Generating 3D Facial Expressions with Recurrent Neural Networks

Hyewon Seo, Guoliang Luo

Generating 3D Facial Expressions with Recurrent Neural Networks

Hyewon Seo and Guoliang Luo

Abstract. Learning based methods have proved effective at high-quality image synthesis tasks, such as content-preserving image rendering with different style, and the generation of new images depicting learned objects. Some of the properties that make neural networks suitable for such tasks, for example robustness to the input's low-level feature, and the ability to retrieve contextual information, are also desirable in 3D shape domain. During last decades, data-driven methods have shown successful results in 3D shape modeling tasks, such as human face and body shape synthesis. Subtle, abstract properties on the geometry that are instantly detected by our eyes but are nontrivial to synthesize, have successfully been achieved by tuning a shape model built from example shapes. Recent successful learning techniques, e.g. deep neural networks, also exploit this shape model, since the regular grid assumption with 2D images does not have a straightforward equivalent in the common shape representation in 3D, thus do not easily generalize to 3D shapes.

Here, we concentrate on the 3D facial expression generation task, an important problem in computer graphics and other application domains, where existing data-driven approaches mostly rely on direct shape capture or shape transfer. At the core of our approach is a recurrent neural network with a landmark-based shape representation. The network is trained to estimate a sequence of pose change, thus generate a specific facial expression, by using a set of motion-captured facial expression sequences. Our technique promises to significantly improve the quality of generated expressions while extending the potential applicability of neural networks to sequence of 3D shapes.

Hyewon Seo
CNRS-University of Strasbourg, France, e-mail: seo@unistra.fr

Guoliang Luo
East China Jiaotong University, China, e-mail: luoguoliang@ecjtu.edu.cn

1 Introduction

Facial shape modeling is a long-sought subject in computer graphics and computer animation, with interesting applications in many areas. Traditionally, facial shape has been sculpted or interactively designed by CG artists by using CG softwares that are equipped with geometric shape interpolation (Kalra et al., 1992) or physics-based simulation of muscle actions (Terzopoulos and Waters, 1990). During 90's computational methods started to appear that aim at automatic reconstruction of 3D shape models from 2D photo images (Lee et al., 2000) or 3D range scans (Lee et al., 1995). Often, only the static shape can be modeled realistically, and the animation of the reconstructed models has been handled as a separate process with devoted techniques for motion capture. Recent evolutions in the technology for capturing moving shapes have changed this paradigm, with multi-view acquisition systems that allow the simultaneous capture of shapes and motions (DeCarlo and Metaxas, 1996; Pighin et al., 1999). Similarly, recent 4D (3D+time) laser-scanners now enable the capture of 3D human face geometry under motion (Beeler et al., 2011; Cosker et al., 2011). These techniques can be assisted by deformation transfer or animation retargeting (Noh and Fidaleo, 2000; Blanz et al., 2003; Vlasic et al., 2006), which helps reuse the captured animation on new facial models. This line or research has evolved to data-driven methods that make use of a set of 3D shape datasets and the priors collected from the data shapes. A common strategy employed by data-driven methods is to learn the model by performing a dimension reduction, often PCA (Principal Component Analysis), on a dataset of face scans. It goes with several different names such as subspace, or morphable model (MM) or statistical model, all of which refer to a same technique that captures shape and texture variations in observed human faces with a set of basis. Based on the *linear model* that captures shape and texture variations in observed human faces with a set of basis, they offer powerful modelling functionalities: A complete 3D model can be reconstructed by using only a single 2D photo as input, and generation of new face models or modifying existing ones can be performed by adjusting a few parameters whose mapping to a full facial model has been found from the database. We summarize MM in Section 2.1, and review other shape representations considerable for facial modeling in Section 2.2.

These days, recent deep learning techniques start to replace linear function approximators with deep neural networks (DNN) in facial modeling tasks, to achieve improved performance. Most of these methods, on the other hand, have focused on the optimal 2D to 3D shape estimation, i.e. generation of a 3D shape from a 2D input photo showing a face of arbitrary pose. Typically, the neural network learns and estimates the Basel Face model (BFM) parameters (Paysan et al., 2009) of the 3D face model from 2D photos (Garrido et al., 2016; Jiang et al., 2018). Compared to linear models like BFM, the DNN uses larger datasets spanning a large variety of not only shape and texture, but also pose or expression so that the network can learn the corresponding parameters of the 3D shape from an arbitrary, 'in-the-wild' facial image. More importantly, it can learn *nonlinear model*, the variation of facial

shapes due to the facial identity, and due to the expression or pose change of a face. In Section 2.3, we review state-of-the-art facial modelers adopting DNNs.

One observation is that through all these works, the expression has been modeled as separate, independent entity from the shape identity. The expression-driven facial deformation is learned as a separate phenomenon from their shape identity-driven variation, and then the two modalities are combined when a new shape is synthesized. Similarly, from an observable shape that often comes with the shape identity and the expression mixed together, a modeler decouples the two entities. Conveniently, the extracted expression component from one person can be easily combined with or transferred to another facial shape, to depict the same semantic expression on the new shape identity. This model is very powerful yet simple, but it cannot capture the potential correlation between two modalities. For example, the shape change elicited by a smiling expression on a young Asian face will be the same as on an old, Caucasian face. In Section 2.3, we address a more challenging alternative, i.e. modelling the subtle correlation between the facial expression sequence and the shape identity.

