

# Synthesising 3D Facial Motion from “In-the-Wild” Speech

Panagiotis Tzirakis<sup>1</sup>, Athanasios Papaioannou<sup>1,2</sup>, Alexander Lattas<sup>1</sup>, Michail Tarasiou<sup>1</sup>,  
Björn Schuller<sup>1,3</sup>, Stefanos Zafeiriou<sup>1,4</sup>

<sup>1</sup> Department of Computing, Imperial College London, UK

<sup>2</sup> Great Ormond Street Institute of Child Health, University College London, UK

<sup>3</sup> ZD.B. Chair of Embedded Intelligence for Health Care and Wellbeing, University of Augsburg, Germany

<sup>4</sup> Center for Machine Vision and Signal Analysis, University of Oulu, Finland

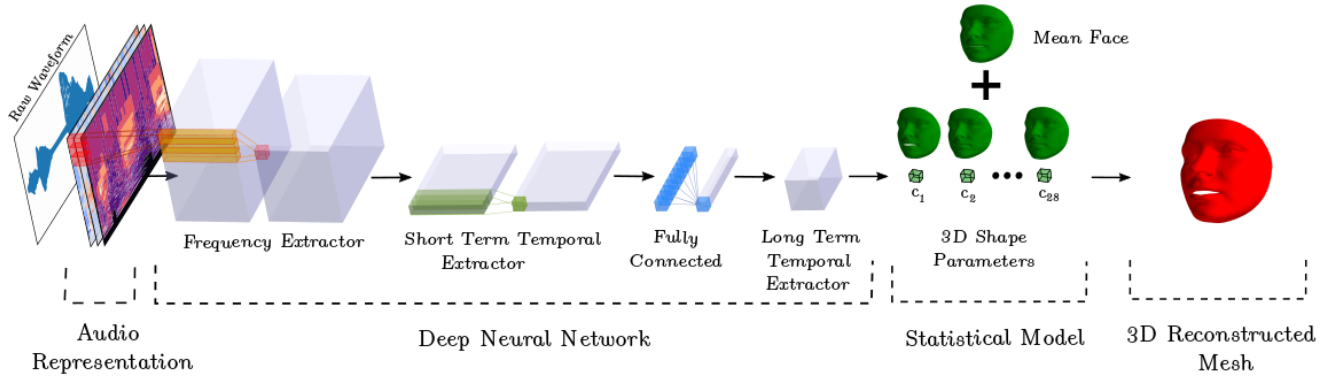


Figure 1. Extraction of 3D facial mesh from raw in-the-wild speech signal.

## Abstract

Synthesising 3D facial motion from speech is a crucial problem manifesting in a multitude of applications such as computer games and movies. Recently proposed methods tackle this problem in controlled conditions of speech. In this paper, we introduce the first methodology for 3D facial motion synthesis from speech captured in arbitrary recording conditions (“in-the-wild”) and independent of the speaker. For our purposes, we captured 4D sequences of people uttering 500 words, contained in the Lip Reading Words (LRW) a publicly available large-scale in-the-wild dataset, and built a set of 3D blendshapes appropriate for speech. We correlate the 3D shape parameters of the speech blendshapes to the LRW audio samples by means of a novel time-warping technique, named Deep Canonical Attentional Warping (DCAW), that can simultaneously learn hierarchical non-linear representations and a warping path in an end-to-end manner. We thoroughly evaluate our proposed methods, and show the ability of a deep learning model to synthesise 3D facial motion in handling differ-

ent speakers and continuous speech signals in uncontrolled conditions.

## 1. Introduction

Synthesis of 3D talking faces is very crucial to many applications including but not limited to computer games, movies post-production (e. g., dubbing), talking faces in virtual reality applications, etc. Currently, the highest quality 3D face synthesis is performed by using facial capture rigs which make use of markers or sensors. Recently, machine learning methods [11, 16, 25], and in particular deep neural networks, have been used in order to train systems that reconstruct the 3D facial geometry of talking faces directly from audio sources. Nevertheless, till now the proposed methods are person or rig-specific [42]<sup>1</sup>, hence do not

<sup>1</sup>In [16] even though the method was trained on a single person, the authors claim that they used the method on other people and the results were meaningful. The above contradicts the current practices in machine learning which require large and diverse sets for good generalisation. Because we could not reproduce their experiments, we contacted the authors

generalise to arbitrary audio sequences. In this paper, we present the first methodology for estimation of the 3D facial motion related to speech directly from raw audio stream.

Estimating the 3D facial motion directly from raw audio samples captured in arbitrary recording conditions is an ill-posed problem since a great number of people uttering a large amount of diverse words need to be captured in 4D (i.e. 3D geometry in time). Even though many efforts have been performed towards collecting 4D expressive faces [7, 38, 37] there is a lack of datasets with talking 4D faces (i.e., 3D speech in time). One such dataset has been proposed by Marshall et al. [20], which captured four people in dyadic interaction (17 mins in total). A limited number of words have been captured in [7] for biometric application, nevertheless the data are not publicly available. Hence, it is very difficult to train a generic method for 3D facial motion reconstruction from audio streams. The most closely related work to ours is by Phamend et al. [25], which used a statistical blendshape model, trained on facial expressions. As we show, these blendshapes cannot model adequately 3D facial motion related to speech.

