Sparse to Dense Dynamic 3D Facial Expression Generation

Naima Otberdout  
Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France  
naima.otberdout@univ-lille.fr

Claudio Ferrari  
Deparment of Architecture and Engineering  
University of Parma, Italy  
claudio.ferrari2@unipr.it

Mohamed Daoudi  
IMT Nord Europe, Institut Mines-Télécom, Univ. Lille, Centre for Digital Systems, F-59000 Lille, France  
mohamed.daoudi@imt-nord-europe.fr

Stefano Berretti  
Media Integration ad Communication Center  
University of Florence, Italy  
stefano.berretti@unifi.it

Alberto Del Bimbo  
Media Integration ad Communication Center  
University of Florence, Italy  
alberto.delbimbo@unifi.it

Abstract

In this paper, we propose a solution to the task of generating dynamic 3D facial expressions from a neutral 3D face and an expression label. This involves solving two sub-problems: (i) modeling the temporal dynamics of expressions, and (ii) deforming the neutral mesh to obtain the expressive counterpart. We represent the temporal evolution of expressions using the motion of a sparse set of 3D landmarks that we learn to generate by training a manifold-valued GAN (Motion3DGAN). To better encode the expression-induced deformation and disentangle it from the identity information, the generated motion is represented as per-frame displacement from a neutral configuration. To generate the expressive meshes, we train a Sparse2Dense mesh Decoder (S2D-Dec) that maps the landmark displacements to a dense, per-vertex displacement. This allows us to learn how the motion of a sparse set of landmarks influences the deformation of the overall face surface, independently from the identity. Experimental results on the CoMA and D3DFACS datasets show that our solution brings significant improvements with respect to previous solutions in terms of both dynamic expression generation and mesh reconstruction, while retaining good generalization to unseen data. Code and models are available at https://github.com/CRISTAL-3DSAM/Sparse2Dense.

1. Introduction

Synthesizing dynamic 3D (4D) facial expressions aims at generating realistic face instances with varying expressions or speech-related movements that dynamically evolve across time, starting from a face in neutral expression. It finds application in a wide range of graphics applications spanning from 3D face modeling, to augmented and virtual reality for animated films and computer games. While recent advances in generative neural networks have made possible the development of effective solutions that operate on 2D images [17, 37], the literature on the problem of generating facial animation in 3D is still quite limited.

To perform a faithful and accurate 3D facial animation,
three main challenges arise. First, the identity of the subject whose neutral face is used as starting point for the sequence should be maintained across time. Second, the applied deformation should correspond to the specified expression/motion that is provided as input, and should be applicable to any neutral 3D face. Incidentally, these are major challenges in 3D face modeling, which require disentangling structural face elements related to the identity, e.g., nose or jaw shape, from deformations related to the movable face parts, e.g., mouth opening/closing. Finally, it is required to model the temporal dynamics of the specified expression so to obtain realistic animations.

Some previous works tackled the problem by capturing the facial expression of a subject frame-by-frame and transferring it to a target model [7]. However, in this case the temporal evolution is not explicitly modeled, so the problem reduces to transferring a tracked expression to a neutral 3D face. Some other works animated a 3D face mesh given an arbitrary speech signal and a static 3D face mesh as input [13, 29]. Also in this case, the temporal evolution is guided by an external input, similar to a tracked expression. Instead, here we are interested in animating a face just starting from a neutral face and an expression label.

In our solution, which is illustrated in Figure 1, the temporal evolution and the mesh deformation are decoupled and modeled separately in two network architectures. A manifold-valued GAN (Motion3DGAN) accounts for the expression dynamics by generating a temporally consistent motion of 3D landmarks corresponding to the input label from noise. The landmarks motion is encoded using the Square Root Velocity Function (SRVF) and compactly represented as a point on a hypersphere. Then, a Sparse2Dense mesh Decoder (S2D-Dec) generates a dense 3D face mesh guided by the landmarks motion for each frame of the sequence. To effectively disentangle identity and expression components, the landmarks motion is represented as a per-frame displacement from a neutral configuration. Instead of directly generating a mesh, the S2D-Dec expands the landmarks displacement to a dense, per-vertex displacement, which is finally used to deform the neutral mesh. The intuition that led to this architecture is the following: the movement induced on the face surface by the underlying facial muscles is consistent across subjects. In addition, it causes the vertex motion to be locally correlated as muscles are smooth surfaces. We thus train the decoder to learn how the displacement of a sparse set of points influences the displacement of the whole face surface. This has the advantage that structural face parts, e.g., nose or forehead, which are not influenced by facial expressions are not deformed, helping in maintaining the identity traits stable. Furthermore, the network can focus on learning expressions at a fine-grained level of detail and generalize to unseen identities.