2 Facial Shape Modeling

2.1 Facial Shape Space

In their 3D morphable model work that are also known as the Basel Face Model (BFM), Blanz and Vetter (1999) have constructed a subspace for facial identity variation to reduce the dense facial geometry (several thousand vertices per face). Using a common polygon mesh representation, each vertex's position and color vary between example faces, but its semantic identity remains the same – A vertex located at the tip of the nose in one face should be located at the tip of the nose in all faces. To obtain a consistent representation across all examples, they use a modified version of 2D optical flow in the cylindrical parameterization of head scans. Consequently, a face is represented by a shape-vector $\mathbf{S} = (x_1, y_1, z_1, x_2 \ldots y_n, z_n)^T \in R^{3n}$ and a texture-vector $\mathbf{T} = (r_1, g_1, b_1, r_2 \ldots g_n, b_n)^T \in R^{3n}$, containing the coordinates and the color values of its n vertices, respectively. From the m exemplar faces that are in correspondence, PCA is applied to m shape vectors and m texture vectors. The facial shape is then described in the space of a reduced dimension, as a vector of weights α to the eigenshapes $\mathbf{s}_i$ i.e. the eigenvectors of the covariance matrix of $\mathbf{S}_i$. The facial color is similarly described as a vector of weights β to the eigencolors $\mathbf{t}_i$:

$$\mathbf{S}(\bar{\alpha}) = \bar{\mathbf{S}} + \sum_{i=1} \alpha_i \cdot \mathbf{s}_i, \mathbf{T}(\bar{\beta}) = \bar{\mathbf{T}} + \sum_{i=1} \beta_i \cdot \mathbf{t}_i,$$

where $\bar{\mathbf{S}}$ and $\bar{\mathbf{T}}$ denotes the mean shape and texture, respectively. In this facial subspace, arbitrary new faces can be generated by varying these parameters (vectors of weights) that control the shape and texture. Model fitting to a given image is

formulated as an optimization by minimizing the image-space discrepancy between the input 2D image and the rendered image of the 3D face synthesized with the current parameter set. Thanks to the learnt linear model, the solution space becomes compact and constrained, thus solvable by common optimization techniques.

High-level facial attributes (femaleness, concave or hooked nose, thickness of eyebrow, etc.) have been shown to be manipulated by forming shape and texture vectors $\Delta\mathbf{S}$ and $\Delta\mathbf{T}$ that, when added to or subtracted from a face, will change a specific attribute while keeping all other attributes as constant as possible. Such attribute vectors are computed as weighted sums of manually-labeled faces. Expression is handled as one of facial attributes. Formally,

$$\mathbf{\Delta S} = \sum_{i=1} \mu_i(\mathbf{S}_i - \bar{\mathbf{S}}), \mathbf{\Delta T} = \sum_{i=1} \mu_i(\mathbf{T}_i - \bar{\mathbf{T}}), \tag{1}$$

where μ_i is the attribute value labeled to $(\mathbf{S}_i, \mathbf{T}_i)$, and $\bar{\mathbf{S}}, \bar{\mathbf{T}}$ are mean shape and texture vectors.

2.2 Other Shape Representations

The 3D shape representation methods can be categorized into the global and the local feature based. The global shape descriptors, such as the shape histogram (Ankerst et al., 1999) and histogram of gradient (Scherer et al., 2010), are based on the global statistical analysis to represent an entire object. On the contrary, local feature descriptors detect local distinctive features, which are more precise and robust against the occlusions. Below we summarize a number of local shape representations that have been adopted for the 3D face data.

Key-points: The representative points for 3D faces can be either the distinctive points based on the quantified measurement of the tension, normal, curvature of each point, or the anatomical landmarks on/around eyes, nose, mouth, etc. For example, shape diameter function is the averaged radial segment length at each point (Shapira et al., 2008); Heat Kernel Signature measures the energy of heat distribution which reflects the local surface shape at each point (Sun et al., 2009); Spin image encodes each point with respect to the normal vector (Johnson, 1997); shape index (histogram of surface normal) can be used to detect the landmarks at the eye corners and the nose tip (Canavan et al., 2015). The key-point extraction process, however, can be computationally heavy and sensitive to occlusions.

Feature-curve: Based on the precise nose tip point detection, typical feature curves of 3D face include the iso-depth contour, the iso-geodesic curve and the radial curves. The quality of these curves highly relies on the correctness of the nose tip detection, and occlusions may cause the incompleteness of the curves (Samad and Iftekharuddin, 2016).

Local surface based: The local surface-based feature descriptors are normally based on the local statistics of the regional geometrical properties such as normal,

geodesic distance, curvatures, etc (Li and Zhang, 2007). Compared to the key-point based methods, such methods are more robust for representing facial expressions.

We note that most of the existing feature representations for 3D faces are for facial recognition, but they may be not directly applicable for 3D face reconstruction or synthetics. For example, given a 3D face model of point clouds, we can easily compute the per-vertex curvatures, which can be applied to visualize the 3D face and recognize the facial expression. However, the reverse process is not quite possible, i.e. one cannot reconstruct the 3D face by using the computed curvatures.

2.3 Modern Facial Modelers using DNN

Recently, the revolutionary development of deep learning started to replace linear function approximators with deep neural networks to achieve drastically improved performance. Most of these methods, on the other hand, are devoted to the generation of a 3D shape from a 2D input photo showing a face of arbitrary pose. Typically, the neural network learns and estimates the BFM parameters of the 3D face model from 2D photos (Garrido et al., 2016; Jiang et al., 2018). Compared to linear models like BFM (Section 2.1), the DNN uses larger datasets spanning a large variety of not only shape and texture, but also pose or expression so that the network can learn the corresponding parameters of the 3D shape from an arbitrary, 'in-the-wild' facial image. More importantly, it can learn nonlinear model, the variation of facial shapes due to the facial identity, and to the expression or pose change of a face.