In this paper, we make the first, to the best of our knowledge, comprehensive effort to estimate 3D facial motion from arbitrary audio streams. To this end, we first capture 4D sequences of people uttering 500 words, contained in an in-the-wild dataset named Lip Reading Words (LRW). After registering the 3D meshes with an adaptive template approach, we learn 3D blendshapes for the speech, which we make publicly available. Each 3D mesh can be parameterised as a set of 3D shape parameters through these 3D blendshapes. Employing a novel time-warping algorithm, named Deep Canonical Attentional Warping (DCAW), we align the speech, that we have 3D ground-truth on, with the corresponding “in-the-wild” speech signals in LRW, and propagate the 3D shape parameters, creating the “in-the-wild” LRW-3D dataset. We train a deep learning model on this dataset and show the ability of the model to predict 3D face motion for speech captured under uncontrolled conditions. Fig. 1 shows the pipeline of our approach. In summary, the contributions of this work are the following:

1. We collect a 4D dataset of people uttering 500 words and learn the first statistical blendshape model for speech which we provide publicly available.
2. In order to train accurate blendshapes, we propose an adaptive shape template method to accelerate the convergence of registration algorithms and achieve a better final shape correspondence.
3. We propose Deep Canonical Attentional Warping (DCAW), a method which learns hierarchical non-linear representations and temporal alignment of two

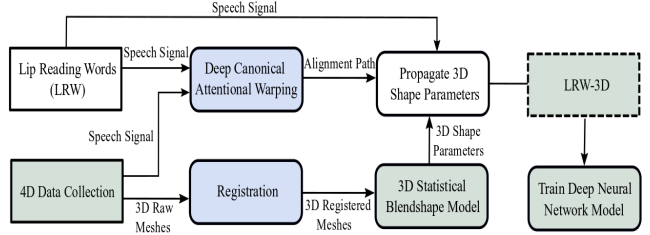


Figure 2. Pipeline of our approach. The proposed methods are represented in light blue, whereas the dataset and models, that we provide publicly available, are shown in light green.

audio signals in an end-to-end manner. Using DCAW we create LRW-3D, and we make publicly available the aligned 3D shape parameters.

4. Finally, we train a speech to 3D deep facial motion model that can operate in nearly real-time, and independently of the speaker in uncontrolled conditions of speech.

## 2. Related Work

There are several traditional approaches [13, 17, 21, 28, 29, 30, 35, 36] that exploit audio signal for 2D or 3D facial animation. More recent approaches utilise deep neural networks (DNN) for this task. These can be categorised in two main groups: linguistic-driven and audio-driven approaches.

**Linguistic-driven approaches.** Language-based methods take advantage of the mapping between phonemes and their visual counterpart visemes. For example, Edwards et al. [11] proposed the JA<sub>w</sub> and LI<sub>p</sub> (JALI) model, a two-dimensional space that represents the jaw and lip movements of a facial animation based on psycholinguistic considerations. The main disadvantage of their study is the need for the speech signal, its text transcript and their alignment to create the facial animation. In another study, Taylor et al. [33] first proposed generating dynamic units for visual speech for realistic visual speech animation. In a more recent study, the authors [32] initially transcribe the speech signal to phoneme labels, which are then fed to a deep fully connected network to predict person-specific shape and appearance parameters obtained by Active Appearance Model (AAM). A main limitation of this work is the need for speech to phoneme labels conversion.

**Audio-driven approaches.** Audio-based methods drive facial animations using only audio cues. Zhou et al. [42] utilises audio features for an automatic and near real-time animation by driving a JALI face-rig. The authors propose VisemeNet, a three stage deep learning model based on Long Short-Term Memory (LSTM) networks that are fed

with audio features. The first two stages comprises of extracting phonemes and landmarks, while the third extracts visemes. In a different study, Karras et al. [16] proposed a deep convolutional neural network for 3D facial animation using auto-correlation audio features. The network models unknown variation of the audio cues by learning a small dimensional vector. Its output is the entire face and is trained using a three-way loss function that ensures temporal stability and correct responses under animation. The main limitation of their method is that the trained network is person-specific and as such it cannot generalise well to different speakers. Pham et al. [25] used the same network architecture as Karras et al. but with melspectrograms as input. Their model was trained to predict the rotation and expression blending parameters extracted by a 3D face tracker. Suwajanakorn et al. [31] used audio features to synthesize videos of Obama focusing on the mouth region and taking the rest of the head and torso from stock footage. However, their method is person-specific with tens of hours of audio signal, and requires heavy hand-engineered work.

In comparison with the aforementioned methods, our approach is able to provide 3D face motion estimation not only in uncontrolled speech conditions but also is speaker-independent, i.e., independent of the speaker. On top of that, we are the first to propose a statistical blendshape model appropriate for speech.

### 3. Building Speech Blendshapes Model

In this section, we describe the pipeline for building the speech blendshapes model. The process consists of: (a) collecting a dataset of 3D meshes with various people uttering a set of words (Sec. 3.1), (b) registering all the meshes to a common template (Sec. 3.2), and (c) building a statistical model on the registered meshes (Sec. 3.3).

#### 3.1. Data Acquisition

To construct a set of blendshapes appropriate for speech we needed to capture 4D sequences (i.e. 3D geometry in time) of people talking. The choice of the utterance spoken by our participants is driven by our goal to train a model that can operate under unconstrained audio conditions (in-the-wild), and is speaker independent. To this purpose, we utilise the 500 words contained in the publicly available Lip Reading Words (LRW) in-the-wild dataset [9] which contains almost 1000 videos per word, summing to approximately 450,000 videos. These videos were captured from TV broadcasts (e.g. news or interviews) in uncontrolled environments and contain 1000 speakers, making it an excellent fit for our purposes.

We used the DI4D dynamic system<sup>2</sup> to capture and build 4D faces. This system consists of six cameras (two pairs

of stereo cameras and one pair of texture cameras, 30FPS,  $1200 \times 1600$ ). Before every recording, a calibration was necessary and was performed by utilising a  $10 \times 10$ , 20 mm checkerboard. Two 4-lamp fluorescent lights were placed on each side to provide consistent and uniform lights. One microphone is used to capture the audio signal in 44,1 kHz sampling rate.

We captured 2 native and 2 non-native English speakers reading out the 500 words from the aforementioned dataset. We should note that the choice of the non-native speakers is to add extra variability to the mouth region when we develop our blendshapes. In total, the whole process took 20 min for each participant, and, approximately, 20,000 3D meshes were acquired for each individual (equivalent of 660 s of recording).

#### 3.2. Registration

**Automatic Annotation.** Before processing our 3D captured meshes, we need to re-parameterise them such that all the meshes have the same number of vertices joined into a common triangulation. Two approaches exist that perform such task and they differ on the space the registration is performed. The first method performs the dense registration in the 2D space, namely the UV-space [24, 10]. The second method registers directly (i.e. in the 3D space) the mesh and the template [1, 22].

