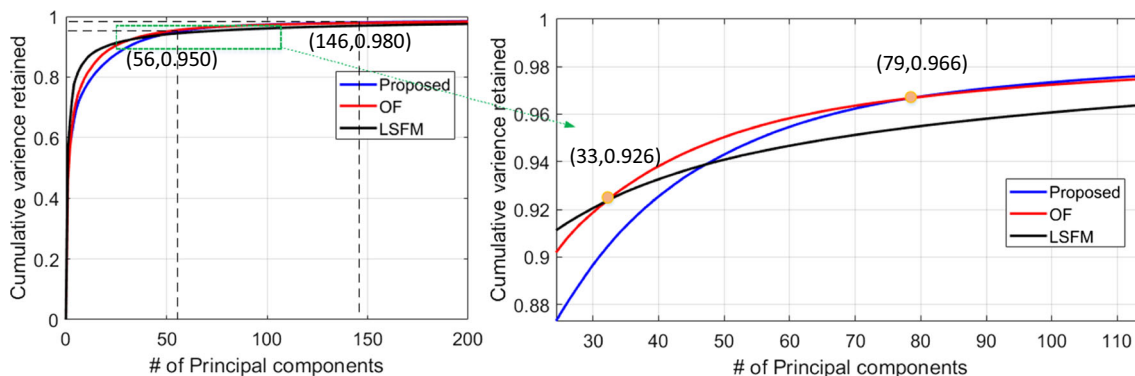
**Fig. 22** Embedding of 1212 Headspace per-subject shape vectors into a 2D space, using gender labels (left, 606 males and 606 females) and age labels (right). We use t-Distributed Stochastic Neighbor Embedding (Van Der Maaten 2014)

method has the lowest generalisation error when compared with the other two pipelines. The proposed method outperforms LSFM and OF in specificity.

### 10.4 Number of Model Components

Compactness, generalisation error and specificity are presented in Fig. 25 for the proposed global model constructed with the proposed training pipeline. It is desirable to have

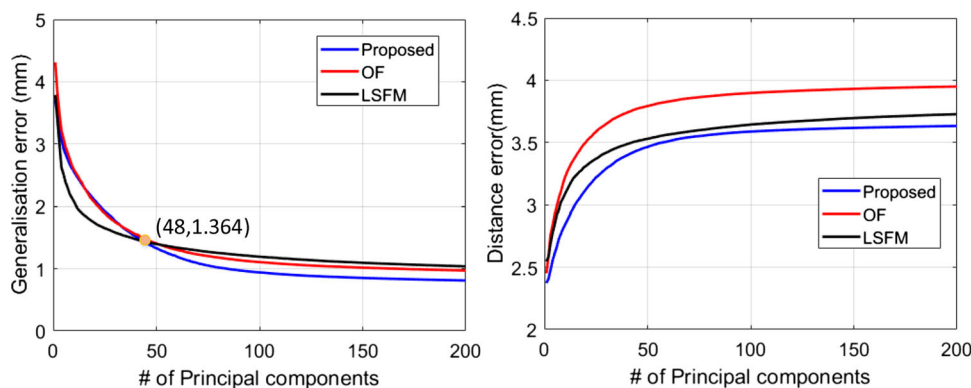
both a lower generalisation and specificity error for a 3DMM, but the figure indicates that they move in opposite directions with an increasing number of model components and therefore there is a clear trade off between the two. The compactness is also important to ensure that most of shape variation is captured by the model using a relatively small number of model parameters. So how many components should be used for a 3DMM? One clear observation that can be made from Fig. 25 is that there are diminishing returns in compactness and generalisation, and a diminishing penalty in specificity, as the number of model components increases. This ‘flattening’ is very soft and mostly occurs between 40 and 100 components. As discussed in the evaluation results, with more than 79 components used, the model constructed by the proposed method has the best performance when compared with LSFM and OF across all three metrics. With more components used, more noisy shape variation is introduced and more computation is required to fit our models. Taking all these into consideration, we employ the first 100 model components for the applications that we describe later, and our LYHM public release has this number of model components.