E2FAR (End to end 3D Face Reconstruction with DNN) by Dou et al. (2017) shows how a trained DNN takes a 2D facial image as input and predicts the optimal identity and expression parameters to minimize the error in the 3D space – the difference between the reconstructed 3D face and the ground truth (the shape that has been used to produce the 2D input image) (Dou et al., 2017). They make use of the BFM without any encoder that extracts shape parameters from input images and concentrate on learning the mapping function $f : I \rightarrow \alpha_d, \alpha_e$, that maps the 2D image I to the BFM shape parameters. With only shapes considered, and the network learns (1) identity parameters α_d, and (2) expression parameters α_e.

MOFA (MOdel-based deep convolutional Facial Autoencoder) (Tewari et al., 2017) shows a good example of commonly adopted NN architecture, i.e. the combination of a CNN encoder and model-based decoder. The CNN encoder learns to extract semantically meaningful parameters from a single image. Similarly to Dou et al. (2017), they use the facial subspace based on the Basel Facial Model, and once again, pose, shape, expression, texture, illumination are parametrized independently. Given a scene description in the form of a semantic code vector, the decoder generates a synthetic image of the corresponding face. The loss function is defined as a photometric error between the synthesized image and the input image. The error combines three error terms, landmark error; photometric error, and statistical regularization error, as is often the case in similar optimization setting.

Nonlinear 3D face morphable model (Nonlinear 3DMM) operates in a similar fashion, with the decoder-encoder architecture (Tran and Liu, 2018). They train their own encoder to extract feature descriptors on the given scene, and use texture image instead of per-vertex color for the sake of better preservation of spatial relation among pixels. Given a set of 2D facial images, an encoder is learned to estimate the shape, texture and projection parameters, and two DNs (decoders) to decode the estimated parameters to a 3D shape and texture, respectively, with an objective that the rendered image with the encoded parameters can approximate the original image well.

Among a few works that adopt other representation than BFM is that of Jackson et al. (2017). They convert the 3D face surface into binary 3D voxels, i.e, the voxels crossed by face surface with 1s, otherwise 0s. The conversion of the un-structured 3D face model into the structured volume form allows a direct adaptation of the advanced DNNs to 3D face data. In specific, they use DNNs to encode the projection process from 3D voxels to 2D images (Yang et al., 2017; Jackson et al., 2017). However, due to the computational costs, the size of the volume is kept small.

3 Facial Animation Modeling

Facial animation modeling has evolved along a similar path as the facial shape modeling, i.e. from interactive key-framing to capture-based reconstruction. Thanks to recent evolutions in the technology for capturing moving shapes, it is now possible to acquire full 4D shapes of human faces including geometry, motion and appearance with advanced multi-view acquisition systems. However, most current techniques focus on modeling the shape instances in a frame-by-frame manner, and do not model the temporal aspect of the shape evolution.

Flame (Faces Learned with Articulated Model and Expressions) presents an extension of BFM to 4D facial model (Li et al., 2017). Given a 4D scans, displacements from a 3D template shape are modeled for each frame as a function of three decoupled parameters describing the shape identity, head pose, and expression. The temporal evolution is not modeled, i.e. the facial parameters have been found for each frame. Formally, the mapping function is defined as

$$M(\beta, \theta, \psi) : \mathbb{R}^{|\beta|\times|\theta|\times|\psi|} \rightarrow \mathbb{R}^{3N},$$

where β, θ, ψ are the shape identity, head pose, and expression parameters, and N is the number of vertices of the template shape. The facial parameters are found for each frame, i.e. the temporal evolution of the found parameters is not modeled

In their work on the facial reenactment, Kim et al. (2018) also reconstruct a sequence of 3D facial models from a video input, by fitting the BFM to each frame: The identity parameter set is estimated in the first frame and is kept constant throughout the frames, and all other parameters are estimated every frame. Again,

the reconstruction as well as estimation is limited to the spatial domain of the facial shape.

In facial animation transfer (Blanz et al., 2003; Vlasic et al., 2006; Thies et al., 2015), source and target sequences of facial poses are analyzed to separate the identity, pose, and expression components in a frame-by-frame manner. Typically, the expression component is extracted from the source video and transferred to replace that of the target video. However, the transferred, expression-driven shape change is a direct function of the source face and its expression, neglecting the shape identity of the target face.

4 Learning-based Generation of 3D Facial Expression Sequences

We build a facial expression estimator model by training a neural network that learns to generate facial expression sequences. Unlike existing methods that treat the expression data as a set of shape instances, we aim at modeling the temporal evolution of the facial shape. RNN seemed appropriate as it can learn temporal relation between consecutive frames. Initially, we considered GAN (Generative Adversarial Network) (Goodfellow et al., 2014) as well, which has shown a superior performance on generating high quality images based on the raw 2D image input (Goodfellow et al., 2014; Brock et al., 2019). However, this would require the additional preprocessing of time-normalization of the facial expression dataset, and its learning capability on temporal dependency may not be as good as RNN. For these reasons, we have adopted RNNs for our work.

4.1 RNN on Time Series Data

Our work builds on the recent success of deep neural networks in sequence data analysis. In particular, RNNs achieved promising results in processing and modeling sequential, time-series data, such as text-to-text translation (Sutskever et al., 2014; Cho et al., 2014), scene description (Vinyals et al., 2015), and music composition (Boulanger-Lewandowski et al., 2012). Unlike many feedforward neural networks, an RNN maintains hidden internal states that is not only dependent on the current input, but also relies on the previous hidden state and hence the previous inputs. It takes inputs, updates its internal state through recurrent connection that spans adjacent time steps, and generates outputs at every time-step iteratively. Therefore, the history of inputs affects the generation of outputs. Formally, the fixed length hidden state $h(t)$ is updated with the current input $x(t)$ by using a nonlinear function f:

$$h(t+1) = f(h(t), x(t)). \tag{2}$$

RNN training is similar to feedforward network straining in the sense that network parameters are updated incrementally via backpropagation. Since RNNs include

recurrent edges that span adjacent time steps the same parameters are shared across all time steps, gradients at the current time step would affect gradient computation at the previous time steps. This process is called back propatation through time (BPTT). In our work, we build our RNNs using LSTM units, which preserves gradients well while BPTT/layers and thus can deal well with long-term dependencies.

