A Robust Interactive Facial Animation Editing System

Eloïse Berson
Dynamixyz
CentraleSupélec, CNRS, IETR, UMR 6164, F-35000
Rennes, France
eloise.berson@dynamixyz.com

Catherine Soladié
CentraleSupélec, CNRS, IETR, UMR 6164, F-35000
Rennes, France
catherine.soladie@centralesupelec.fr

Vincent Barrielle
Dynamixyz
vincent.barrielle@dynamixyz.com

Nicolas Stoiber
Dynamixyz
nicolas.stoiber@dynamixyz.com

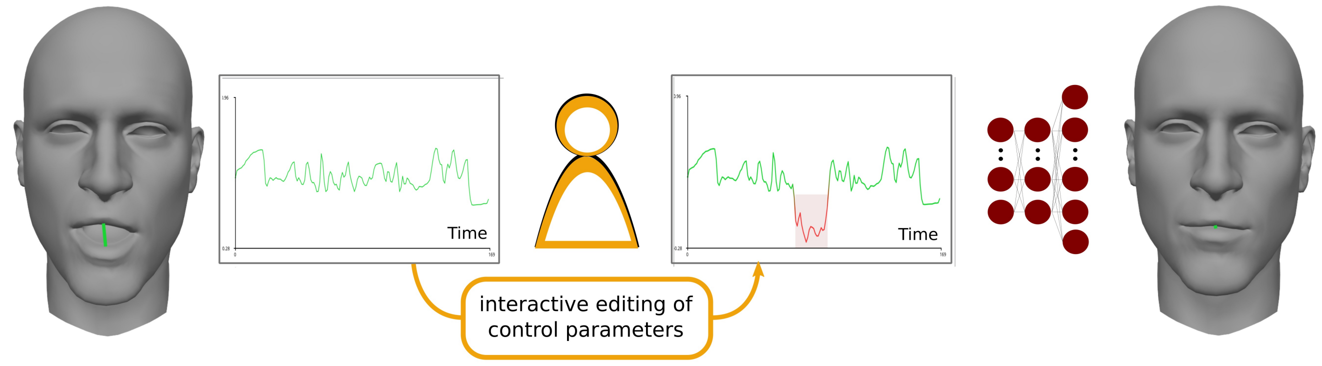

Figure 1: Editing pipeline. An interactive interface enables the user to easily edit meaningful control parameters that are automatically mapped to a realistic facial animation.

ABSTRACT

Over the past few years, the automatic generation of facial animation for virtual characters has garnered interest among the animation research and industry communities. Recent research contributions leverage machine-learning approaches to enable impressive capabilities at generating plausible facial animation from audio and/or video signals. However, these approaches do not address the problem of animation edition, meaning the need for correcting an unsatisfactory baseline animation or modifying the animation content itself. In facial animation pipelines, the process of editing an existing animation is just as important and time-consuming as producing a baseline. In this work, we propose a new learning-based approach to easily edit a facial animation from a set of intuitive control parameters. To cope with high-frequency components in facial movements and preserve a temporal coherency in the animation, we use a resolution-preserving fully convolutional neural network that maps control parameters to blendshapes coefficients sequences. We stack an additional resolution-preserving animation autoencoder after the regressor to ensure that the system outputs natural-looking animation. The proposed system is robust and can handle coarse, exaggerated edits from non-specialist users. It also retains the high-frequency motion of the facial animation. The training and the tests are performed on an extension of the B3D(AC)ˆ2 database [10], that we make available with this paper at http://www.rennes.centralesupelec.fr/biwi3D.