In summary, the main contributions of our work are: (i) we propose an original method to generate dynamic sequences of 3D expressive scans given a neutral 3D mesh and an expression label. Our approach has the capability of generating strong and diverse expression sequences, with high generalization ability to unseen identities and expressions; (ii) we adapt a specific GAN architecture [37] for dynamic 3D landmark generations, and design a decoder for expressive mesh reconstruction from a neutral mesh and landmarks. Differently from common auto-encoders, the proposed S2D-Dec learns to generate a per-vertex displacement map from a few control points, allowing accurate mesh deformations where the structural face parts remain stable; (iii) we designed a novel reconstruction loss that weights the contribution of each vertex based on its distance from the landmarks. This proved to augment the decoder capability of generating accurate expressions.

2. Related Work

Our work is related to methods for 3D face modeling, facial expression generation guided by landmarks, and dynamic generation of 3D faces, i.e., 4D face generation.

3D face modeling. The 3D Morphable face Model (3DMM) as originally proposed in [2] is the most popular solution for modeling 3D faces. The original model and its variants [3, 6, 19, 21, 34, 36, 38] capture face shape variations both for identity and expression based on linear formulations, thus incurring in limited modeling capabilities. For this reason, non-linear encoder-decoder architectures are attracting more and more attention. This comes at the cost of reformulating convolution and pooling/unpooling like operations on the non-regular mesh support [5, 33, 42]. Ranjan et al. [40] proposed an auto-encoder architecture that builds upon newly defined spectral convolution operators, and pooling operations to down/up-sample the mesh. Bouritsas et al. [4] improved upon the above by proposing a novel graph convolutional operator enforcing consistent local orderings on the vertices of the graph through the spiral operator [32]. Despite their impressive modeling precision, a recent work [19] showed that they heavily suffer from poor generalization to unseen identities. This limits their practical use in tasks such as face fitting or expression transfer. We finally mention that other approaches exist to learn generative 3D face models, such as [1, 35]. However, instead of dealing with meshes they use alternative representations for 3D data, such as depth images or UV-maps.

To overcome the above limitation, we go beyond self-reconstruction and propose a mesh decoder that, differently from previous models, learns expression-specific mesh deformations from a sparse set of landmark displacements.

Facial expression generation guided by landmarks. Recent advances in neural networks made facial landmark detection reliable and accurate both in 2D [10, 15, 43] and 3D [24, 46]. Landmarks and their motion are a viable way
to account for facial deformations as they reduce the complexity of the visual data, and have been commonly used in several 3D face related tasks, e.g., reconstruction [21, 31] or reenactment [18, 22]. Despite some effort was put in developing landmark-free solutions for 3D face modeling [8, 9, 23], some recent works investigated their use to model the dynamics of expressions. Wang et al. [44] proposed a framework that decouples facial expression dynamics, encoded into landmarks, and face appearance using a conditional recurrent network. Oterdout et al. [37] proposed an approach for generating videos of the six basic expressions given a neutral face image. The geometry is captured by modeling the motion of landmarks with a GAN that learns the distribution of expression dynamics.

These methods demonstrated the potential of using landmarks to model the dynamics of expressions and generate 2D videos. In our work, we instead tackle the problem of modeling the dynamics in 3D, exploring the use of the motion of 3D landmarks to both model the temporal evolution of expressions and animate a 3D face.

4D face generation. While many researchers tackled the problem of 3D mesh deformation, the task of 3D facial motion synthesis is yet more challenging. A few studies addressed this issue by exploiting audio features [29, 45], speech signal [13] or tracked facial expressions [7] to generate facial motions. However, none of these explicitly model the temporal dynamics and resort to external information.