4.2 Learning to generate 3D Facial animations

Overview. Here we leverage the well-known capability of a recurrent network to capture temporal information, in order to model the facial expression sequence modeling. We set our goal to generate new sequences to animate an arbitrary facial shape by using a trained network. We could have employed generative adversarial networks (GANs) instead (Goodfellow et al., 2014; Brock et al., 2018), but it would be computationally more expensive and less reliable for ensuring temporal fluidity. Indeed, an RNN is not only capable of learning temporal relation between frames but also more suitable for handling sequential data with arbitrary lengths. The main inspiration of our work comes from a recent advances neural image caption (NIC), which makes use of a deep CNN that encodes a given image into a fixed-length vector representation, and uses the vector as the initial hidden state of a decoder RNN that generates the target caption sentences. Here, we propose to directly use the landmark locations of a neutral face as the initial input to a decoder RNN, as the representation of the facial geometry (landmark coordinate vectors) lies in the same dimensional space as the sequence data (landmark displacement vectors). Expression-specific prior is assumed, that is, each network is devoted to one specific facial expression elicited by a basic emotion.

Data preparation. We have used the facial mocap data from Binghamton University (BU-3DFE) Yin et al. (2006), consisting of 606 facial expression sequences captured from 101 people (58 females and 43 male subjects). For each subject, six universal facial expressions (anger, disgust, fear, happiness, sadness and surprise) are elicited, whose shape and texture have been recorded at a video frame rate of 25 fps. Also provided is a sequence of 3D coordinates of 83 landmarks located on the face (Figure 1). With a number of exceptions, most sequences begin and end with neutral head/face expression poses.

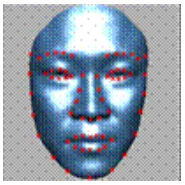

Fig. 1 83 landmarks have been defined in our facial mocap data (BU-3DFE) (Yin et al., 2006).

We use the landmark displacements (i.e. offset coordinates from the previous frame), rather than absolute coordinates. Thus, from the original sequence data containing the ordered list of landmark coordinates, we generate landmark displacement data. Initially we tried to decouple the head motion from the expression-driven deformation and removed the rigid motion in the landmark displacement data. However, we found from some early experiments that the head motion encoded in the landmark displacements actually increases the

expressivity of generated expressions. This is especially noticeable in 'surprise' expression, where many subjects lean their heads slightly backward and so does the generated expression when the head motion had been included in the training data. Thus, we use the original landmark displacement data including the head motion. Thus, from the original data containing the ordered list of landmark positions, we generate landmark coordinate displacements including the head motion.

Each sequence contains a varying number of frames. In the 'happy' dataset, for example, the sequence length varies from 69 to 115 frames. We uniformized the length of the sequence by choosing a constant as the number of frames for all sequences. A sequence data with longer frames has been be cut at the end, and the one with shorter frames has been zero-padded until the end.

Data representation. Since we aim to model the sequence as a whole instead of modeling each frame individually, we must employ a light-weight representation for the facial shape and avoid using vectors/matrices of large dimensionality. In our modeler, we use the landmark-based shape representation in order to maintain a moderate data dimension. A facial expression is then represented as a sequence of landmark displacement vectors applied to each landmark point on a rest pose.

Formally, our training data can be described with the input and output pairs $(\mathbb{X}, \mathbb{Y})$. Let $\mathbb{X}$ be the input observations and $\mathbf{X}^i \in \mathbb{X}$ a sample from our observation, where $i \in 1,..n$; n is the number of subjects in the dataset.

$\mathbf{X}^i \in \mathbb{M}_{f\times 3m}$ contains the expression sequence from the i-th subject, where f+1 is the number of frames, and m is the number of landmarks, respectively. It is an ordered set of landmark displacement vectors, as written by

$$\mathbf{X}^i = \left[\boldsymbol{x}_1^i \; \boldsymbol{x}_2^i \; \ldots \boldsymbol{x}_t^i \right]^T,$$

where $\boldsymbol{x}_j^i = \left[dx_{1,j}^i, dy_{1,j}^i, dz_{1,j}^i, \ldots, dz_{m,j}^i \right]$ is a row vector of size $3m$ denoting the landmark displacements between (t+1)-th frame t-th frame ($t = 1, \ldots f$). In our data, $\mathbf{X}^i$ has been recorded with 83 landmarks and its sequence length f has been normalized to 135, so it is 135×249 (83 times 3) dimensional. We have tried to apply PCA to reduce the data dimension to dozens but the gain in computation time had been insignificant.

RNN to learn the facial expression. Given a 3D face mesh (in a rest pose) whose landmark locations have been identified, a neural network is trained to predict the sequence of landmark displacements, which will animate the given mesh when added to the given mesh sequentially. Formally, the variable number of facial expression poses (as represented by landmark displacement vectors) previously seen by the network are expressed by a fixed length hidden state $\boldsymbol{h}_t$, which is updated with the current input $\boldsymbol{x}_t$ by using a nonlinear function, i.e., $\boldsymbol{h}_{t+1} = h(\boldsymbol{h}_t, \boldsymbol{x}_t)$. The output $\boldsymbol{y}_t$ is evaluated as a linear function of the hidden state, i.e., $\boldsymbol{y}_t = g(\boldsymbol{h}_t)$, which can be implemented as a fully connected (FC) layer.