CCS CONCEPTS

• **Computing methodologies** → **Motion processing**; *Neural networks*; • **Human-centered computing** → *Graphical user interfaces.*

KEYWORDS

Facial animation, interactive motion edition, learning-based approach, dataset

ACM Reference Format:
Eloïse Berson, Catherine Soladié, Vincent Barrielle, and Nicolas Stoiber. 2019. A Robust Interactive Facial Animation Editing System. In *Motion, Interaction and Games (MIG '19), October 28–30, 2019, Newcastle upon Tyne, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3359566.3360076

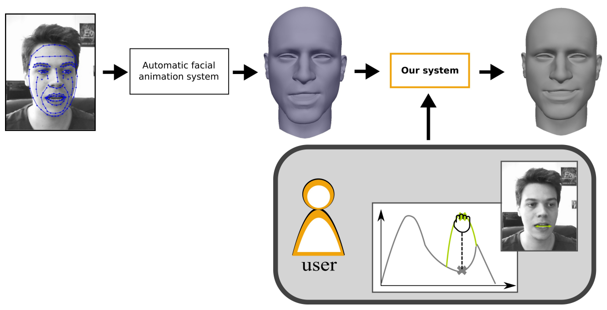

Figure 2: System overview. Our editing system allows a non-specialist user to easy and quickly interfere in the traditional facial animation pipeline to refine an animation.

1 INTRODUCTION

Producing a convincing facial animation for a virtual character is a tedious and time-consuming task which requires talent and experience, both rare resources. Many works have been conducted in the field of animation research to come up with automatic facial animation generation systems that would accelerate this process. Traditionally, automatic animation systems consist in an end-to-end pipeline, taking as input either audio [34], image or text [29], and generating a full sequence of animation data (typically as a sequence of blendshapes coefficients). Although impressive results have been reached, it is common to require human intervention to refine this baseline animation, either to correct mistakes or to make some adjustment to adapt the motion to another artistic intent. These modifications are done by an animator trained in manipulating low level 3D animation parametrization such as blendshapes coefficients. This edition pipeline implies technical and artistic skills, as well as a considerable amount of time to end up with a coherent and satisfactory final animation.

In this paper, we present an editing tool that allows non-specialist users to easily and quickly refine an existing animation. The main challenge in facial animation is to ensure the naturalness of the motion. Indeed, the human are experts at observing faces and can detect even subtle implausible movements, notably the missing of lip contact when the mouth closes during a speech. We develop a machine-learning-based approach that learn from a dataset, and trains to produce natural-looking animation from a small set of input parameters. By training on natural animation space-time patterns, our system learns to preserve the temporal coherency of the motion and ensure smooth and continuous animation. As contributions such as Seol et al. [27], our system is designed to be efficiently integrated in the traditional facial animation pipeline as shown in Figure 2. However, unlike Seol et al. [27] who focuses on producing an efficient system dedicated to a professional use, our goal is to provide an alternative solution for non-specialist users. We specifically design our system to be robust to inadequate user edits, and handle exaggerate or conflicting inputs. Besides, instead of complicated facial control parameterizations we propose to use intuitive high-level control parameters as input to the system, such as specifying the distance between the lips over time. The system runs at low latency, enabling us to create a graphical interface for users to interactively modify the output animation until getting a satisfying result.

One challenge when dealing with facial animation is to preserve the high-frequency patterns of the motion, as they are responsible for important communication cues (eye closures, lip contacts). This is particularly true for learning-based solutions, that leverage large datasets of complex, possibly conflicting animation patterns [13]. Among the shortcomings of these solutions is the ability to preserve the different frequency components of the animation and to adapt the behavior of the system to inconsistent inputs. In this work, we define an architecture based on a fully convolutional network with skip layers designed specifically to preserve high-frequency components. Besides, we aim at a system that is resilient to coarse editing by non-specialist users. To that end, we train an additional denoising autoencoder that we stack at the end of the network to ensure a natural-looking final animation output.

In order to be a suitable solution for non-expert users to create powerful facial animation pipelines, an editing tool has to meet the following requirements: 1. Usability: a user should be able to personalize a 3D animation without advanced animation skills. The number of control parameters should be small, and those should be meaningful and easy to manipulate. The system should run fast enough to enable interactive editing. The user can then iteratively modify its animation, either by editing a few frames or by imposing full-sequence constraints, until a satisfying result is produced (see Section 5.4). 2. Plausibility: the complex space-time patterns of human facial motion should be respected (see Section 5.1). In particular, high-frequency facial movements should be present. 3. Robustness: the final animation should remain plausible regardless of the user modifications (see Section 5.3). 4. Subject and content independent: any type and style of facial animation should be able to be edited (see Section 5.2). In the following sections, we describe a learning-based editing system that addresses all these requirements.