In this paper, we follow the latter approach as UV-based correspondence approaches introduce non-linearities into the process and require an extra step for rasterizing the UV image [5]. To this end, we perform the registration in the 3D space between a neutral 3D mesh and the mean shape of Large Scale Facial Model (LSFM) [5]. The registration is performed by utilising Non-rigid Iterative Closest Point (NICP), which has as a prerequisite the template and the mesh to be close in terms of euclidean distance.

The high number of 3D meshes do not allow us to manually annotate 3D facial landmarks. To automate the process we utilise a sparse alignment method proposed by Booth et al. [6] that can automatically compute sparse annotations on the 3D meshes. More specifically, for each mesh we apply the face detection and alignment framework proposed by Zhang et al. [40] on its corresponding 2D texture image, where the correspondence to the 3D coordinates are known, to robustly locate a set of 68 sparse annotations (landmarks) [39] in the 2D space. Exploiting the correspondence between the texture image and the 3D mesh, we get the corresponding 3D positions of the landmarks.

**Adaptive Template and Dense Registration.** The majority of NICP shape registration methods use the same 3D template shape to deform all 3D meshes of a dataset [5]. Even though this approach can be sufficient for large datasets with neutral shapes, it is not the case when there is a large variation in terms of expressions and lip move-

<sup>2</sup><http://www.di3d.com>

ments in each mesh. In our captured 3D meshes the position and shape of the mouth change frequently so if a single template shape is used important parts of it would not be close to the corresponding parts of 3D meshes. Hence, the registered results would have visible errors and inaccurate correspondences.

To alleviate from this problem, we propose an adaptive template approach where for each 3D mesh in our dataset we adapt the original template using sparse shape information (i.e. 68 landmarks).

In particular, by leveraging the expression blendshapes created by Cheng et al. [7], we compute the 3D shape parameters for each mesh through linear regression between the landmarks of the neutral shape and the landmarks of the reconstructed instance as

$$\mathbf{c}_o = \arg \min_{\mathbf{c}} \|\mathbf{l}_n - \mathbf{A}(\mathbf{x}_n + \mathbf{U}_s \mathbf{c})\|_F^2 \quad (1)$$

where  $\mathbf{l}_n \in \mathbb{R}^{3m}$  is a vector with  $m$  landmarks of the neutral shape,  $\mathbf{A} \in \mathbb{R}^{3m \times 3n}$  is an indicator matrix, indicating the position of the 3D landmarks on the reconstructed mesh,  $\mathbf{x}_n \in \mathbb{R}^{3n}$  is the neutral registered shape,  $\mathbf{U}_s \in \mathbb{R}^{3n \times q}$  is a matrix with the  $q$  blend shapes and  $\mathbf{c} \in \mathbb{R}^q$  is the 3D shape parameters vector.

After the calculation of the 3D shape parameters  $\mathbf{c}_o$  that minimise the expression in Eq. 1, we can adapt the neutral template to the current mesh:  $\mathbf{x}_{adapt} = \mathbf{x}_n + \mathbf{U}_s \mathbf{c}_o$ . Finally, NICP can be performed between the adaptive template and the mesh. We should note that even though the blendshapes ( $\mathbf{U}_s$ ) describe various expressions and not speech, they give a good prior for our template. The registration process is illustrated Fig. 3.

### 3.3. Creating Speech Blendshapes

Our final step is to build a set blendshapes appropriate for speech. We start by subtracting the neutral mesh from each registered mesh in the sequence, creating vectors of differences, i.e.,  $\mathbf{d} = \mathbf{x}_n - \mathbf{x}_{adapt} \in \mathbb{R}^{3n}$ . After we stack these vectors into a matrix  $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_T] \in \mathbb{R}^{3n \times T}$ , we apply Principal Component Analysis (PCA) to identify the deformation components  $\mathbf{U}_b$ . We keep 28 blendshapes corresponding to 99.9% of the total variance in the sequence. Hence, a new shape instance can be generated:  $\mathbf{x}_{new} = \mathbf{x}_n + \mathbf{U}_b \boldsymbol{\lambda}$ , where  $\boldsymbol{\lambda}$  are the 3D shape parameters of our model. Finally, for each mesh in our sequence, we compute the 3D shape parameters that constitute our ground truth.

## 4. Lip Reading Words in 3D (LRW-3D)

With the construction of our speech-driven statistical model, we can now use the speech signal of our participant to propagate the 3D shape parameters to the speech signals of the LRW dataset. Our process starts by segmenting our

participants' speech signal to the 500 words, which is accomplished in a semi-automatic manner. More particularly, we first utilise the approach proposed by Elsner et al. [12] to segment our speech signal. This step is crucial to our pipeline as a single faulty segmentation can result in almost a thousand false samples in our dataset, which can lead to ill-generalisable models. To this purpose, we listened each segment, and, when required, we manually adjusted them.

### 4.1. Deep Canonical Attentional Warping (DCAW)

To accurately propagate the 3D shape parameters of our participant to the LRW dataset, we need to eliminate any temporal variations arising in the data. Hence, we compute a temporal alignment between the signal of each word uttered by our participant and the corresponding signals of the LRW. To compute this alignment, we propose *Deep Canonical Attentional Warping (DCAW)*, a novel method that can maximally correlate two data sequences (or views) and find a temporal alignment in an *end-to-end* manner. We leverage deep recurrent convolutional neural networks to spatially transform the raw speech signals, and utilise attention mechanism for the alignment.