Figure 2 illustrates the predictor network architecture used in this paper: An LSTM (long short term memory) network consisting of multiple LSTM layers with

a fully-connected (FC) decoder unit. LSTM is a variant of RNN which preserves gradients well while backpropagating through time/layers and thus can deal with long-term dependencies. The input to the network is the current displacement vector $\mathbf{x}_t = \left(dx_1^t, dy_1^t, \ldots, dz_m^t\right)$ encoding x, y, and z offsets of each landmark. The input vector $\mathbf{x}_t$ is passed to weighted connections to a stack of recurrently connected hidden layers to compute first the hidden vector sequences $\mathbf{h}^n = \left(h_1^n, \ldots, h_T^n\right), n = 1 \ldots 3$ and then the output vector $\mathbf{y}_t$. $\mathbf{y}_t$ is the predicted landmark displacement vector in the next time step, $\mathbf{x}_{t+1}$, and is used to predict the next landmark displacement by being fed as input to the network in the next time step. During the training of network, we directly minimize the sum of squared error over the predicted displacements and the ones from the observation data. Thus we define our loss function that measures the mean squared error over the displacements.

$$L_{disp} = \sum \|\mathbf{y}_t - \mathbf{x}_{t+1}\|^2$$

where $\mathbf{y}_t$ is the predicted output at time t, and $\mathbf{x}_{t+1}$ is the corresponding ground truth at time t+1.

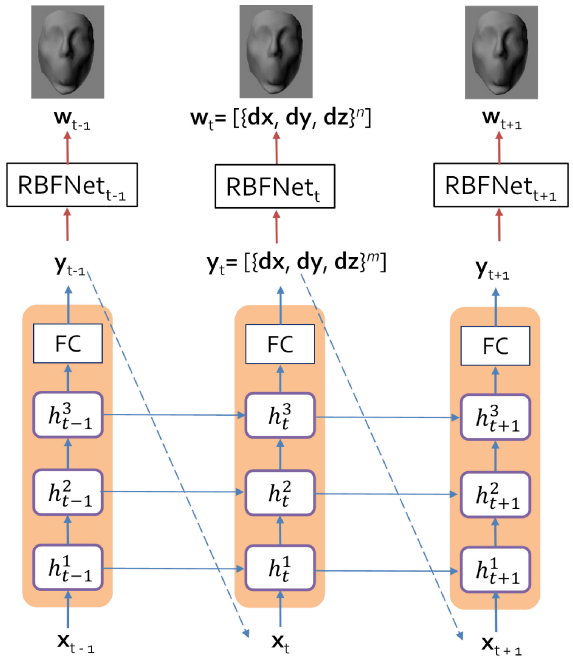

Fig. 2 Deep recurrent neural network architecture of our facial animation synthesizer. The rounded rectangles represent LSTM cells, each containing 128 neurons, and the rectangles fully connected layer. The solid lines represent weighted connections and the dashed lines predictions. The predicted landmark displacements y_t are fed as input to RBF nets, which computes a deformation field.

In our experiments, we trained each expression network with 3 hidden layers with 128 hidden units and a dense output layer, by using Tensorflow (Tensorflow) deep learning library (Abadi et al., 2015). The hidden layers used the *tanh* nonlinear function, although other activation functions could have been used. Figure 3 illustrates the blocks of training data. The input data block $\{\mathbf{X}^i\}$ assembles the sequences of landmark displacement vectors from all subjects and is $n \times f \times 3m$ dimensional. The first element $\mathbf{X}_1$ of every expression sequence is set to a zero vector, signaling the start of the sequence. An output sequence $\mathbf{Y}^i$ has been generated by left-shifting $\mathbf{X}^i$ by one, i.e. $\mathbf{y}_t^i = \mathbf{x}_{t+1}^i$ (t=1,...$f-1$), and by filling the last frame with a zero-vector, $\mathbf{y}_f^i = [0, ...0]^T$.

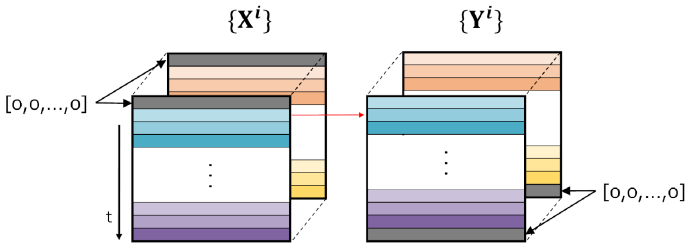

Fig. 3 Training data for the network. An input data block, $\{\mathbf{X}^i\}$, is the compilation of individual expression sequences $\mathbf{X}^i$ of i-th subject, where each row is a concatenation of displacement vectors (dx, dy, and dz). The output sequence, $\{\mathbf{Y}^i\}$, has been generated by left-shifting the $\mathbf{X}^i$ tensor by one.The initial frames of $\mathbf{X}^i$ as well as the last frames of $\mathbf{Y}^i$ are set to zero-vectors.

The model parameters were tuned by using Adam optimization method, with the learning rate $\alpha = 0.001$ and default values for the other parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = e^{-8}$. Although deep RNNs are known to take a long time to train, thanks to the compressed nature of the landmark-based facial animation data, we were able to train our RNNs in approximately 30 minutes on an Intel Core i5 (3.2 GHz) personal computer, with 10 000 epochs over 101 samples with batch size 5.