Our machine-learning system relies on a dataset of facial animation sequences. To train the full system, we worked on modifying and extending the existing 3D facial animation dataset B3D(AC)ˆ2 database [10]. We will release the extended dataset for reproducibility of our results.

Our contributions are:

- A new facial animation editing system based on convolutional neural networks, which enables to quickly edit a temporal talking facial animation with few intuitive control parameters. Based on a time resolution-preserving architecture, our system is capable of generating complex and plausible facial motion pattern. The proposed framework features a regressor designed to map low dimensional control parameters to blendshapes coefficients sequences. It is followed by an autoencoder meant to ensure the naturalness of the outputted animation sequences.
- A robust solution dedicated to non-specialist users that is resilient to implausible inputs constraints. We use a denoising training strategy to improve the reliability of our system. The originality comes from the indirect noisy inputs used to train the stacked autoencoder, and an additional loss term

encouraging mouth closure preservations during talking facial animations.

- The release of our enhanced 3D audiovisual animation database from Fanelli et al. [10] with notably a parameterization of the animations with the widely-used blendshapes formalism. With the use of a professional software we added 2D eyelids and mouth annotations and improved the overall quality of the animation and the depiction of cues such as eyelids and lip contacts.

2 RELATED WORK

In the professional world, animation edition is done by directly manipulating the temporal curves of complex facial parameterizations (blendshapes coefficients being an industry-standard parametrization). Hence, traditional animation production requires animators with technical and artistic skills, experience and remains time-consuming even for those. Previous works have addressed this problem by providing efficient animation editing systems either based on geometry-driven approaches or data-driven approaches.

2.1 Geometric animation edition

Early facial expression manipulation approaches are based on key frames edition. The key frames can be made of linear combinations of face meshes coming from a pre-captured database [6, 15, 32, 33]. The goal is to find the blending weights corresponding to user constraints. In these approaches, the user manipulates 2D control points which are either image features [33], motion markers [15] or the 2D projection of 3D vertices [6, 32]. Other works consider the key frame editing problem as solving the 3D vertices position of the edited mesh. To reduce the dimension of the facial model, Lau et al. [17] used PCA to obtain a subspace representation. Then, they derived the 3D face vertices positions from strokes, points or curves constraints drawn by the users on a 2D screen. An alternative to PCA to obtain a semantically meaningful data representation is the Independent Component Analysis (ICA) [4, 23]. This parametrization gives the possibility to distinguish between editing facial emotional components and speech-related components. More recent works develop editing systems that can be easily integrated in the animation pipeline. The animation editing problem consists in finding the underlying blendshapes coefficients. The users directly manipulate the vertices of the mesh [2, 18, 30] or draw 2D strokes on a screen [5]. The number of users constraints is generally smaller than the blendshapes model parameters, the optimization problem is thus regularized through different criteria: by constraining the value of the blendshapes coefficients [18], by using a statistical model [2, 31], by constructing an orthogonal blendshapes model [21], by using geometric constraints [25] or by adding face areas boundary constraints [30]. To improve the applicability of the edited method, most of the previous works segment the face into hierarchical regions a priori [15, 21, 23, 33], or a posteriori using an influence map for each control points [6, 32], or else using the ICA transform for decorrelation [4].
These methods are mainly frame-based and most of them do not consider the temporal consistency of movements.