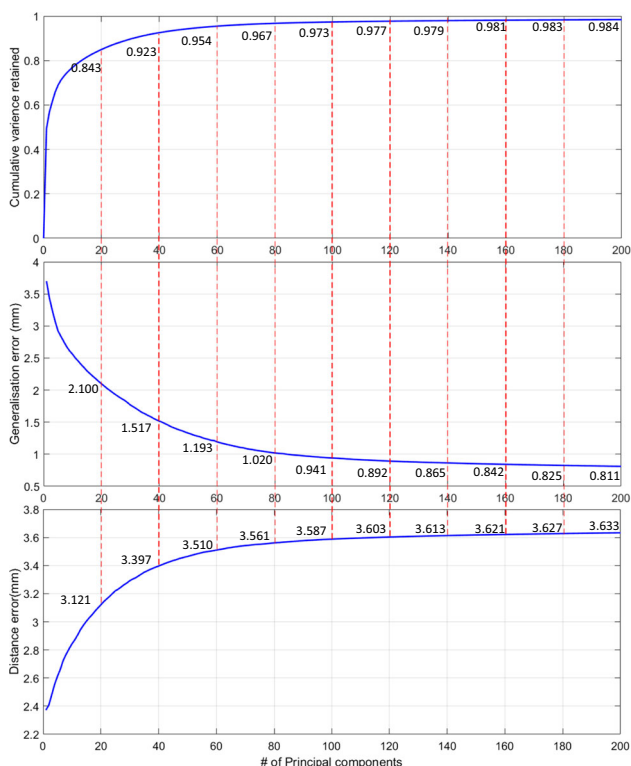


**Fig. 23** Compactness as a function of the number of principal components for a global 3D head model built with: (1) the proposed system, (2) the Open Framework (OF) (Gerig et al. 2017) and (3) the LSFM pipeline (Booth et al. 2016). A higher model compactness is better. Right: a zoom

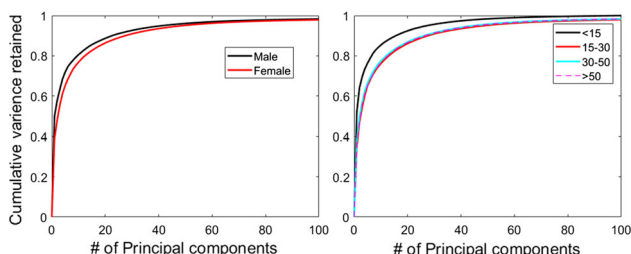
in of the graph on the left, showing the performance cross-over points in terms of number of model components. Each 3DMM employed the same 1212 Headspace subjects

**Fig. 24** Generalisation error (left) and specificity error (right) as a function of the number of principal components for a 3D head model built with: (1) the proposed system, (2) the Open Framework (OF) (Gerig et al. 2017) and (3) the LSFM pipeline (Booth et al. 2016). For both error metrics, a lower error is better. Each 3DMM employed the same 1212 Headspace subjects

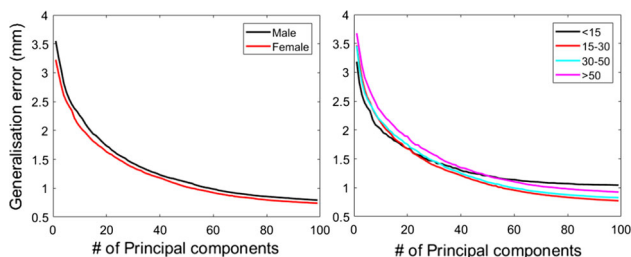




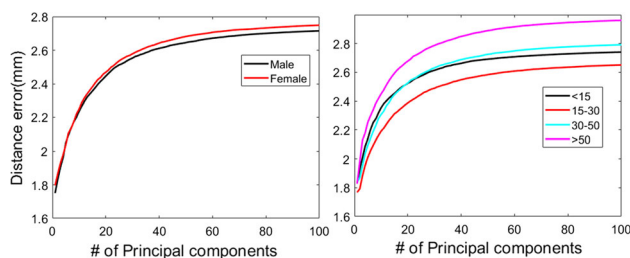
**Fig. 25** Performance metrics of the proposed global head model in term of the number of principal components of the model. Compactness (top), Generalisation error (centre), Specificity (bottom). This set of graphs facilitates the selection of the number of model components for performance metric trade off. The 3DMM employed 1212 Headspace subjects



**Fig. 26** Compactness of demographic sub-models. Left: gender (606 males, 606 females). Right: age



**Fig. 27** Generalisation of demographic sub-models. Left: gender (606 males, 606 females). Right: age



**Fig. 28** Specificity of demographic sub-models. Left: gender (606 males, 606 females). Right: age

**Table 2** Texture map image quality assessment using three metrics

	SSIM	MS-SSIM	IW-SSIM
Per-vertex color	0.8790	0.8618	0.6238
Texture mapping	<b>0.8926</b>	<b>0.8712</b>	<b>0.6505</b>