After the network has been trained, each output vector $\mathbf{y}_t$ recursively feeds back into the network at the next time step, until a full-length sequence is generated. At each frame, $\mathbf{y}_t$ is also used to parametrize an RBF network over the Euclidian space, which computes the deformation of the full face mesh by evaluating the displacement of each vertex on the mesh (see the next subsection). Note that the RBF parametrization followed by the full-mesh deformation is performed at each frame.

Landmark to full mesh deformation. The trained RNN networks generate a sequence of displacement vector for the landmark set, from which we compute a sequence of deformed mesh of a given a face shape (Figure 4). Among many available techniques that we could use for the mesh warping, we use Radial basis function (RBF) networks, a universal solver for scattered interpolation problems (Powell, 2007). Consider a real valued function $w_x(\mathbf{v}) : \mathbb{R}^3 \rightarrow \mathbb{R}$ that approximates the deformation (along x-axis, without losing the generality) of the face mesh given a

sparse set of function values known at landmark locations $\left\{\mathrm{w}_x\left(\mathbf{v}^i\right) = d_x^i \mathbf{s}\right\}$ where $\mathbf{v}^i$ is the location of the i-th maker and d_x^i denotes its displacement along x-axis. Note that the RBF works on a multidimensional domain but a scalar function, thus, we compute $\mathrm{w}(\mathbf{x})$ for each dimension, i.e., $\mathrm{w}_x(\mathbf{x})$, $\mathrm{w}_y(\mathbf{x})$, and $\mathrm{w}_z(\mathbf{x})$ for the displacement along each coordinate. $\mathrm{w}_x(\cdot)$ is assumed to be as a weighted sum of radial basis function and a linear term, i.e.,

$$\mathrm{w}_x(\mathbf{v}) = \sum_i^m q_i \cdot \phi\left(\left\|\mathbf{v} - \mathbf{v}^i\right\|\right) + p(\mathbf{v}) \tag{3}$$

where $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}$ is a radial based function (Gaussian in our case), q_i are scalar weights of i-th kernel, $\mathbf{v}^i \in \mathbb{R}^3$ are the kernel centers of RBF, m is the number of interpolants, and $p(\mathbf{x})$ is a polynomial. To determine q_i and $p(\mathbf{x})$, we use the known function values at interpolates, i.e., the displacements at landmark locations, as written by $w_x(v^i) = d_x^i, j = 1, \ldots, m$. This results in a linear system

$$\begin{bmatrix} \mathrm{w}_x\left(\mathbf{v}^1\right) \\ \vdots \\ \mathrm{w}_x\left(\mathbf{v}^m\right) \end{bmatrix} = \begin{bmatrix} \phi_{11} & \ldots & \phi_{1m} \\ \vdots & \ddots & \vdots \\ \phi_{m1} & \ldots & \phi_{mm} \end{bmatrix} \begin{bmatrix} q_1 \\ \vdots \\ q_m \end{bmatrix} = \begin{bmatrix} d_x^1 \\ \vdots \\ d_x^m \end{bmatrix} \tag{4}$$

where $\phi_{ij} = \exp\left(-\frac{\left\|\mathbf{v}^i - \mathbf{v}^j\right\|}{2\sigma^2}\right)$. As the matrix $\boldsymbol{\phi}$ can be computed from the Gaussian evaluation using landmark distances as input, and the displacement vector $\mathbf{d}_x$ is known, the weight vector $\mathbf{q}$ can be found by solving for the linear system, i.e., $\mathbf{q} = \boldsymbol{\phi}^{-1} \cdot \mathbf{d}_x$ which determines the function $\mathrm{w}_x(\cdot)$.

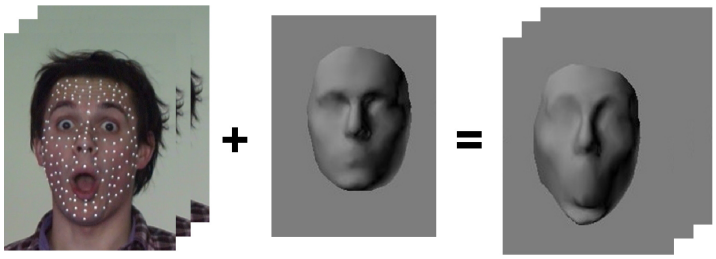

Fig. 4 Given a sequence of landmark set and a static mesh, we generate a sequence of deformed mesh driven by the landmark locations.

Once the warp functions $\mathrm{w}_x(\cdot)$, $\mathrm{w}_y(\cdot)$, $\mathrm{w}_z(\cdot)$ are found, we can evaluate them at each vertex location of the full mesh so that it conforms to the displaced landmarks. The warping functions should be solved for every frame, since each frame yields new interpolants. With the number of landmarks less than 100, it can be solved efficiently by using the LU decomposition.

Results. Figure 5 shows some of the expression sequences generated by our 'Anger' network. Note that we have applied the landmark displacement prediction to a same face geometry, to mask off the visual effects originating from different shape identities. In Figure 6, the generated expressions are applied to different face models. The deformations deriving a specific facial expression are learned properly and gives plausible results. This is despite the fact that the captured landmark data is sometimes very noisy, and that the shape variety of facial models is large.

Fig. 5 Snapshots of 'Anger' expression sequences generated by our model.

The overall evaluation is a neural-net inference followed by RBF warping of a full mesh. The training and evaluation for the landmark-to-mesh deformation is performed in a per-frame basis. Each frame takes about 0.1 seconds for a mesh comprised of 10 000 vertices, accounting for a total time of only a few seconds for the generation of an entire sequence.