To the best of our knowledge, the work in [39] is the only approach that specifically addressed the problem of dynamic 3D expression generation. In that framework, the motion dynamics is modeled with a temporal encoder based on an LSTM, which produces a per-frame latent code starting from a per-frame expression label. The codes are then fed to a mesh decoder that, similarly to our approach, generates a per-vertex displacement that is summed to a neutral 3D face to obtain the expressive meshes. Despite the promising results reported in [39], we identified some limitations. First, the LSTM is deterministic and for a given label the exact same displacements are generated. Our solution instead achieves diversity in the output sequences by generating from noise. Moreover, in [39] the mesh decoder generates the displacements from the latent codes, making it dependent from the temporal encoder. In our solution, the motion dynamics and mesh displacement generation are decoupled, using landmarks to link the two modules. The S2D-Dec is thus independent from Motion3DGAN, and can be used to generate static meshes as well given an arbitrary set of 3D landmarks as input. This permits us to use the decoder for other tasks such as expression/speech transfer. Finally, as pointed out in [39], the model cannot perform extreme variations well. Using landmarks allowed us to define a novel reconstruction loss that weighs the error of each vertex with respect to its distance from the landmarks, encouraging accurate modeling of the movable parts. Thanks to this, we are capable of accurately reproducing from slight to strong expressions, and generalize to unseen motions.

3. Proposed Method

Our approach consists of two specialized networks as summarized in Figure 2. Motion3DGAN accounts for the temporal dynamics and generates the motion of a sparse set of 3D landmarks from noise, provided an expression label, e.g., happy, angry. The motion is represented as per frame landmark displacements with respect to a neutral configuration. These displacements are fed to a decoder network, S2D-Dec, that constructs the dense point-cloud displacements from the sparse displacements given by the landmarks. These dense displacements are then added to a neutral 3D face to generate a sequence of expressive 3D faces corresponding to the initial expression label. In the following, we separately describe the two networks.

3.1. Generating Sparse Dynamic 3D Expressions

Facial landmarks were shown to well encode the temporal evolution of facial expressions [28, 37]. Motivated by this fact, we generate the facial expression dynamics based on the motion of 3D facial landmarks. Given a set of $k$ 3D landmarks, $Z(t) = (x_i(t), y_i(t), z_i(t))_{i=1}^k$, with $Z(0)$ being the neutral configuration, their motion can be seen as a trajectory in $\mathbb{R}^{k \times 3}$ and can be formulated as a parameterized curve in $\mathbb{R}^{k \times 3}$ space. Let $\alpha : I = [0, 1] \rightarrow \mathbb{R}^{k \times 3}$ represent the parameterized curve, where each $\alpha(t) \in \mathbb{R}^{k \times 3}$. For the purpose of modeling and studying our curves, we adopt the Square-Root Velocity Function (SRVF) proposed in [41]. The SRVF $q(t) : I \rightarrow \mathbb{R}^{k \times 3}$ is defined by:

$$q(t) = \frac{\alpha(t)}{\sqrt{\|\alpha(t)\|_2}},$$

with the convention that $q(t) = 0$ if $\dot{\alpha}(t) = 0$. This function proved effective for tasks such as human action recognition [14] or 3D face recognition [16]. Similar to this work, Oterdout et al. [37] proposed to use the SRVF representation to model the temporal evolution of 2D facial landmarks, which makes it possible to learn the distribution of these points and generate new 2D facial expression motions.

In this paper, we extend this idea to 3D by proposing the Motion3DGAN model to generate the motion of 3D facial landmarks represented using the SRVF encoding in (1).

Following [37], we remove the scale variability of the resulting motions by scaling the $L_2$-norm of these functions to 1. As a result, we transform the motion of 3D facial landmarks to points on a hypersphere $C = \{q : [0, 1] \rightarrow \mathbb{R}^{k \times 3}, \|q\| = 1\}$. The resulting representations are manifold-valued data that cannot be handled with traditional generative models.