To overcome that limitation, there are works dealing with the dynamic nature of animation data, instead of performing edition on static expressions. Li and Deng [21] propose sequence edition by fitting a Catmull-Rom spline on the edited blendshapes weight sequences. This technique does not necessarily preserve the naturalness of the motion, as nothing encourages the motion to be physically correct. Inspired from space-time constraints body motion systems [11], Ma and colleagues [23] create a style learning editing framework. The editing of style is applied to similar frames in the sequence. While it is an efficient solution to reduce time spent in the animation editing process, this solution does not ensure temporal coherency. Applying the same edit to similar frames with a different context leads to inconsistent motion. Seol et al. [27] and Akhter et al. [1] propose a temporal solution to propagate the edits across the surrounding frames by solving a movement matching equation or by using a spatiotemporal bilinear model. Although, these methods provide smooth results, their temporal resolution depends on hyperparameters that need to be manually adjusted rendering the editing task more difficult to tune. Moreover, the system of Seol et al. [27] is not robust to inconsistent users edits. For example, in the case of exaggerated user constraints, this method generates implausible animations. As we target non-specialist users, our system needs to ensure the final motion to be realistic.

2.2 Data-based animation edition

In contrast to keyframe-based geometric editing methods, space-time methods consider the manipulation of entire temporal motion data-blocks. An effective technique to perform temporal coherent edition and generation is motion graphs [16, 32]. This technique consists in building a graph where nodes encode static poses or short-term motion blocks. The graph can be navigated to recreate plausible animation sequence. The edges between nodes encode the likelihood of the transition between those two blocks being plausible, so realistic animation reconstruction consist in finding paths of minimal cost in the graph. While motion graph is a relevant technique for our purpose, it imposes high memory usage as it requires retaining the whole graph for inference. Moreover, a balance has to be achieved between expressivity, which can be obtained by a graph with a large number of connections, and physical consistency, which is better enforced with a sparser graph featuring only consistent transitions.

More recent works in the line of data-based method have adopted new models for space-time human motion editing systems. The first one to propose a fully learning-based human motion editing system is the seminal work of Holden et al. [13]. They map high level control parameters to a learned body motion manifold presented earlier by the same authors [14]. Navigating this manifold of body motion allows to easily alter and control body animations, while preserving their plausibility. Recently, Habibie and colleagues [12], as well as Martinez and coworkers [24], designed state-of-the art dynamic motion modeling systems, demonstrating the high potential of learning-based approach in human motion manipulation.

Closest to our work, tackling the same challenge of editing an animation using simple high level parameters, is the work of Holden et al. [13]. Unlike body motion however, facial motions have a lot of high-frequency temporal components such as blinking, lips

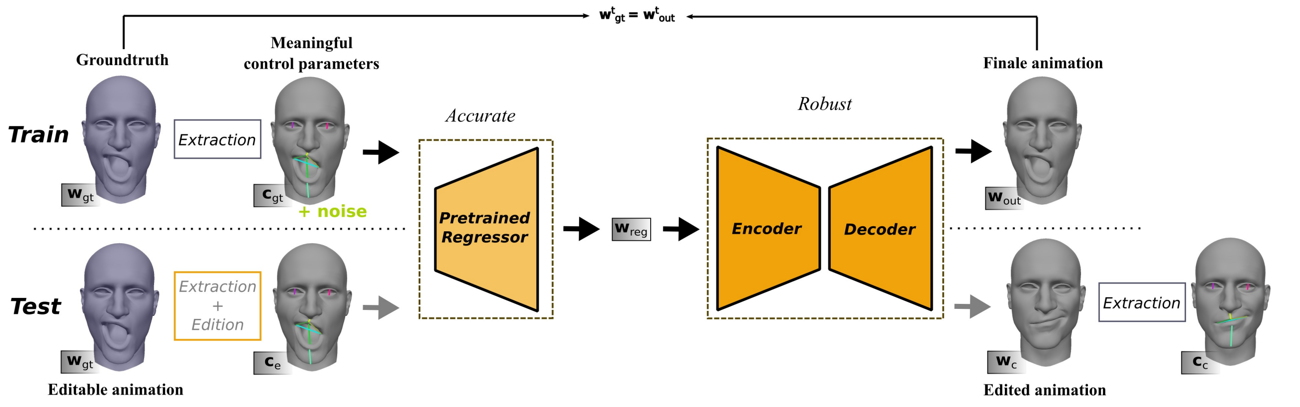

Figure 3: System description. (Top) At train time, fixing the parameters of the regressor, the autoencoder learns to reconstruct the initial blendshape weights from the noisy meaningful control parameters. (Bottom) At test time, the edited control parameters lead to an accurate blendshapes weights sequence thank to the regressor. The stacked autoencoder allows inaccurate edition ensuring a realistic edited animation.