The *attentional warping* is performed by computing attention weights between each feature frame of the one view (source) with all the features from the other view (target). We should point out that our data are monotonic and as such we utilise a monotonic attention mechanism [27]. Mathematically, given two views  $\mathbf{X}_k \in \mathbb{R}^{d_k \times T_k}$ , and their features  $\mathbf{h}_i^k, i = \{1, 2, \dots, T_k\}$ , where  $d_k$  and  $T_k$  are the dimensionality and length of view  $k \in \{1, 2\}$ , respectively, the attention between each target feature and the source ones is computed as follows:

$$\alpha_{\mathbf{h}_i^1, \mathbf{h}_j^2} = \frac{\exp(\text{score}(\mathbf{h}_i^1, \mathbf{h}_j^2))}{\sum_{k=1}^{T_2} \exp(\text{score}(\mathbf{h}_i^1, \mathbf{h}_k^2))}, \quad (2)$$

where  $\exp$  is the exponential function, and  $\text{score}$  can be any attention function, such as Bahdanau [4] or Luong [19]. For our purposes, we use the Bahdanau score:

$$\text{score}(\mathbf{h}_i^1, \mathbf{h}_j^2) = \mathbf{v}_i^T \tanh(\mathbf{W}_t \mathbf{h}_i^1 + \mathbf{W}_s \mathbf{h}_j^2), \quad (3)$$

where  $\mathbf{v}_i^T$ ,  $\mathbf{W}_t$  and  $\mathbf{W}_s$  are learnable parameters of the model.

The outcome of Eq. 2, is the attentional matrix  $\mathbf{A} \in \mathbb{R}^{T_1 \times T_2}$  that includes the weights between the two views. We use this matrix to warp the target features with the source ones by multiplying it with the source features, i.e.,  $\mathbf{H}^1 \mathbf{A}$ , where  $\mathbf{H}^1 \in \mathbb{R}^{d \times T_1}$  is the feature matrix containing all the features of the first view.

The alignment between the two views is found such that the features between the views are maximally correlated. This optimisation is formulated as a least-square problem:



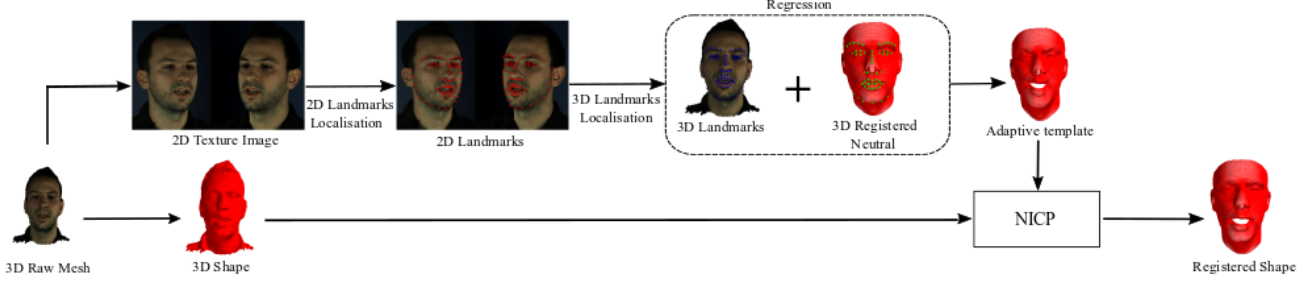


Figure 3. Registration pipeline. The process starts by extracting 2D landmarks from the texture image, and then their corresponding position in the 3D mesh. By applying a regression between the 3D landmarks of the raw mesh and the neutral registered mesh we create an adaptive template. Using this template and the 3D shape of the raw mesh NICP can accurately register the raw mesh.

$$\begin{aligned}
 & \arg \min_{\theta_1, \theta_2} \|f_1(\mathbf{X}_1; \theta_1) - f_2(\mathbf{X}_2; \theta_2)\|_F^2 \\
 & \text{subject to} \quad \begin{aligned}
 & f_1(\mathbf{X}_1; \theta_1) f_1(\mathbf{X}_1; \theta_1)^T = \mathbf{I} \\
 & f_2(\mathbf{X}_2; \theta_2) f_2(\mathbf{X}_2; \theta_2)^T = \mathbf{I} \\
 & f_1(\mathbf{X}_1; \theta_1) f_2(\mathbf{X}_2; \theta_2)^T = \mathbf{D} \\
 & f_1(\mathbf{X}_1; \theta_1) \mathbf{1} = f_2(\mathbf{X}_2; \theta_2) \mathbf{1} = \mathbf{0},
 \end{aligned} \quad (4)
 \end{aligned}$$

where  $f_k(\mathbf{X}_k; \theta_k)$  with  $k \in \{1, 2\}$  represents the output of the two neural networks with parameters  $\theta_k$  and input  $\mathbf{X}_k$ , respectively,  $\mathbf{D}$  is a diagonal matrix, and  $\mathbf{1}$  is an appropriate dimensionality vector of all ones.

We find the optimal parameters for each network with the use of backpropagation. Our problem is a variant of Deep Canonical Correlation Analysis (DCCA) [2], and as such the optimal objective value can be computed as the sum of the  $k$  largest singular values of  $\mathbf{K}_{DCAW} = \Sigma_{11}^{-1/2} \Sigma_{12} \Sigma_{22}^{-1/2}$ , where  $\Sigma_{ij} = \frac{1}{T_2 - 1} f_i(\mathbf{X}_i; \theta_i) \mathbf{C}_T f_j(\mathbf{X}_j; \theta_j)^T$  and  $\mathbf{C}_T = \mathbf{I} - \frac{1}{T_2} \mathbf{1} \mathbf{1}^T$  is the centering matrix. The optimal objective is found by maximising the nuclear norm  $\|\mathbf{K}_{DCAW}\|_* = \text{tr}(\sqrt{\mathbf{K} \mathbf{K}^T})$ , i. e.,

$$\arg \max_{\theta_1, \theta_2} \|\mathbf{K}_{DCAW}\|_* \quad (5)$$

We use gradient ascent to optimise Eq. 5. Since the gradient cannot be computed analytically we use the subgradient [3] by computing the singular value decomposition of  $\mathbf{K}_{DCAW} = \mathbf{U} \mathbf{S} \mathbf{V}^T$ , then the subgradient for the last layer of the network can be defined as follows

$$\mathbf{L}_+ = \Sigma_{11}^{-1/2} \mathbf{U} \mathbf{V}^T \Sigma_{22}^{-1/2} f_2(\mathbf{X}_2; \theta_2) \mathbf{C}_T \quad (6)$$

$$\mathbf{L}_- = \Sigma_{11}^{-1/2} \mathbf{U} \mathbf{S} \mathbf{U}^T \Sigma_{11}^{-1/2} f_1(\mathbf{X}_1; \theta_1) \mathbf{C}_T \quad (7)$$

$$\frac{\partial \|\mathbf{K}_{DCAW}\|_*}{\partial f_1(\mathbf{X}_1; \theta_1)} = \frac{1}{T_2 - 1} (\mathbf{L}_+ - \mathbf{L}_-). \quad (8)$$