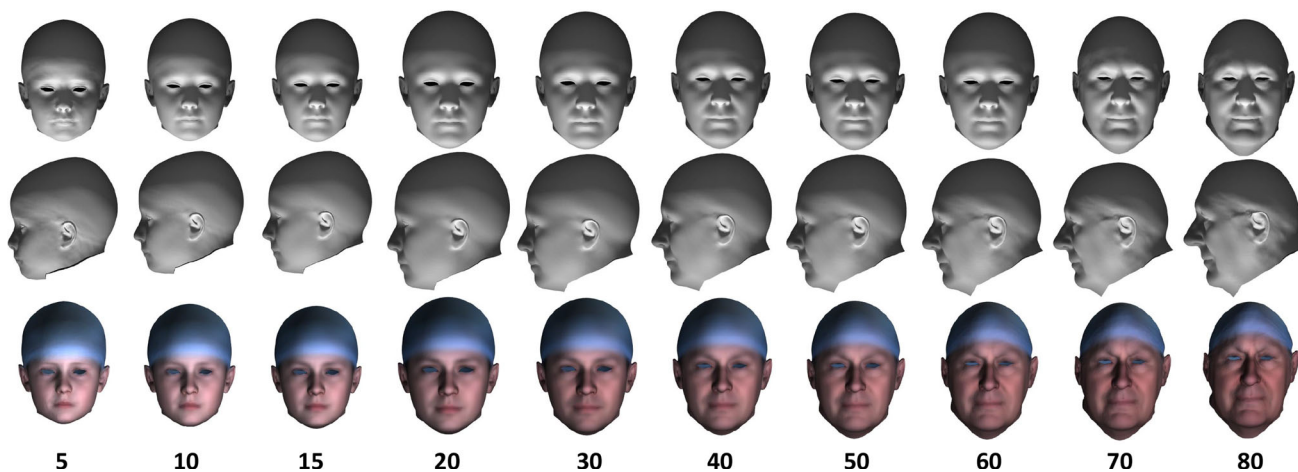
Bold values indicate the best performance over the two systems tested

### 10.5 Quantitative Evaluation of Submodels

According to Headspace demographic data, we build sub-models based on gender and age. As can be seen from Fig. 26, LYHM-male and LYHM (<15) have better compactness than other demographic specific models. In Fig. 27, LYHM-female and LYHM (15–30) have the lower generalisation error, which implies that LYHM-female and LYHM (15–30) have better performance in describing unseen examples. Specificity measures how well a model is able to generate instances that are similar to real data. Figure 28 shows 3DMM specificity. LYHM-male and LYHM (15–30) both have low distance error for specificity, which implies that LYHM-male and LYHM (15–30) are the best submodels for generating instances close to real data.

### 10.6 Texture Map Image Quality Assessment

As shown in Fig. 13, the proposed texture map technique outperforms per-vertex texture image qualitatively. We use several performance metrics, namely: SSIM (Wang et al. 2004), MS-SSIM (Wang et al. 2003) and IW-SSIM (Wang and Li 2011) to measure the texture map quantitatively. Under the assumption that human visual perception is highly adapted for extracting structural information from a scene, Structural SIMilarity (SSIM) is based on the degradation of structural information (higher is better). We save the rendering of the captured image, the proposed texture map and the per-vertex texture image into the same image size. When using the full-reference image quality assessment indices, we treat the rendering of the captured image as the reference image for image quality assessment. The rendering of the proposed texture map and per-vertex texture image are compared with this reference image, respectively. As can be seen



**Fig. 29** Age-based shape and texture regression, using 606 male subjects from the Headspace dataset. Subject age is shown from 5 to 80 years. Top row: frontal view of shape. Centre row: side view of shape.

Bottom row: frontal view of shape with texture rendered. A standard linear regression predicts shape and texture parameters of the LYHM-male submodel using an age variable

in Table 2, the proposed texture mapping technique improves the texture image quality for texture modelling, when compared with that of per-vertex color texture image.