synchronization, and mouth closures. Although the system of [13] demonstrated impressive results on body motion, we found that their architecture is not particularly suited to this particular aspect of facial animation. Using it in our scenario leads to over-smoothed, unappealing facial animations, which we illustrate in Section 5. We therefore adapt this approach for the purpose of facial motion, and tackle the high-frequency issue using a resolution-preserving neural network. Our work is based on a one dimensional fully convolutional network inspired from Ronnenberger and colleagues [26], with skip connections between the down-sampling and the up-sampling parts in order to preserve high-frequency details. To the best of our knowledge, we are the first to study a temporal editing system based on a resolution-preserving neural network.

The remainder of the paper is organized as follows. In Section 3, we will present the dataset work we have conducted, which enabled us to train and test our model. We then describe our model in Section 4. We focus particularly on the benefit of the added autoencoder and the specific way of training it. We compared our system with related works in Section 5 and conducted several experiments highlighting the performance and benefits of our architecture. Finally, we demonstrate the usability of our framework in a realistic animation production pipeline.

3 DATASET PREPARATION

For our experiments, we use the 3D Audio-Visual Corpus of Affective Communication [10], which contains 3D scans and RGB images of 14 actors reciting 40 sentences with and without emotion. However, the quality of scan data prevents us from having a good depiction of subtle mouth closures and blinking. Those are however crucial to verbal and non-verbal communicational cues that facial animation convey. Besides, in order to integrate this tool to the traditional facial animation pipeline, we want facial expressions to be encoded through the standard blendshapes parametrization, which will be used as input to our system.

We address the above issues by fitting a common deformable template, with a sparser mesh, to the neutral geometry of each actor. Then, we transfer a blendshape model onto the aligned deformable template.

The alignment consists of three stages. First, we align a 3D morphable model [3] with the neutral mesh of each actor using a non-rigid ICP algorithm, optimizing the pose and the identity coefficients. Inspired by Li et al. [20], we improve the quality of the alignment around the mouth and eyelids using 2D image landmarks information for each frame, obtained with a commercial face tracking software [9], still optimizing for pose and identity coefficients. We further refine the process by optimizing the vertices directly, through a non-rigid ICP with Laplacian prior [19]. Then, using deformation transfer [28], we transfer a pre-existing blendshape model sharing the topology of the morphable model onto the deformed mesh. At this point, we obtain a subject-specific blendshapes model for the 14 actors. Finally, we derive our final dataset of 29 blendshapes animation weights by fitting the model on the tracked 2D landmarks in each frame of each actors' video performance using [9].

4 SYSTEM DESCRIPTION

In this section, we describe our facial animation editing system in more detail. First, we justify the choice of the control parameters which constitutes the input of our system. Then, we elaborate on the structure of the neural network that forms the heart of our system. The network is composed of two parts. The first part is a regressor which maps high-level inputs to a blendshapes weights sequences. The second one is a stacked autoencoder that cleans the blendshapes weights sequence to ensure a realistic final animation. Both are fully convolutional, and operate on space-time signals, meaning they perform temporal convolutions on a time window of their input parameters.