Finally, we should point out that DCAW can be extended to handle multiple data sequences by utilising an objective similar to Multi-set CCA, i. e.,

$$\sum_{i,j} \|K_{DCAW}^{i,j}\|_* \quad (9)$$

## 4.2. Word Alignment

We can now use DCAW to compute an alignment path between the speech signal of words uttered by our participant and the corresponding signals of the LRW dataset. To this end, we train a recurrent convolutional neural network for each word (i. e. 500 networks - see Sec. 5 for topology) by fixing one of the views to our participant's speech signal and the other to the corresponding speech signals of LRW, and get the alignment path between them.

Utilising the alignment path of the speech signals, we propagate the 3D shape parameters computed for our participant to the LRW dataset. In the case where an audio frame of the LRW is aligned to multiple frames from our dataset, the mean of the 3D shape parameters of these frames is computed and assigned as the ground truth of that frame. If the opposite holds, namely, an audio frame from our speech signal is aligned to multiple audio frames of the LRW, then the same ground truth is assigned to all LRW audio frames.

By transferring our 3D shape parameters to the LRW dataset, we create a large word-level dataset in-the-wild for 3D dense shape estimation from speech. This allows us to train models that can generalise to every speaker, and at the same time to in-the-wild speech signals.

## 5. Training

Three aspects are relevant to our training: (i) the input representation, (ii) the network topology, and (iii) the objective function utilised for training and evaluating the model. We describe in detail in the rest of the section.

**Input Representation.** All audio signals have sampling rate at 44.1 kHz, and after we remove the DC offset, we

normalise its volume to 0 dB, namely, using the full  $[-1, 1]$  range. No other pre-processing step takes place.

We use mel-spectrograms as our input representation of the audio signal. This representation is appropriate for our task because it is derived by approximating the frequencies perceived by the human cochlea [26]. Thus, the information is similar to the perceived human hearing.

For each visual frame, we derive mel-spectrograms in an audio window of length 400 ms so that we can take into account co-articulation effects in the signal. For each audio window, we compute 128 mel-frequency parameters utilising a window of length 20 ms (882 samples) with 10 ms (441 samples) overlap. Hence, a 2D representation is formed of size  $41 \times 128$ . By calculating its first and second temporal derivatives, and place all three 2D representations in a different channel, we form a  $41 \times 128 \times 3$  representation, which is the input to our convolutional recurrent neural network.

**Network Topology.** Our deep neural network topology is inspired by Karras et al. [16], and is comprised of four parts: (a) frequency extractor, where features are extracted vertically, namely, exploiting the frequency domain of the input representation, with kernel and stride size of 3 and 2, respectively, (b) short term temporal extractor, where features are extracted horizontally, namely, from the temporal domain of the extracted representation of the previous step, with kernel and stride size of 3 and 2, respectively, (c) a non-linear transformer, that non-linearly transform of the convolutional extracted features 128 dimensionality, and (d) a long term temporal extractor represented with a recurrent neural network of 1-layer LSTM cell of 128 dimensions, which captures the long term temporal dynamics in the data. The last layer is a fully connected that produces the 3D shape parameters of our model. The filter size for all convolution layers is set to 64.

**Objective Function.** Most of the studies in the literature use as objective function the Mean Squared Error (MSE). However, we propose to use an objective function that is based on the Concordance Correlation Coefficient ( $\rho_c$ ), which is also used as our evaluation metric. The correlation coefficient evaluates the agreement level between the predictions and the ground truth by scaling their correlation coefficient with their mean square difference. More particularly, for each shape parameter  $i$  we define the concordance loss  $\mathcal{L}_c^i$  between the ground truth  $\mathbf{x}$  and the prediction  $\mathbf{y}$  as follows:

$$\mathcal{L}_c^i = 1 - \rho_c = 1 - \frac{2\sigma_{xy}^2}{\sigma_x^2 + \sigma_y^2 + (\mu_x - \mu_y)^2}, \quad (10)$$

where  $\mu_x = \mathbb{E}(\mathbf{x})$ ,  $\mu_y = \mathbb{E}(\mathbf{y})$ ,  $\sigma_x^2 = \text{var}(\mathbf{x})$ ,  $\sigma_y^2 = \text{var}(\mathbf{y})$  and  $\sigma_{xy}^2 = \text{cov}(\mathbf{x}, \mathbf{y})$ .

For our purposes we train our networks to simultaneously predict all 28 3D shape parameters. We should note that each shape parameter explains different variability percentage in our data, and hence contributes differently to the final 3D reconstructed mesh. We take this fact into account and we add as weight  $w_i$  to each shape parameter concordance loss  $\mathcal{L}_c^i$ , the variability percentage the shape parameter represents. Our overall loss function is defined as  $\mathcal{L} = \sum_{i=1}^{28} w_i \mathcal{L}_c^i$ .

## 6. Experiments

We perform extensive evaluation of our proposed methods by: (i) comparing the DCAW with the current state-of-the-art method for maximally correlating two views and finding an alignment (Sec. 6.2), (ii) comparing the proposed speech blendshapes with the blendshapes of the FaceWarehouse (Sec. 6.3), and (iii) validating our approach for 3D face motion from "in-the-wild" speech (Sec. 6.4).