To learn the distribution of the SRVF representations, we propose Motion3DGAN as an extension of Motion-
GAN [37], a conditional version of the Wasserstein GAN for manifold-valued data [27]. It maps a random vector \( z \) to a point on the hypersphere \( C \) conditioned on an input class label. Motion3DGAN is composed of two networks trained adversarially: a generator \( G \) that learns the distribution of the 3D landmark motions, and a discriminator \( D \) that differentiates between real and generated 3D landmark motions. Motion3DGAN is trained by a weighted sum of an adversarial loss \( L_{adv} \) and a reconstruction loss \( L_r \) such that \( L_M = \alpha_1 L_{adv} + \alpha_2 L_r \). The former is given by:

\[
L_{adv} = \mathbb{E}_{q \sim P_q} [D(\log_p(q), c)] - \mathbb{E}_{z \sim P_z} [D(\log_p(\exp_p(G(z, c))))] + \lambda \mathbb{E}_{q \sim P_q} \left( \| \nabla_q D(\hat{q}) \|_2 - 1 \right)^2 \tag{2}
\]

In (2), \( q \sim P_q \) is an SRVF sample from the training set, \( c \) is the expression label (e.g., mouth open, eyebrow) that we encode as a one-hot vector and concatenate to a random noise \( z \sim P_z \). The last term of the adversarial loss represents the gradient penalty of the Wasserstein GAN [25]. Specifically, \( \hat{q} \sim P_q \) is a random point sampled uniformly along straight lines between pairs of points sampled from \( P_q \) and the generated distribution \( P_g \):

\[
\hat{q} = (1 - \tau) \log_p(q) + \tau \log_p(\exp_p(G(z, c))),
\]

where \( 0 \leq \tau \leq 1 \), and \( \nabla_q D(\hat{q}) \) is the gradient w.r.t. \( \hat{q} \). The functions \( \log_p(\cdot) \) and \( \exp_p(\cdot) \) are the logarithm and the exponential maps, respectively, defined in a particular point \( p \) of the hypersphere. They map the SRVF data forth and back to a tangent space of \( C \) (details in the supplementary material). Finally, the reconstruction loss is defined as:

\[
L_r = \| \log_p(\exp_p(G(z, c))) - \log_p(q) \|_1 \tag{3}
\]

where \( \| \cdot \|_1 \) represents the \( L_1 \)-norm, and \( q \) is the ground truth SRVF corresponding to the condition \( c \). The generator and discriminator architectures are similar to [37].

The SRVF representation is reversible, which makes it possible to recover the curve \( \alpha(t) \) from a new generated SRVF \( q(t) \) by:

\[
\alpha(t) = \int_0^t \| q(s) \| q(s) ds + \alpha(0) \tag{4}
\]

Where \( \alpha(0) \) represents the initial landmarks configuration \( Z(0) \). Accordingly, using this equation, we can apply the generated motion to any landmark configuration, making it robust to identity changes.

### 3.2. From Sparse to Dense 3D Facial Expressions

Our final goal is to animate the neutral mesh \( S^0 \) to obtain a novel 3D face \( S^t \) reproducing some expression, yet maintaining the identity structure of \( S^0 \). Given this, we point at generating the displacements of the mesh vertices from the sparse displacements of the landmarks to animate \( S^t \). In the following, we assume all the meshes have a fixed topology, and are in full point-to-point correspondence.

Let \( \mathcal{L} = \{S^0_1, S^{gt}_1, Z^0_1, Z^{gt}_1\}, \ldots, (S^0_n, S^{gt}_n, Z^0_n, Z^{gt}_n)\} \) be the training set, where \( S^0 \) is a neutral 3D face, \( S^{gt} = \{p^{gt}_1, \ldots, p^{gt}_N\} \in \mathbb{R}^{N \times 3} \) is a 3D expressive face, \( Z^0 \in \mathbb{R}^{k \times 3} \) and \( Z^{gt} \in \mathbb{R}^{k \times 3} \) are the 3D landmarks corresponding to \( S^0 \) and \( S^{gt} \), respectively. We transform this set to a training set of sparse and dense displacements, \( \mathcal{L}_t = \{(D_1, d_1), \ldots, (D_m, d_m)\} \) such that, \( D_i = S^{gt}_i - S^0 \) and \( d_i = Z^{gt}_i - Z^0 \). Our goal here is to find a mapping \( h: \mathbb{R}^{k \times 3} \rightarrow \mathbb{R}^{N \times 3} \) such that \( D_i \approx h(d_i) \).