4.1 Meaningful high-level control parameters

We aim at a system that takes intuitive high-level parameters as input, for users to easily translate their desired modifications into animation. Particularly important in facial animation is the rendition of speech, so we want the control parameters to be able to specify all plausible mouth shapes that occur during natural speech. These parameters have to be simple and meaningful to be intuitively manipulated by non-professional users. Thus, we choose eight inter-vertex distances shown in Figure 4 as our control parameters. The horizontal and vertical inner-lips distances as well as the eyelids distances determine the state of mouth/eye closure, two important expressive cues. To enable editing the emotional expressiveness of the animation such modifying the smile intensity, we add the distance between the upper-lip center and the mouth corners. The lips protrusion, activated by pronouncing palate sounds such as "sh or ch" or doing a kiss shape are manipulated with the distances between the nose bridge and the upper-lip center and between the chin and the bottom-lip center. We found this to be a rather minimal set for our approach. Less parameters would result in ambiguous specifications for face shape, leading to a noisy regressor output. In this work we always measure those distances on a blendshapes-based character with fixed morphology. This ensures that the distance patterns we extract from the dataset's animations of section 3 are actor-independent.

While our network can learn full-face motion patterns, we found that generalization of the results is improved if we split the facial controls in three groups that exhibit low motion correlation with each other in the database: lower-face, upper-face and eyelids. An independent network will be trained for each group, with its own relevant high-level control parameters as input, and appropriate blendshapes coefficients as output. This splitting of the face is common in previous research works and practical applications [15, 32].

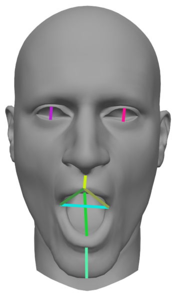

Figure 4: Eight meaningful control parameters extracted from the mesh.

4.2 Regression from low dimension control parameters to blendshapes weights

Motivated by the observation that facial animation is composed of high-frequency features, we moved away from previous motion-modeling network architectures and built a resolution-preserving neural network to regress the control parameters ($\mathbf{c}_{gt}$) to blendshapes weights $\mathbf{w}_{reg}$ as shown in Figure 3. The value for control parameters have been calculated on a fixed morphology character, animated with the blendshapes weights ($\mathbf{w}_{gt}$) extracted from the database.

The regressor is a fully one-dimensional convolutional neural network with skip layers, a structure sometimes loosely described as U-net. Its architecture is depicted in Figure 5a. We use one-dimensional max-pooling layers and up-sampling layers to respectively down-sample and up-sample the temporal dimension. Each convolutional block in Figure 5 is composed of a batch normalization layer, a convolutional layer and the elu activation function [7]. As input to the regressor we use a time-window of 64 frames. We extract those windows from complete sequences with a time-overlap ratio of 0.75. As preprocessing, we subtract the mean controller values calculated on the whole trainset. All the filters in the network have a size of 3. Our loss function is composed of two terms [13]: the mean square error (MSE) between the $\mathbf{w}_{gt}$ and $\mathbf{w}_{reg}$, $\mathcal{L}_{MSE}$, and a L2 regularization on the weights $\beta * \mathcal{L}_{reg}$. We set the trade-off parameter β equals to 1. We employ the Adam optimizer for training with a batch size of 128 and an initial learning rate of 0.001 with a decay ratio of 0.95 every five consecutive epochs with no validation loss improvement.

We use sequences from 13 subjects of the dataset to train our network. This amounts to around 85 minutes of facial animation, which we split into a training set and a validation with a 0.95 ratio. The final state of the network we conserve is the epoch the lowest validation loss.

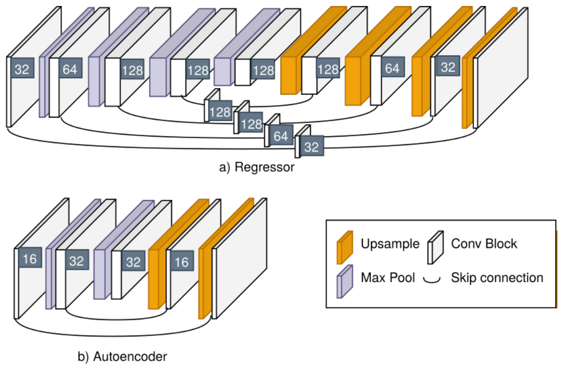

Figure 5: Architecture of the regressor (a) and the autoencoder (b)

4.3 Autoencoder for ensuring the naturalness of the animation

Our network features an animation autoencoder whose role is to clean-up the output of the regressor. Our regressor is a rather straightforward mapping network, so it will faithfully transcribe any user command, easily extrapolating to cases of unrealistic facial animation. However, we opted for a robust system, which would unsure staying in a realistic animation space no matter the user input. The added autoencoder serves that purpose. Its architecture is depicted in Figure 5b.