### 11 Applications

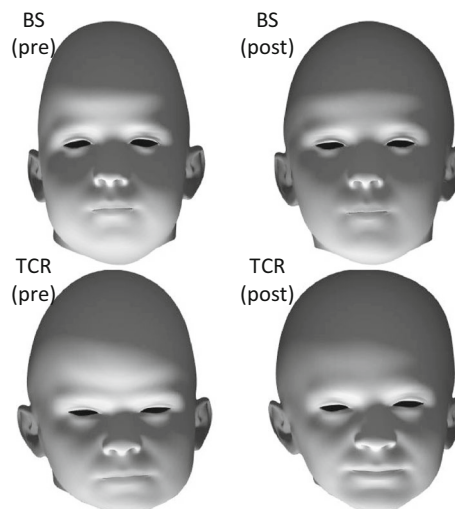
We now demonstrate two applications of our 3DMMs: age regression in Sect. 11.1, clinical intervention outcome evaluation in Sect. 11.2.

#### 11.1 Age Regression

Model regression can validate the model’s generalisation ability on some property (e.g. age, gender). We use the 606 male Headspace subjects, used to build the LYHM-male submodel, in a linear regression from age to the 3DMM shape parameters, and from age to the 3DMM texture parameters. Age is supplied as subject metadata within the Headspace dataset and standard linear regression is employed. After this learning process, the shape/texture parameters can be regressed, when given some age value. Figure 29 demonstrates a sequence of 3D images generated from age regression of the 3DMM, ranging from 5 to 80 years. Craniofacial development is as expected, in that the cranium develops before the face in children, whereas the shape of the cranium is does not appear to change significantly in adults.

#### 11.2 Clinical Intervention Outcome Evaluation

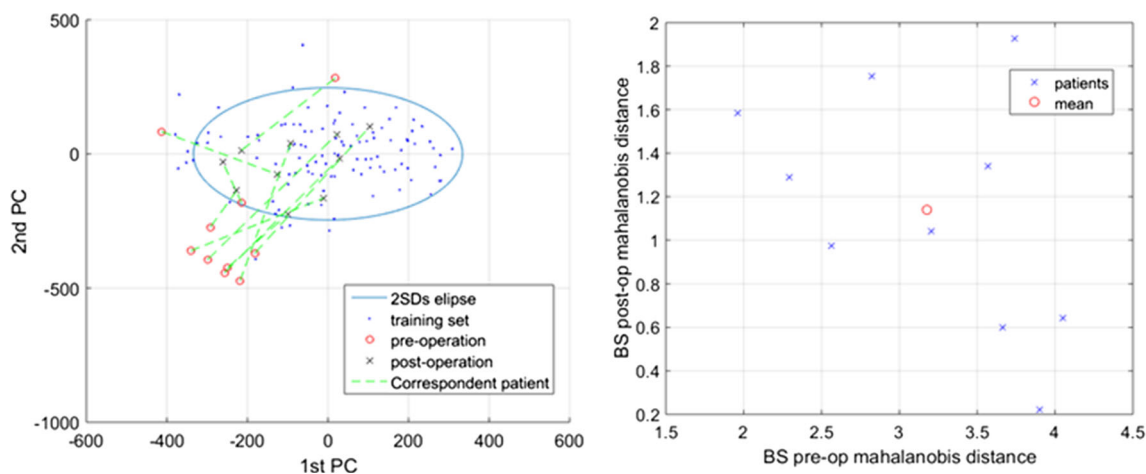
We use our modeling to describe post-surgical cranial change in a sample of 17 craniosynostosis patients (children), 10 of



**Fig. 30** Template registration of two patients undergoing two different surgical interventions. Top row: patient A undergoing Barrel Staving (BS) intervention. Bottom row: patient B undergoing Total Calvarial Reconstruction (TCR) intervention. Left column: shows pre-operative shape. Right column: shows post-operative shape