We designed the function \( h \) as a decoder network (S2D-Dec), where the mapping is between a sparse displacement of a set of landmarks and the dense displacement of the entire mesh points. Finally, in order to obtain the expressive mesh, the dense displacement map is summed to a 3D face in neutral expression, i.e., \( S^t = S^0 + D^t \). The S2D-Dec network is based on the spiral operator proposed in [4]. Our architecture includes five spiral convolution layers, each one followed by an up-sampling layer. More details on the architecture can be found in the supplementary material.

In order to train this network, we propose to use two different losses, one acting directly on the displacements and the other controlling the generated mesh. The reconstruc-
tion loss of the dense displacements is given by,
\[ L_{dr} = \frac{1}{N} \sum_{i=1}^{N} \| D_i^q - D_i^{gt} \|_1, \quad (5) \]
where \( D^q \) and \( D^{gt} \) are the generated and the ground truth dense displacements, respectively. To further improve the reconstruction accuracy, we add a loss that minimizes the error between \( S^q \) and the ground truth expressive mesh \( S^{gt} \). We observed that vertices close to the landmarks are subject to stronger deformations. Other regions like the forehead, instead, are relatively stable. To give more importance to those regions, we defined a weighted version of the \( L1 \) loss:
\[ L_{pr} = \frac{1}{N} \sum_{i=1}^{N} w_i \cdot \| p_i^q - p_i^{gt} \|_1. \quad (6) \]
We define the weights as the inverse of the Euclidean distance of each vertex \( p_i \) in the mesh from its closest landmark \( Z_j \), i.e., \( w_i = \frac{1}{\min_{j} d(p_i, Z_j)}, \quad \forall j \). This provides a coarse indication of how much each \( p_i \) contributes to the expression generation. Since the mesh topology is fixed, we can pre-compute the weights \( w_i \) and re-use them for each sample. Weights are then re-scaled so that they lie in \([0, 1]\). Vertices corresponding to the landmarks, i.e., \( p_i = Z_j \) for some \( j \), are hence assigned the maximum weight. We will show this strategy provides a significant improvement with respect to the standard \( L1 \) loss. The total loss used to train the \( S2D-Dec \) is given by \( L_{S2D} = \beta_1 L_{dr} + \beta_2 L_{pr}. \)

4. Experiments

We validated the proposed method in a broad set of experiments on two publicly available benchmark datasets.

CoMA dataset [40]: It is a common benchmark employed in other studies [4, 40]. It consists of 12 subjects, each one performing 12 extreme and asymmetric expressions. Each expression comes as a sequence of meshes \( S \in \mathbb{R}^{N \times 3} \) (140 meshes on average), with \( N = 5, 023 \) vertices.

D3DFACS dataset [11]: We used the registered version of this dataset [30], which has the same topology of CoMA. It contains 10 subjects, each one performing a different number of facial expressions. In contrast to CoMA, this dataset is labeled with the activated action units of the performed facial expression. It is worth to note that the expressions of D3DFACS are highly different from those in CoMA.

4.1. Training Details

In order to keep Motion3DGAN and \( S2D-Dec \) decoupled, they are trained separately. We used CoMA to train Motion3DGAN, since this dataset is labeled with facial expression classes. We manually divided each sequence into sub-sequences of length 30 starting from the neutral to the apex frame. The first sub-sequence for each subject and expression is used as test set. We use all the others for training as we generate from a random noise at test time.\(^1\). Then, we encoded the motion of \( k = 68 \) landmarks from the sub-sequences in the SRVF representation, and used them to train Motion3DGAN. The landmarks were first centered and normalized to unit norm. Each of the 12 expression labels was encoded as a one-hot vector, concatenated with a random noise vector of size 128.

To comprehensively evaluate the capability of \( S2D-Dec \) of generalizing to either unseen identities or expressions, we performed subject-independent and expression-independent cross-validation experiments. For the subject-independent experiment, we used a 4-fold cross-validation protocol for CoMA, training on 9 and testing on 3 identities in each fold. On D3DFACS, we used the last 7 identities for training and the remaining 3 as test set. Concerning the expression-independent splitting, we used a 4-fold cross-validation protocol for CoMA, training on 9 and testing on 3 expressions in each fold. For D3DFACS, given the different number of expression per subject, the first 11 expressions were used for testing and trained on the rest.