### 6.1. Experimental Setup

The hyperparameters of the models are kept the same throughout all of the experiments. As our optimisation function, we use the Adam optimiser [18] with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.99$ , and a initial learning rate of  $5 \times 10^{-4}$ , with batch size set to 50. Finally, the initialisation of the feature extraction network was performed following He et al. [15], whereas our recurrent network weights are initialised following Glorot based initialisation [14]. We should note that zero padding was used to samples that do not match the maximum sequence length of the batch. We discard the zero-padded frames that do not belong to the sample by applying a mask.

### 6.2. Deep Canonical Attentional Warping

We compare our proposed DCAW with the state-of-the-art method for representation learning and temporal warping, the Deep Canonical Temporal Warping (DCTW) [34]. The performance of the two methods is evaluated on two datasets: (a) the MMI Facial Expression Dataset [23], which contains more than 2900 videos of 75 different subjects, each performing a particular combination of Action Units (i. e., facial muscle activations). We predict the AU12 and utilise the same approach and network architecture (with a recurrent network on top) as in [34]. (b) We use the LRW dataset to perform template matching, where one of the views is a speech signal of a word and the other is a speech signal of a sentence containing, in a random location, the word. The methods need to accurately find the word in the sentence. For our purposes, we use 10 randomly chosen words with 100 samples.

The performance measure is the alignment error introduced in [41]. More particularly, given  $m$  sequences, each algorithm infers a warping path, i. e.,  $P_{alg} =$

$[p_{alg}^1, \dots, p_{alg}^m] \in \mathbb{R}^{l_{alg} \times m}$ , and the alignment error is computed with the ground truth path  $P_{grd} = [p_{grd}^1, \dots, p_{grd}^m] \in \mathbb{R}^{l_{grd} \times m}$  as follows:

$$\text{Error} = \frac{\text{dist}(P_{grd}, P_{alg}) + \text{dist}(P_{alg}, P_{grd})}{l_{grd} + l_{alg}}, \quad (11)$$

where  $\text{dist}(P_{grd}, P_{alg}) = \sum_{i=1}^{l_1} \min_{j=1}^{l_2} \|p_1^{(i)} - p_2^{(j)}\|_2$ .

Table 1 depicts the results for both experiments. Our method outperforms DCTW in both of them, and for the LRW one we find the result to be statistically significant ( $\alpha < 0.05$ ). These results validate our choice of using DCAW for aligning our participant’s speech signal with the ones from the LRW dataset.

Dataset	DCTW	DCAW
MMI	0.59	<b>0.61</b>
LRW	0.64	<b>0.72</b>

Table 1. Results (wrt the mean alignment error) of the DCAW and DCTW methods on the MMI and LRW datasets.

### 6.3. Blendshapes Comparison

We compare quantitatively and qualitatively the proposed blendshapes with the FaceWarehouse ones, that were used by Pham et al. [25], by testing the generalisation capacity of the blendshapes to represent unseen 3D facial meshes.

More particularly, we compute the error as the per-vertex Euclidean distance between every mesh of the test subject (i.e. not included in the training process) and its corresponding projection to the subspace defined by the blendshapes. An average value is computed over all vertices. For the whole sequence the mean error of the FaceWarehouse is 1.29, and our proposed blendshapes is 0.87. Fig. 4 depicts the error in a sequence of 2,550 frames. It is clear that the proposed blendshapes outperform the FaceWarehouse ones by a high margin. Finally, for qualitative purposes, we show three samples of the original 3D facial shapes and how they are reconstructed by the FaceWarehouse and the proposed blendshapes.

To further demonstrate the generalisation capability of our speech blendshapes, we show our model’s prediction and the ground truth 3D shape parameters on three individuals. Fig. 5, shows the results.

### 6.4. Audio Experiments

We start the validity of our method by performing two kind of experiments using the LRW-3D dataset: (i) *speaker-independent*, where we measure the performance of our model on different speakers that pronounce the same words,

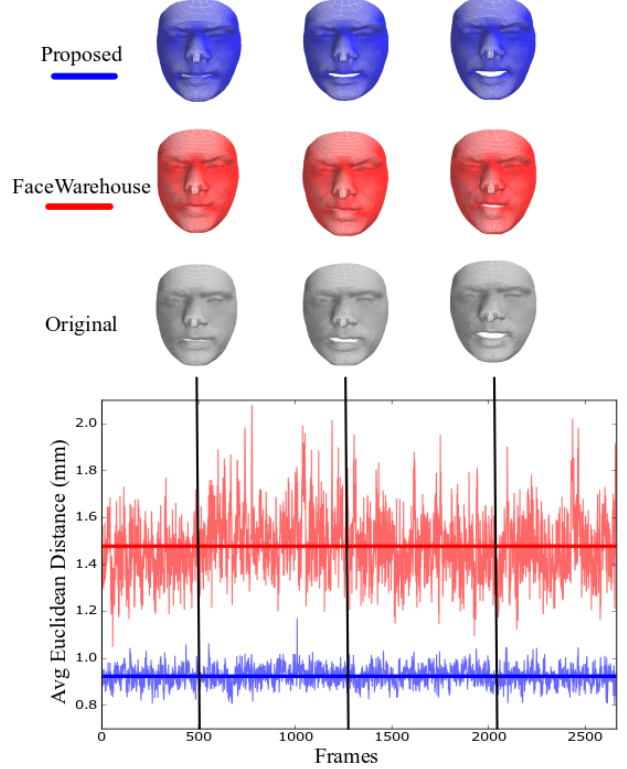


Figure 4. Average Euclidean distance for a sequence of frames between the FaceWarehouse and the proposed blendshapes. - Best viewed in colour.