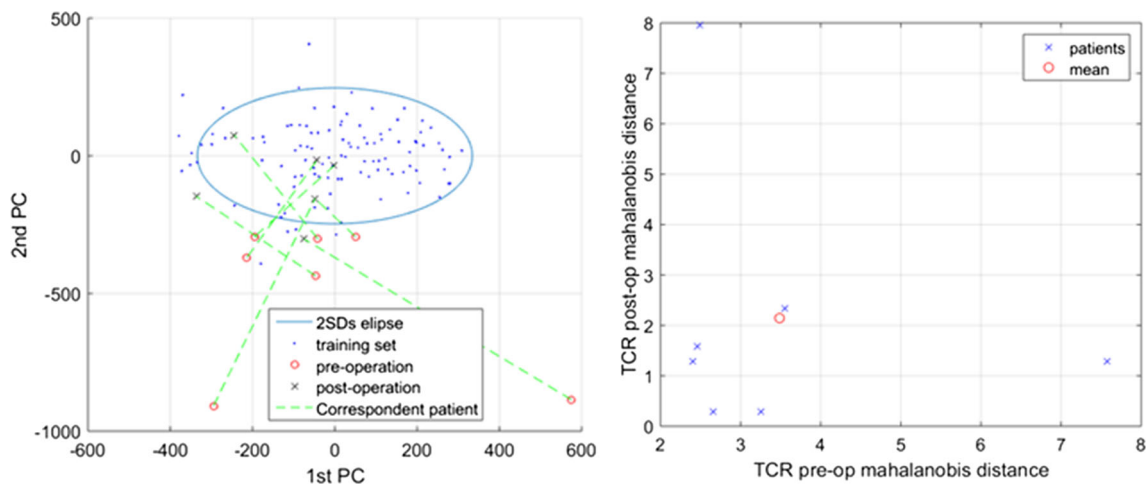
which have undergone one type of cranial corrective procedure *Barrel Staving* (BS) and the other 7, another cranial corrective procedure *Total Calvarial Reconstruction* (TCR).

We build a scale-normalized cranial model with the face removed to focus on cranial shape variation only. The model is constructed using Headspace subjects under 15 years and we note that major cranial shape changes are not thought to occur after 2 years. Thus the model is applicable to all but very young children. Note that we are merely illustrating how our 3DMMs can evaluate surgical procedures. However, in



**Fig. 31** Barrel Staving (BS) intervention outcomes in terms of 3D cranial shape. Left: the first two principal components of the patient’s cranial shape compared to a model training population (children under 15 in Headspace). The blue ellipse shows the 2SD boundary of this population. Preoperative shapes are shown by red circles and postop-

erative shapes are black crosses, with the green dashed line indicating corresponding patients. Right: postoperative shape plotted against preoperative shape in terms of Mahalanobis distance from the general population mean, using the first two principal components of the 3D cranial model. All patients move closer to the mean



**Fig. 32** Total Calvarial Reconstruction (TCR) intervention outcomes in terms of 3D cranial shape. Left: the first two principal components of patient’s cranial shape compared to a model training population (children under 15 in Headspace). The blue ellipse shows the 2SD boundary of this population. Preoperative shapes are shown by red circles and

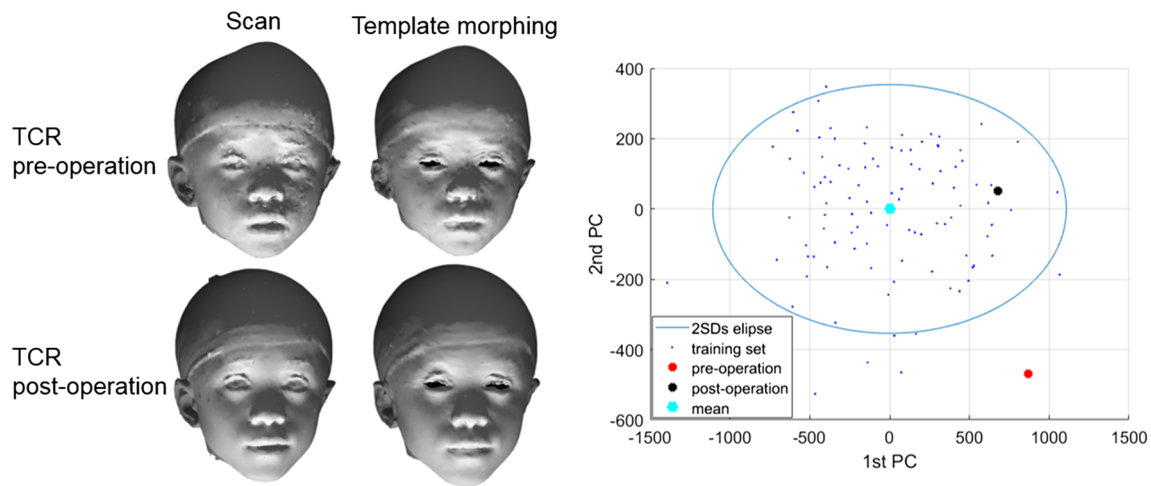
postoperative shapes are black crosses, with the green dashed line indicating corresponding patients. Right: postoperative shape plotted against preoperative shape in terms of Mahalanobis distance from the general population mean, using the first two principal components of the 3D cranial model

this case study, the relatively small number of patients and the young age of some of the patients, makes concrete inferences about the relative quality of the procedures unsafe.