5 Discussion

We have addressed, and provided an overview of, facial shape and animation modeling. In particular, a new deep learning-based method for facial expression generation has been presented, which models a facial animation as a temporal entity, i.e. a sequence of deformations applied to a facial mesh as represented by a set of sparsely sampled landmarks. LSTM RNNs have been trained with dis-placements of land-

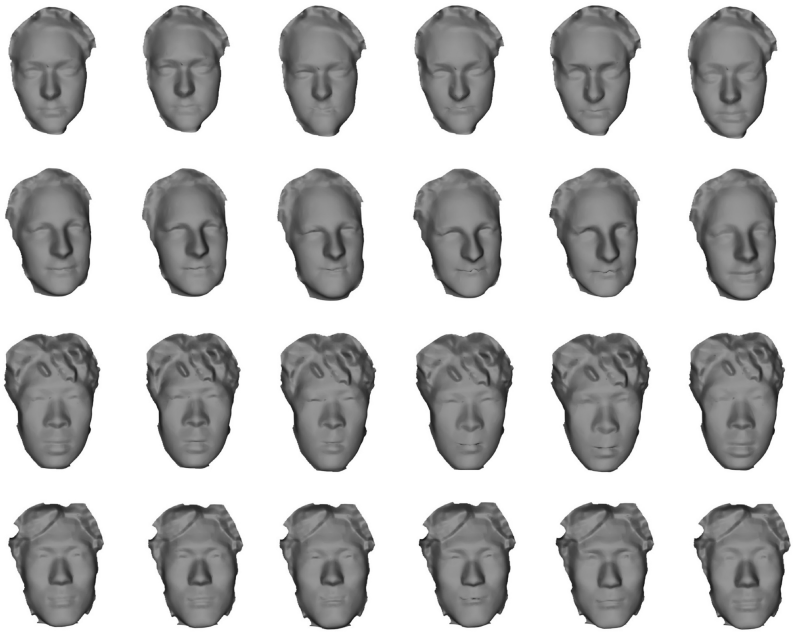

Fig. 6 Facial expression sequences generated by our 'Anger' network have been applied to different faces.

mark locations, one for each expression of a basic emotion. Unlike existing modelers where only static expression poses are learned in a frame-by-frame manner, our modeler learns the temporal evolution of the facial pose change as a whole, thus enabling a 4D facial expression modeling. As a result, our technique offers a promising solution for the facial expression sequence modeling, significantly improving the quality of generated expression while extending the potential applicability of neural networks to 4D shape data (time-varying shapes).

The landmark-based representation of facial mesh used in this work requires a same landmark configuration on the database and on a new face mesh where the generated animation will be applied. Moreover, the RBF-based landmark-to-full mesh animation can sometimes result in unstable deformation caused by the extrapolation, notably along the mesh boundary. These limitations can be complemented by deep learning-based methods for automatic landmark extraction, and for landmark-driven fine mesh deformation. Another possibility is to adopt more compact geometric representations that represent the full facial mesh with a moderate data size. In the future, our model can be trained on a larger database and thus can take advantage of additional data.

References

Kalra P, Mangili A, Thalmann NM, Thalmann D (1992) Simulation of facial muscle actions based on rational free form deformations. Computer Graphics Forum 11(3):59–69, DOI https://doi.org/10.1111/1467-8659.1130059, URL https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-8659.1130059, https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-8659.1130059

Terzopoulos D, Waters K (1990) Physically-based facial modelling, analysis, and animation. The Journal of Visualization and Computer Animation 1(2):73–80

Lee WS, Gu J, Thalmann N (2000) Generating animatable 3d virtual humans from photographs. Comput Graph Forum 19, DOI 10.1111/1467-8659.00392

Lee Y, Terzopoulos D, Waters K (1995) Realistic modeling for facial animation. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, Association for Computing Machinery, New York, NY, USA, SIGGRAPH '95, p 55âĂŞ62, DOI 10.1145/218380.218407, URL https://doi.org/10.1145/218380.218407

DeCarlo D, Metaxas D (1996) The integration of optical flow and deformable models with applications to human face shape and motion estimation. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Los Alamitos, CA, USA, p 231, DOI 10.1109/CVPR.1996.517079, URL https://doi.ieeecomputersociety.org/10.1109/CVPR.1996.517079

Pighin F, Szeliski R, Salesin DH (1999) Resynthesizing facial animation through 3d model-based tracking. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol 1, pp 143–150 vol.1, DOI 10.1109/ICCV.1999.791210

Beeler T, Hahn F, Bradley D, Bickel B, Beardsley P, Gotsman C, Sumner R, Gross M (2011) High-quality passive facial performance capture using anchor frames. ACM Trans Graph 30:75, DOI 10.1145/2010324.1964970

Cosker D, Krumhuber E, Hilton A (2011) A facs valid 3d dynamic action unit database with applications to 3d dynamic morphable facial modeling. In: 2011 International Conference on Computer Vision, pp 2296–2303, DOI 10.1109/ICCV.2011.6126510

Noh JY, Fidaleo D (2000) Animated deformations with radial basis functions. In: In ACM Virtual Reality and Software Technology (VRST, pp 166–174

Blanz V, Basso C, Poggio T, Vetter T (2003) Reanimating faces in images and video. Computer Graphics Forum 22(3):641–650, DOI https://doi.org/10.1111/1467-8659.t01-1-00712

Vlasic D, Brand M, Pfister H, Popovic J (2006) Face transfer with multilinear models. ACM Transactions on Graphics 24, DOI 10.1145/1185657.1185864

Paysan P, Knothe R, Amberg B, Romdhani S, Vetter T (2009) A 3d face model for pose and illumination invariant face recognition. DOI 10.1109/AVSS.2009.58