We trained both Motion3DGAN and \( S2D-Dec \) using the Adam optimizer, with learning rate of 0.0001 and 0.001 and mini-batches of size 128 and 16, respectively. Motion3DGAN was trained for 8000 epochs, while 300 epochs were adopted for \( S2D-Dec \). The hyper-parameters of the Motion3DGAN and \( S2D-Dec \) losses were set empirically to \( \alpha_1 = 1, \quad \alpha_2 = 10, \quad \beta_1 = 1 \) and \( \beta_2 = 0.1 \). We chose the mean SRVF of the CoMA data as a reference point \( p \), where we defined the tangent space of \( \mathcal{C} \).

4.2. 3D Expression Generation

For evaluation, we set up a baseline by first comparing against standard 3DMM-based fitting methods. Similar to previous works [20, 31], we fit \( S^n \) to the set of target landmarks \( Z^n \) using the 3DMM components. Since the deformation is guided by the landmarks, we first need to select a corresponding set from \( S^n \) to be matched with \( Z^n \). Given the fixed topology of the 3D faces, we can retrieve the landmark coordinates by indexing into the mesh, i.e., \( Z^n = S^n(I_z) \), where \( I_z \in \mathbb{N}^n \) are the indices of the vertices that correspond to the landmarks. We then find the optimal deformation coefficients that minimize the Euclidean error between the target landmarks \( Z^n \) and the neutral ones \( Z^n \), and use the coefficients to deform \( S^n \). In the literature, several 3DMM variants have been proposed. We experimented the standard PCA-based 3DMM and the DL-3DMM in [20]. We chose this latter variant as it is conceptually similar to our proposal, being constructed by learning a dictionary of deformation displacements. For fair comparison, we built the two 3DMMs using a number of deformation components comparable to the size of the \( S2D-Dec \) input, i.e., \( 68 \times 3 = 204 \). For PCA, we used either 38 components (retaining the 99% of the variance) and 220, while for DL-3DMM we used 220 dictionary atoms.

\(^1\)For reproducibility, the list of the sub-sequences used to train Motion3DGAN will be publicly released.
Table 1. Reconstruction error (mm) on expression-independent (left) and identity-independent (right) splits: comparison with PCA-$k$ 3DMM ($k$ components), DL-3DMM (220 dictionary atoms), and Neural3DMM.

<table>
<thead>
<tr>
<th>Method</th>
<th>CoMA</th>
<th>D3DFACS</th>
<th>CoMA</th>
<th>D3DFACS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA-220</td>
<td>0.76 ± 0.73</td>
<td>0.42 ± 0.44</td>
<td>0.80 ± 0.73</td>
<td>0.56 ± 0.56</td>
</tr>
<tr>
<td>PCA-38</td>
<td>0.90 ± 0.84</td>
<td>0.44 ± 0.45</td>
<td>0.93 ± 0.82</td>
<td>0.58 ± 0.56</td>
</tr>
<tr>
<td>DL3DMM [20]</td>
<td>0.86 ± 0.80</td>
<td>0.73 ± 1.15</td>
<td>0.89 ± 0.79</td>
<td>1.15 ± 1.50</td>
</tr>
<tr>
<td>Neural [4]</td>
<td>0.75 ± 0.85</td>
<td>0.59 ± 0.86</td>
<td>3.74 ± 2.34</td>
<td>2.09 ± 1.37</td>
</tr>
<tr>
<td>Ours</td>
<td>0.52 ± 0.59</td>
<td>0.28 ± 0.31</td>
<td>0.55 ± 0.62</td>
<td>0.27 ± 0.30</td>
</tr>
</tbody>
</table>

With the goal of comparing against other deep models, we also considered the Neural3DMM [4]. It is a mesh autoencoder tailored for learning a non-linear latent space of face variations and reconstructing the input 3D faces. In order to compare it with our model, we modified the architecture and trained the model to generate an expressive mesh $S^g$ given its neutral counterpart as input. To do so, we concatenated the landmarks displacement (of size 204) to the latent vector (of size 16) and trained the network towards minimizing the same $L_{pr}$ loss used in our model. All the compared methods were trained on the same data. Finally, we also identified the FLAME model [31]. Unfortunately, the training code is not available, and using the model pre-trained on external data would not be a fair comparison.