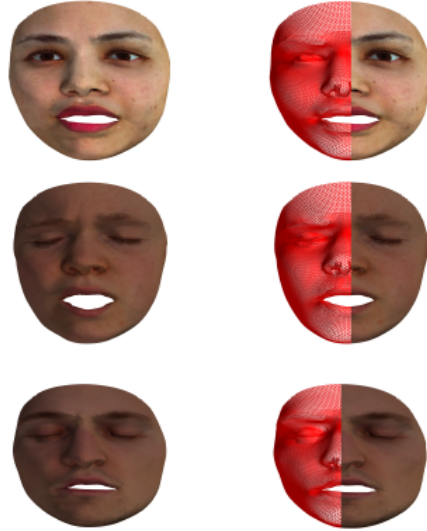


Figure 5. Depicting the ground truth 3D shape parameters (left column) and the predictions (right column) for three individuals. - Best viewed in colour.

and (ii) *word- and speaker-independent*, where we evaluate our model on different speakers that pronounce differ-

Dataset	Methodology	$\mu_{28}$	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$
LRW	Speaker Ind.	.621 (.679)	.643 (.712)	.607 (.684)	.563 (.596)	.652 (.683)	.582 (.641)
	Word/Speaker Ind.	.502 (.554)	.536 (.556)	.582 (.618)	.557 (.624)	.395 (.405)	.411 (.426)
LRS	Continuous Speech	.463	.482	.414	.443	.475	.504

Table 2. Results with respect to  $\rho_c$  for the experiments: speaker independence, word/speaker independence, and sentences. The mean value of the estimation of the 28 3D parameters ( $\mu_{28}$ ), and the first five 3D parameters is depicted. In parenthesis the results on the validation set.

ent words. For both experiments our test set is comprised of one of the participant’s audio signal with the corresponding 3D shape parameters, which are excluded from the training process. Finally, we measure the performance of our model on continuous speech signal.

**Speaker-Independent.** In our first experiment we test the performance of the model when the same words are uttered by different individuals. To this end, we split the 1000 speakers of the dataset to 900 for training, and the rest 100 for validation. Hence, our training set is comprised of 370,000 samples, and our validation set of the rest 80,000. Table 2 depicts the results of the mean, and the first five shape parameters in terms of the  $\rho_c$  metric. The results indicate the validity of our method to generalise to every speaker and in uncontrolled conditions.

**Word- and Speaker-Independent.** Considering the high performance of the model for different speakers, in this experiment we test its performance for different words and speakers. Hence, we split our dataset to a training set that contains 450 words and 900 speakers, and the validation set contains the rest of the 50 words and 100 speakers. In total, the training set contains approximately 355,000 samples, and the validation set approximately 95,000 samples. We should point out that the words that comprise the validation set were chosen such that they contain the same phonemes as the ones in the training set.

To improve the generalisation capacity and reduce overfitting of our model, we perform a random time-segmentation of our training samples. More particularly, each sample in the training batch is randomly segmented from its both ends but always keeping at least 50 % of the frames of the original sample. This is particularly beneficial to our recurrent network architectures as now the temporal dynamics in the training set vary.

Table 2 depicts the results of the mean, and the first five shape parameters in terms of the  $\rho_c$  metric. The performance drops compared to the previous experiment as now the temporal information in the validation set is different from the training one. However, this does not limit the capacity of the model to be able to accurately reconstruct the 3D meshes.

**Sentence-level Experiments.** We also test the performance of our model in continuous speech signals. More specifically, we captured a native speaker (different than the previous experiments) pronouncing 50 sentences (3 to 8 sec

long), taken from the Lip Reading Sentences (LRS) in-the-wild dataset [8], and extracted his 3D parameters. After the extraction of mel-spectrograms from the raw waveform, we feed them to the model and estimate its test performance in continuous speech signals. Table 2 depicts the results of the mean, and the first five shape parameters in terms of the  $\rho_c$  metric. We observe that the performance of our model remains also high in this experiment. We should point out that our model is trained with short temporal dynamics (10 to 30 frames long), and as such it cannot accurately predict 3D shape parameters for longer sequences such as sentences. We tackle this difficulty by splitting the speech signal of the sentences to sequences of 15 frames long and feed them separately our model. We apply a temporal filter on the predictions of our model, to remove temporal discontinuities added by the LSTM. In the supplementary material videos are provided that show the effectiveness of our method.

## 7. Conclusions

We presented a methodology for constructing 3D facial meshes from speech cues captured in uncontrolled conditions. More particularly, we learned a statistical blendshape model by capturing 4D sequences of people uttering 500 words selected from the Lip Reading Words (LRW) in-the-wild dataset. To align the words uttered from our participant with the words of the LRW, we proposed Deep Canonical Attentional Warping (DCAW), a novel method that simultaneously learns deep representations and an alignment path between two sequences. We thoroughly experimented with our proposed methods and showed the ability of a trained deep learning model on the create LRW-3D to generalise to different speakers and in uncontrolled conditions of speech.

For future work we intend to incorporate expression in our blendshapes such that accurate emotional speech can be obtained. In addition, we will test our model on languages different than English. On top of that, we will capture individuals talking on different languages to expand the generalisation capacity of our blendshapes.