Garrido P, Zollhöfer M, Casas D, Valgaerts L, Varanasi K, Pérez P, Theobalt C (2016) Reconstruction of personalized 3d face rigs from monocular video. ACM Transactions on Graphics 35(3):28, URL https://academic.microsoft.com/paper/2398381847

Jiang L, Zhang J, Deng B, Li H, Liu L (2018) 3d face reconstruction with geometry details from a single image. IEEE Transactions on Image Processing 27(10):4756–4770, URL https://academic.microsoft.com/paper/2593956217

Ankerst M, Kastenmüller G, Kriegel HP, Seidl T (1999) 3d shape histograms for similarity search and classification in spatial databases. Lecture Notes in Computer Science pp 207–226

Scherer M, Walter M, Schreck T (2010) Histograms of oriented gradients for 3d object retrieval URL https://academic.microsoft.com/paper/2467579536

Shapira L, Shamir A, Cohen-Or D (2008) Consistent mesh partitioning and skeletonisation using the shape diameter function. The Visual Computer 24(4):249–259, URL https://academic.microsoft.com/paper/1989540182

Sun J, Ovsjanikov M, Guibas LJ (2009) A concise and provably informative multi-scale signature based on heat diffusion. symposium on geometry processing 28(5):1383–1392, URL https://academic.microsoft.com/paper/2100657858

Johnson AE (1997) Spin-images: A representation for 3-d surface matching URL https://academic.microsoft.com/paper/168966905

Canavan SJ, Liu P, Zhang X, Yin L (2015) Landmark localization on 3d/4d range data using a shape index-based statistical shape model with global and local constraints. Computer Vision and Image Understanding 139:136–148, URL https://academic.microsoft.com/paper/1667484106

Samad MD, Iftekharuddin KM (2016) Frenet frame-based generalized space curve representation for pose-invariant classification and recognition of 3-d face. IEEE Transactions on Human-Machine Systems 46(4):522–533, URL https://academic.microsoft.com/paper/2344785877

Li X, Zhang H (2007) Adapting geometric attributes for expression-invariant 3d face recognition. In: IEEE International Conference on Shape Modeling and Applications 2007 (SMI '07), pp 21–32, URL https://academic.microsoft.com/paper/2095595190

Dou P, Shah SK, Kakadiaris IA (2017) End-to-end 3d face reconstruction with deep neural networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1503–1512, URL https://academic.microsoft.com/paper/2605701576

Tewari A, Zollhöfer M, Kim H, Garrido P, Bernard F, Pérez P, Theobalt C (2017) Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. arXiv preprint arXiv:170310580 URL https://academic.microsoft.com/paper/2952080583

Tran L, Liu X (2018) Nonlinear 3d face morphable model. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 7346–7355, URL https://academic.microsoft.com/paper/2796822548

Jackson AS, Bulat A, Argyriou V, Tzimiropoulos G (2017) Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. arXiv preprint arXiv:170307834 URL https://academic.microsoft.com/paper/2951863354

Yang J, Liu Q, Zhang K (2017) Stacked hourglass network for robust facial landmark localisation. In: 2017 IEEE Conference on Computer Vi-

sion and Pattern Recognition Workshops (CVPRW), pp 2025–2033, URL https://academic.microsoft.com/paper/2736728583

Li T, Bolkart T, Black MJ, Li H, Romero J (2017) Learning a model of facial shape and expression from 4d scans. ACM Transactions on Graphics 36(6):194, URL https://academic.microsoft.com/paper/2769666294

Kim H, Garrido P, Tewari A, Xu W, Thies J, Niessner M, Pérez P, Richardt C, Zollhöfer M, Theobalt C (2018) Deep video portraits. international conference on computer graphics and interactive techniques 37(4):1–14, URL https://academic.microsoft.com/paper/2806833697

Thies J, Zollhöfer M, Nießner M, Valgaerts L, Stamminger M, Theobalt C (2015) Real-time expression transfer for facial reenactment. ACM Trans Graph 34(6), DOI 10.1145/2816795.2818056, URL https://doi.org/10.1145/2816795.2818056

Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville AC, Bengio Y (2014) Generative adversarial nets. In: Advances in Neural Information Processing Systems 27, pp 2672–2680, URL https://academic.microsoft.com/paper/2099471712

Brock A, Donahue J, Simonyan K (2019) Large scale gan training for high fidelity natural image synthesis. In: ICLR 2019 : 7th International Conference on Learning Representations, URL https://academic.microsoft.com/paper/2893749619

Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. arXiv preprint arXiv:14093215 URL https://academic.microsoft.com/paper/2949888546

Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:14061078 URL https://academic.microsoft.com/paper/2950635152

Vinyals O, Toshev A, Bengio S, Erhan D (2015) Show and tell: A neural image caption generator. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3156–3164, URL https://academic.microsoft.com/paper/1895577753

Boulanger-Lewandowski N, Bengio Y, Vincent P (2012) Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. arXiv preprint arXiv:12066392 URL https://academic.microsoft.com/paper/1819710477

Brock A, Donahue J, Simonyan K (2018) Large scale GAN training for high fidelity natural image synthesis. CoRR abs/1809.11096, URL http://arxiv.org/abs/1809.11096, 1809.11096

Yin L, Wei X, Sun Y, Wang J, Rosato MJ (2006) A 3d facial expression database for facial behavior research. In: The 7th International Conference on Automatic Face and Gesture Recognition (FGR06), pp 211–216, URL https://academic.microsoft.com/paper/2137306662

Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow IJ, Harp A, Irving G, Isard M, Jia Y, Józefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray DG, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar

K, Tucker PA, Vanhoucke V, Vasudevan V, Viégas FB, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:160304467

Powell MJD (2007) A view of algorithms for optimization without derivatives 1 URL https://academic.microsoft.com/paper/2187467903