The mean per-vertex Euclidean error between the generated meshes and their ground truth is used as standard performance measure, as in the majority of works [4,19,39,40]. Note that we exclude the Motion3DGAN model here as we do not have the corresponding ground-truth for the generated landmarks (they are generated from noise). Instead, we make use of the ground-truth motion of the landmarks.

4.2.1 Comparison with Other Approaches

Table 1 shows a clear superiority of S2D-Dec over state-of-the-art methods for both the protocols and datasets, proving its ability to generate accurate expressive meshes close to the ground truth in both the case of unseen identities or expressions. In Figure 3, the cumulative per-vertex error distribution on the expression-independent splitting further highlights the precision of our approach, which can reconstruct 90%-98% of the vertices with an error lower than 1mm. While other fitting-based methods retain satisfactory precision in both the protocols, we note that the performance of Neural3DMM [4] significantly drop when unseen identities are considered. This outcome is consistent to that reported in [19], in which the low generalization ability of these models is highlighted. We also note that results for the identity-independent protocol were never reported in the original papers [4,40]. Overall, our solution embraces the advantages of both approaches, being as general as fitting solutions yet more accurate.

Figure 4 shows some qualitative examples by reporting error heatmaps in comparison with PCA, DL-3DMM [20] and Neural3DMM [4] for the identity-independent splitting. The ability of our model as well as PCA and DL-3DMM to preserve the identity of the ground truth comes out clearly, in accordance with the results in Table 1. By contrast, Neural3DMM shows high error even for the neutral faces, which proves its inability to keep the identity of an unseen face. Indeed, differently from the other methods, Neural3DMM encodes the neutral face in a latent space and predicts the 3D coordinates of the points directly, which introduces some changes on the identity of the input face. This evidences the efficacy of our S2D-Dec, that instead learns per-point displacements instead of point coordinates.

4.2.2 Ablation Study

We report here an ablation study to highlight the contribution of each loss used to train S2D-Dec, with particular focus on our proposed weighted-$L1$ reconstruction loss. We conducted this study on the CoMA dataset using the first three identities as a testing set and training on the rest. This evaluation is based on the mean per-vertex error between the generated and the ground truth meshes. We evaluated three baselines, $S1$, $S2$ and $S3$. For the first baseline ($S1$), we trained the model with the displacement reconstruction loss in (5) only. In $S2$, we added the standard $L1$ loss to $S1$, which corresponds to our loss in (6) without the landmark distance weights. To showcase the importance of weighting the contribution of each vertex, in $S3$ we added the landmark distance weights to the $L_{pr}$ loss. Results are shown...
in Table 2, where the remarkable improvement of our proposed loss against the standard $L1$ turns out evident. This is explained by the fact that assigning a greater weight to movable face parts allows the network to focus on regions that are subject to strong facial motions, ultimately resulting in realistic samples.

### 4.3. 4D Facial Expression Evaluation

Since Motion3DGAN generates samples from noise to encourage diversity, the generated landmarks and meshes slightly change at each forward pass. Thus, computing the mean per-vertex error with respect to ground-truth shapes as done in [39] cannot represent a good and reliable measure in this case. So, we evaluated the quality of the generated expression sequences implementing an expression classification solution, similar to [39]. We trained a classifier with one LSTM layer followed by a fully connected layer to recognize the 12 dynamic facial expressions of CoMA given a landmarks sequence as input. We trained this classifier on the same sequences used to train Motion3DGAN. The first expression sample of each identity form the test set, resulting in 144 testing samples.

Since neither the dataset nor the code in [39] are available for comparison, based on the information therein, we implemented a similar architecture relying on an LSTM to generate the per-frame expression and use it as baseline. The LSTM is trained to generate the motion of landmarks from an input code indicating the temporal evolution of the expression from neutral to the apex phase. Also this model was trained on the same data used for Motion3DGAN.