We plot the preoperative and postoperative cranial model parameters of the patients. The expected result is that the parameterisations should show the head shapes moving nearer to the mean of the training examples. Figure 30 demonstrates the full head meshes of the patients after registration to the 3D scans, for both preoperative and postoperative shapes. The results are shown in Figs. 31 and 32. The parameterisations are validated to move nearer to the mean, which is at the origin of the plots. To our knowledge, this is

the first use of full head 3DMMs in a craniofacial clinical study.

Figure 33 demonstrates a case study on cranial shape change of a specific patient. Clinicians are also interested in the influence of operation type on facial shape. Here, we use the shape of the full head, both face and cranium, in the analysis. We can clearly observe the improvement after operation when viewing the 3D shape. This is validated by the shape analysis. The preoperative shape parameters are outside the  $2\sigma$  (2 standard deviation) ellipse of the training set, while the postoperative shape parameters are within this ellipse.



**Fig. 33** A case study on the full head shape of a specific patient. Top row: preoperative shape. Bottom row: postoperative shape. Left column: 3D scan. Right column: morphed template. The graph (far right)

of the model shape parameters shows that the patient's head shape moves from outside to inside the 2SD ellipse of the 'under 15 years' Headspace training population

For a quantitative evaluation, we calculate the Mahalanobis distance of each patient. As can be seen in Figs. 31-right and 32-right, the mean of Mahalanobis distance for all patients decreases from 3.21 to 1.18 standard deviations in terms of the BS operation. For the TCR operation, the mean of Mahalanobis distance for all patients decreases from 3.52 to 2.23 standard deviations. The improvement for the BS intervention is 63.24% and the improvement for the TCR intervention is 36.65%. In this case study, the BS intervention has better clinical outcomes than TCR. However we note that the BS intervention is more appropriate for very young children (under 1 year old), while TCR is more appropriate for children older than a year. This is likely to influence the results. However, as mentioned earlier, the patient population sample is too small to be conclusive, and our 3DMM is more appropriate for the TCR group of patients than the younger BS group of patients.

## 12 Conclusion

We released the first publicly-available full head dataset with age, gender and ethnicity metadata for academic research. We proposed a fully-automatic 3DMM training pipeline and used it to build the first shape-texture 3DMM of the full head. The correspondence framework avoids over-fitting and under-fitting in template morphing. The adaptive template improves the correspondence accuracy in local regions, while the LB Regularized Projection helps in decreasing the correspondence error in the normal direction to the shape surface. The correspondence accuracy is state-of-the-art, in terms of publicly-available pipelines. The texture mapping technique captures high quality texture for texture modelling.

The proposed 3DMMs have a powerful ability in reconstruction of incomplete data and model regression to observe the influence of age on craniofacial growth. The flexibility of reconstruction from incomplete craniofacial data helps in many computer vision applications. We present the first use of statistical 3D craniofacial shape models in a clinical study.

**Acknowledgements** We thank *Google Faculty Awards* and our Google sponsor, Forrester Cole, for supporting this research in 2017–2018. We thank the Royal Academy of Engineering and the Leverhulme Trust for priming this work in 2013–2014, via their *Senior Research Fellowship* awards. Headspace data collection was supported by QIDIS from the National Commissioning Group. We thank Rachel Armstrong, Headspace data collection coordinator.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

- Albrecht, T., Knothe, R., & Vetter, T. (2008). Modeling the remaining flexibility of partially fixed statistical shape models. In *2nd MICCAI workshop on mathematical foundations of computational anatomy* (pp. 160–169).
- Amberg, B., Romdhani, S., & Vetter, T. (2007). Optimal step nonrigid ICP algorithms for surface registration. In *IEEE conference on computer vision and pattern recognition* (pp. 1–7).
- An, Z., Deng, W., Yuan, T., & Hu, J. (2018). Deep transfer network with 3D morphable models for face recognition. In *2018 13th IEEE international conference on automatic face gesture recognition* (pp. 416–422).