## References

- [1] B. Amberg, S. Romdhani, and T. Vetter. Optimal step non-rigid icp algorithms for surface registration. In *Computer Vision and Pattern Recognition*, pages 1–8, 2007. 3
- [2] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *International Conference on Machine Learning*, pages 1247–1255, 2013. 5
- [3] F. R. Bach. Consistency of trace norm minimization. *Journal of Machine Learning Research*, 9(Jun):1019–1048, 2008. 5
- [4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 4
- [5] J. Booth, A. Roussos, A. Ponniah, D. Dunaway, and S. Zafeiriou. Large scale 3d morphable models. *International Journal of Computer Vision*, 126(2-4):233–254, 2018. 3
- [6] J. Booth, A. Roussos, S. Zafeiriou, A. Ponniah, and D. Dunaway. A 3d morphable model learnt from 10,000 faces. In *Conference on Computer Vision and Pattern Recognition*, pages 5543–5552, 2016. 3
- [7] S. Cheng, I. Kotsia, M. Pantic, and S. Zafeiriou. 4dfab: A large scale 4d database for facial expression analysis and biometric applications. In *Conference on Computer Vision and Pattern Recognition*, pages 5117–5126, 2018. 2, 4
- [8] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman. Lip reading sentences in the wild. In *Conference on Computer Vision and Pattern Recognition*, pages 3444–3453, 2017. 8
- [9] J. S. Chung and A. Zisserman. Lip reading in the wild. In *Asian Conference on Computer Vision*, pages 87–103, 2016. 3
- [10] D. Cosker, E. Krumhuber, and A. Hilton. A faces valid 3d dynamic action unit database with applications to 3d dynamic morphable facial modeling. In *International Conference on Computer Vision*, pages 2296–2303, 2011. 3
- [11] P. Edwards, C. Landreth, E. Fiume, and K. Singh. Jali: an animator-centric viseme model for expressive lip synchronization. *Transactions on Graphics (TOG)*, 35(4):127, 2016. 1, 2
- [12] M. Elsner and C. Shain. Speech segmentation with a neural encoder model of working memory. In *Conference on Empirical Methods in Natural Language Processing*, pages 1070–1080, 2017. 4
- [13] B. Fan, L. Wang, F. K. Soong, and L. Xie. Photo-real talking head with deep bidirectional lstm. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4884–4888, 2015. 2
- [14] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010. 6
- [15] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *International Conference on Computer Vision*, pages 1026–1034, 2015. 6
- [16] T. Karras, T. Aila, S. Laine, A. Herva, and J. Lehtinen. Audio-driven facial animation by joint end-to-end learning of pose and emotion. *Transactions on Graphics (TOG)*, 36(4):94:1–94:12, 2017. 1, 3, 6
- [17] A. Katsamanis, G. Papandreou, and P. Maragos. Face active appearance modeling and speech acoustic information to recover articulation. *Transactions on Audio, Speech, & Language Processing*, 17(3):411–422, 2009. 2
- [18] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
- [19] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015. 4
- [20] A. D. Marshall, P. L. Rosin, J. Vandeventer, and A. Aubrey. 4d cardiff conversation database (4d ccdb): A 4d database of natural, dyadic conversations. pages 157–162, 2015. 2
- [21] W. Matthyses and W. Verhelst. Audiovisual speech synthesis: An overview of the state-of-the-art. *Speech Communication*, 66:182–217, 2015. 2
- [22] A. Myronenko and X. Song. Point set registration: Coherent point drift. *Transactions on Pattern Analysis & Machine Intelligence*, 32(12):2262–2275, 2010. 3
- [23] M. Pantic, M. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. In *International Conference on Multimedia and Expo*, pages 5–pp, 2005. 6
- [24] A. Patel and W. A. Smith. 3d morphable face models revisited. In *Computer Vision and Pattern Recognition*, pages 1327–1334, 2009. 3
- [25] H. X. Pham, Y. Wang, and V. Pavlovic. End-to-end learning for 3d facial animation from speech. In *International Conference on Multimodal Interaction*, pages 361–365, 2018. 1, 2, 3, 7
- [26] L. R. Rabiner and R. W. Schafer. *Theory and applications of digital speech processing*, volume 64. 6
- [27] C. Raffel, M.-T. Luong, P. J. Liu, R. J. Weiss, and D. Eck. Online and linear-time attention by enforcing monotonic alignments. In *International Conference on Machine Learning*, pages 2837–2846, 2017. 4
- [28] S. Sako, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Hmm-based text-to-audio-visual speech synthesis. In *International Conference on Spoken Language Processing*, 2000. 2
- [29] G. Salvi. Using hmms and anns for mapping acoustic to visual speech. 1999. 2
- [30] G. Salvi, J. Beskow, S. Al Moubayed, and B. Granström. Synface: speech-driven facial animation for virtual speech-reading support. *EURASIP Journal on Audio, Speech, and Music Processing*, 2009:3, 2009. 2
- [31] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *Transactions on Graphics (TOG)*, 36(4):95, 2017. 3
- [32] S. Taylor, T. Kim, Y. Yue, M. Mahler, J. Krahe, A. G. Rodriguez, J. Hodgins, and I. Matthews. A deep learning approach for generalized speech animation. *Transactions on Graphics (TOG)*, 36(4):93, 2017. 2
- [33] S. L. Taylor, M. Mahler, B.-J. Theobald, and I. Matthews. Dynamic units of visual speech. In *Special Interest*

*Group on Computer GRAPHics and Interactive Techniques-Eurographics Symposium on Computer Animation*, pages 275–284, 2012. 2

- [34] G. Trigeorgis, M. A. Nicolaou, B. W. Schuller, and S. Zafeiriou. Deep canonical time warping for simultaneous alignment and representation learning of sequences. *Transactions on Pattern Analysis & Machine Intelligence*, (5):1128–1138, 2018. 6
- [35] A. Wang, M. Emmi, and P. Faloutsos. Assembling an expressive facial animation system. In *Special Interest Group on Computer GRAPHics and Interactive Techniques symposium on Video games*, pages 21–26, 2007. 2
- [36] L. Xie and Z.-Q. Liu. Realistic mouth-synching for speech-driven talking face using articulatory modelling. *Transactions on Multimedia*, 9(3):500–510, 2007. 2
- [37] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3d dynamic facial expression database. In *Automatic Face & Gesture Recognition, 2008. FG’08. 8th IEEE International Conference on*, pages 1–6. IEEE, 2008. 2
- [38] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. A 3d facial expression database for facial behavior research. In *Automatic face and gesture recognition*, pages 211–216, 2006. 2
- [39] S. Zafeiriou, G. G. Chrysos, A. Roussos, E. Ververas, J. Deng, and G. Trigeorgis. The 3d menpo facial landmark tracking challenge. In *International Conference on Computer Vision*, pages 2503–2511, 2017. 3
- [40] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 3
- [41] F. Zhou and F. De la Torre. Generalized time warping for multi-modal alignment of human motion. In *Conference on Computer Vision and Pattern Recognition*, pages 1282–1289, 2012. 6
- [42] Y. Zhou, Z. Xu, C. Landreth, E. Kalogerakis, S. Maji, and K. Singh. Visemenet: Audio-driven animator-centric speech animation. *Transactions on Graphics (TOG)*, 37(4):161:1–161:10, 2018. 1, 2