For testing, we generated 144 sequences with both Motion3DGAN and the LSTM generator. The sequences are consistent with those of the Motion3DGAN test set described above. These are then used to generate their corresponding meshes with our S2D-Dec. In Table 3, we report results in terms of classification accuracy and Frechet Inception Distance (FID) [26]. The metrics are computed either using the landmark sequences directly generated by Motion3DGAN and LSTM (Gen-LM row), or those extracted from the generated meshes (Det-LM row). Given that both Motion3DGAN and LSTM act as landmark generators, we also report the results obtained using sequences coming from a “perfect” generator, that is the ground truth landmark sequences of the test set (GT lands. column). This represents a sort of upper bound for the classification accuracy. We note that the features used to compute the FID metric are extracted by the last fully connected layer of our trained classifier, which outputs 512 features per sequence.

In Table 3, we observe that, in all the cases, Motion3DGAN surpasses the accuracy of LSTM to a large extent, providing a clear evidence that the generated sequences better capture the expression dynamics. This is also supported by the lower FID, which indicates that the Motion3DGAN samples better approximate the ground truth motions. The same conclusion is drawn from the closer recognition rate of Motion3DGAN to that obtained with the ground truth sequences. Furthermore, the accuracy increases by first generating the corresponding meshes and then re-extracting the landmarks from them. This suggests the S2D-Dec is capable of maintaining the particular motion, which is also supported by the similar recognition rate obtained with ground truth landmarks (73%) and those detected on their corresponding meshes generated with S2D-Dec (73.61%).

### 4.4. Applications

Our solution has some nice properties that open the way to various applications as shown in Figure 5.

**4D facial expression generation:** In the top two rows of Figure 5, we show the ability of Motion3DGAN to generate sequences for the same expression label that are highly varied on the SRVF representation, of interpolating between generated motions. Given two points on the sphere $q_1$ and $q_2$, representing the motion sequences of two expressions, the geodesic path $\psi(\tau)$ between them is given by,

$$\psi(\tau) = \frac{1}{\sin(\theta)}(\sin(1-\tau)\theta q_1 + \sin(\theta \tau)q_2),$$

where $\theta = \arccos(q_1, q_2)$. This path determines all the points $q_i$ existing between $q_1$ and $q_2$, each of them corresponding to a sequence of landmarks. Using our S2D-Dec, we can transform them to a 4D facial expression. Furthermore, while Motion3DGAN generates only neutral to apex sequences, we can exploit this interpolation to generate mixed 4D facial expressions that switch between the apex phase of different expressions by considering the last frame of each interpolated sequence. Figure 5 illustrates the interpolated faces between the apex frames of two expres-
Figure 5. Applications – From top to bottom: Diversity: the same identity performing the same class of facial expression (mouth side) with two different motions generated by Motion3DGAN. Interpolation: dynamic expressions resulting from the interpolation between two expressions peaks. Transfer: speech transfer from one identity to another. Face neutralization: for each of the four examples, we show the input expressive face, the neutralized face with S2D-Dec, and the ground truth neutral face of the given identity.

5. Conclusions and Limitations

In this paper, we proposed a novel framework for dynamic 3D expression generation from an expression label, where two decoupled networks separately address modeling the motion dynamics and generating an expressive 3D face from a neutral one. We demonstrated the improvement with respect to previous solutions, and showed that using landmarks is effective in modeling the motion of expressions and the generation of 3D meshes. We also identified two main limitations: first, our S2D-Dec generates expression-specific deformations, and so cannot model identities. Moreover, while Motion3DGAN can generate diverse expressions and allows interpolating on the sphere to obtain complex facial expressions, the samples are of a fixed length (i.e., 30 meshes, from neutral to apex). However, as shown in the applications, S2D-Dec can deal with motion of any length since it is independent from Motion3DGAN.

6. Acknowledgments

This work was supported by the French State, managed by National Agency for Research (ANR) National Agency for Research (ANR) under the Investments for the future program with reference ANR-16-IDEX-0004 ULNE and by the ANR project Human4D ANR-19-CE23-0020. This paper was also partially supported by European Union’s Horizon 2020 research and innovation program under grant number 951911 - AI4Media.
References