- Basso, C., Verri, A., & Herder, J. (2007). Fitting 3D morphable models using implicit representations. *Journal of Virtual Reality and Broadcasting*, 4(18), 1–10.
- Beeler, T., & Bradley, D. (2014). Rigid stabilization of facial expressions. *ACM Transactions on Graphics (TOG)*, 33(4), 44.
- Besl, P. J., & McKay, N. D. (1992). Method for registration of 3-D shapes. In *Sensor fusion IV: Control paradigms and data structures* (Vol. 1611, pp. 586–607). International Society for Optics and Photonics.
- Blanz, V., & Vetter, T. (1999). A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (pp. 187–194).
- Blanz, V., & Vetter, T. (2003). Face recognition based on fitting a 3D morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9), 1063–1074.
- Bolkart, T., & Wuhrer, S. (2013). Statistical analysis of 3D faces in motion. In *2013 International conference on 3D vision-3DV 2013* (pp. 103–110). IEEE.
- Bolkart, T., & Wuhrer, S. (2015). A groupwise multilinear correspondence optimization for 3D faces. In *Proceedings of the IEEE international conference on computer vision* (pp. 3604–3612).
- Bookstein, F. L. (1989). Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6), 567–585.
- Booth, J., Roussos, A., Ponniah, A., Dunaway, D., & Zafeiriou, S. (2018). Large scale 3D morphable models. *International Journal of Computer Vision*, 126(2–4), 233–254.
- Booth, J., Roussos, A., Zafeiriou, S., Ponniah, A., & Dunaway, D. (2016). A 3D morphable model learnt from 10,000 faces. In *Proceedings of CVPR* (pp. 5543–5552).
- Brunton, A., Lang, J., Dubois, E., & Shu, C. (2011). Wavelet model-based stereo for fast, robust face reconstruction. In *2011 Canadian conference on computer and robot vision (CRV)* (pp. 347–354).
- Cao, C., Weng, Y., Zhou, S., Tong, Y., & Zhou, K. (2014). Facewarehouse: A 3D facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3), 413–425.
- Chen, Y., & Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3), 145–155.
- Cootes, T. F., Edwards, G. J., & Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 6, 681–685.
- Cootes, T. F., & Taylor, C. J. (1995). Combining point distribution models with shape models based on finite element analysis. *Image and Vision Computing*, 13(5), 403–409.
- Cootes, T. F., Taylor, C. J., Cooper, D. H., & Graham, J. (1995). Active shape models—their training and application. *Computer vision and image understanding*, 61(1), 38–59.
- Creusot, C., Pears, N. E., & Austin, J. (2013). A machine-learning approach to keypoint detection and landmarking on 3D meshes. *International Journal of Computer Vision*, 102(1), 146–179.
- Dai, H., Pears, N., & Duncan, C. (2017a). A 2D morphable model of craniofacial profile and its application to craniosynostosis. In *Medical image understanding and analysis, communications in computer and information science* (Vol. 723).
- Dai, H., Pears, N., Smith, W., & Duncan, C. (2017b). A 3D morphable model of craniofacial shape and texture variation. In *2017 IEEE international conference on computer vision (ICCV)* (pp. 3104–3112). IEEE.
- Dai, H., Pears, N., Smith, W., & Duncan, C. (2018a). Symmetric shape morphing for 3D face and head modelling. In *2018 13th IEEE international conference on automatic face and gesture recognition (FG 2018)* (pp. 91–97). IEEE.
- Dai, H., Pears, N., Smith, W., & Duncan, C. (2018b). Symmetric shape morphing for 3D face and head modelling. In *2018 13th IEEE international conference on automatic face gesture recognition (FG 2018)* (pp. 91–97).
- De Smet, M., & Van Gool, L. (2010). Optimal regions for linear model-based 3D face reconstruction. In *Asian conference on computer vision* (pp. 276–289).
- Dryden, I. L., & Mardia, K. V. (1998). *Statistical shape analysis*. Chichester: John Wiley and Sons.
- Duncan, C., Armstrong, R., Pears, N. E., Dai, H., & Smith, W. (2018). The headspace dataset. <https://www-users.cs.york.ac.uk/~nep/research/Headspace/>. Accessed 5 Nov 2019.
- Garrido, P., Zollhöfer, M., Casas, D., Valgaerts, L., Varanasi, K., Pérez, P., et al. (2016). Reconstruction of personalized 3D face rigs from monocular video. *ACM Transactions on Graphics*, 35(3), 28:1–28:15.
- Gerig, T., Forster, A., Blumer, C., Egger, B., Lüthi, M., Schönborn, S., & Vetter, T. (2017). Morphable face models: An open framework. CoRR [arXiv:1709.08398](https://arxiv.org/abs/1709.08398).
- Golovinskiy, A., Matusik, W., Pfister, H., Rusinkiewicz, S., & Funkhouser, T. (2006). A statistical model for synthesis of detailed facial geometry. *ACM Transactions on Graphics (TOG)*, 25, 1025–1034.
- Harrison, C. R., & Robinette, K. M. (2006). *Principles of fit to optimize helmet sizing*. Technical report, Air Force Research Lab Wright-Patterson.
- Kendall, D. G. (1984). Shape manifolds, procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society*, 16(2), 81–121.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110.
- Lüthi, M., Gerig, T., Jud, C., & Vetter, T. (2017). Gaussian process morphable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40, 1860–1873.
- Madsen, D., Lüthi, M., Schneider, A., & Vetter, T. (2018). Probabilistic joint face-skull modelling for facial reconstruction. In *Proceedings of CVPR* (pp. 5295–5303).
- Myronenko, A., & Song, X. (2010). Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12), 2262–2275.
- Paysan, P., Knothe, R., Amberg, B., Romdhani, S., & Vetter, T. (2009). A 3D face model for pose and illumination invariant face recognition. In *Sixth IEEE international conference on advanced video and signal based surveillance, 2009. AVSS'09* (pp. 296–301).
- Petr, M., & Ivana, K. (2015). Hairstyles modeling for police identikit. In *Proceedings of the 31st Spring conference on computer graphics* (pp. 151–158). ACM.
- Salazar, A., Wuhrer, S., Shu, C., & Prieto, F. (2014). Fully automatic expression-invariant face correspondence. *Machine Vision and Applications*, 25(4), 859–879.
- Saragih, J. M., Lucey, S., & Cohn, J. F. (2011). Real-time avatar animation from a single image. In *IEEE international conference on automatic face and gesture recognition 2011* (pp. 213–220).
- Sorkine, O., & Alexa, M. (2007). As-rigid-as-possible surface modeling. In *Proceedings of the fifth Eurographics symposium on geometry processing* (pp. 109–116).
- Styner, M. A., Rajamani, K. T., Nolte, L. P., Zsemlye, G., Székely, G., Taylor, C. J., & Davies, R. H. (2003). Evaluation of 3D correspondence methods for model building. In *Information processing in medical imaging* (pp. 63–75).
- ter Haar, F. B., & Veltkamp, R. C. (2008). 3D face model fitting for recognition. In *European conference on computer vision* (pp. 652–664).
- Thompson, D. W. (1917). *On growth and form*. Cambridge University Press.
- Tran, L., & Liu, X. (2018). Nonlinear 3D face morphable model. arXiv preprint [arXiv:1804.03786](https://arxiv.org/abs/1804.03786).



- Van Der Maaten, L. (2014). Accelerating t-SNE using tree-based algorithms. *The Journal of Machine Learning Research*, 15(1), 3221–3245.
- Vlasic, D., Brand, M., Pfister, H., & Popović, J. (2005). Face transfer with multilinear models. *ACM Transactions on Graphics (TOG)*, 24, 426–433.
- Wang, Z., & Li, Q. (2011). Information content weighting for perceptual image quality assessment. *IEEE Transactions on Image Processing*, 20(5), 1185–1198.
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612.
- Wang, Z., Simoncelli, E. P., & Bovik, A. C. (2003). Multiscale structural similarity for image quality assessment. In *The thirty-seventh Asilomar conference on signals, systems and computers, 2003* (Vol. 2, pp. 1398–1402). IEEE.
- Wu, Y., & Ji, Q. (2019). Facial landmark detection: A literature survey. *International Journal of Computer Vision*, 127(2), 115–142.
- Yang, F., Bourdev, L., Shechtman, E., Wang, J., & Metaxas, D. (2012). Facial expression editing in video using a temporally-smooth factorization. In *2012 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 861–868). IEEE.
- Yang, F., Wang, J., Shechtman, E., Bourdev, L., & Metaxas, D. (2011). Expression flow for 3D-aware face component transfer. In *ACM transactions on graphics (TOG)* (vol. 30, p. 60).
- Yin, L., Chen, X., Sun, Y., Worm, T., & Reale, M. (2008). A high-resolution 3D dynamic facial expression database. In *8th IEEE international conference on automatic face and gesture recognition, 2008. FG'08* (pp. 1–6). IEEE.
- Zhou, Y., & Zaferiou, S. (2017). Deformable models of ears in-the-wild for alignment and recognition. In *2017 12th IEEE international conference on automatic face and gesture recognition (FG 2017)* (pp. 626–633). IEEE.
- Zhu, X., & Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *Proceedings of CVPR* (pp. 2879–2886).

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.