Ensuring that the network produces realistic animation is due to both the presence of the autoencoder and to the following denoising training strategy. Training autoencoders as denoisers -meaning feeding them with noised inputs and clean outputs- is common practice, but we found that the resulting autoencoder is very dependent on the noise characteristics. In our case, since the noise is supposed to mimic unrealistic user inputs, we found it difficult to find a good noise model. Instead we chose to train the whole end-to-end system as a denoiser, while keeping the regressor weights constant (except the statistics of the batch normalization layers) and optimize the autoencoder's weights to reconstruct $\mathbf{w}_{gt}$. In practice, we modify around 20% of the control parameter inputs of the regressor $\mathbf{c}_{gt}$ (see Figure 3) with salt-and-pepper noise. We found that this creates noisy animation patterns for the autoencoder to train to clean-up that are closer to what the system would encounter in a real run-time scenario. For the autoencoder to conserve the high-frequency features of the regressed output we use a convolutional architecture similar to that of the regressor (Figure 5).

The blendshapes parameterization is not the most representative of the importance of each movement they encode. Movements such as mouth openings/closures carry more expressive and communicational weight than others such as nose movements. The loss that our network learns to minimize should reflect this aspect. To the MSE loss on all blendshapes coefficients we therefore add a loss on the difference between some intervertices distances on the model animated with $\mathbf{w}_{gt}$ and $\mathbf{w}_{out}$.

$$\mathcal{L} = \mathcal{L}_{MSE} + \alpha * \mathcal{L}_{distance} \quad (1)$$

Typically, $\mathcal{L}_{distance}$ measures the distances between the lips, and between the eyelids. This term helps ensuring an accurate mouth closure during a talking facial animation [22]. For our experiments, the parameter α is set to 1. Training the model takes less than 2 hours on a NVIDIA GeForce GTX 1070 GPU.

5 EXPERIMENTS & RESULTS

In this section, we present experimental results of our facial animation editing system. First, we evaluate our system by comparing its integrity to the recent related work of [13], which addresses a similar set of requirements, albeit for body animation applications. We retained their system's architecture, adapting it for our specific inputs and outputs. The discrepancy between quantitative measures and qualitative look of the animations lead us to use special metrics for a more complete comparison (Section 5.1). This comparison confirms the suitability of the proposed neural network system for the purpose of facial animation, as well as its capacity to create plausible facial animation preserving the complex dynamic of the facial movements.

To assess the data-dependency and reproducibility of our system, we apply it on a different recently released database (see Section 5.2) and measure quantitative performance. In Section 5.3, we study the robustness of our system to implausible user constraints, and analyze the role of the system's components. Finally, as our system runs with low latency, we demonstrate in Section 5.4 its potential as an interactive animation tool by showing examples of edition performed on animations resulting of facial tracking.

5.1 Comparison with state-of-the-art approach

Our system is designed for animation edition and control, but it will only be useful if its architecture can handle and represent

Table 1: Quantitative comparison between the regressor and the full system on the test set.

	MSE (lower face)	MSE (eyelids)
Regressor only	0.0028	0.0064
Holden et al. [13]	0.004	0.009
Our system	0.0082	0.0086

sufficiently varied facial motion. Of particular interest is the ability to preserve the high-frequency components of facial animation, which are important for human communication. In practice, we evaluate how close the generated animation $\mathbf{w}_c$ is to ground-truth $\mathbf{w}_{gt}$ when the edited control parameters $\mathbf{c}_e$ are kept unchanged, equal to $\mathbf{c}_{gt}$ (see Figure 3). We evaluate this metric on the whole database using the leave-one-subject-out strategy.

To our knowledge, there is no work directly addressing the problem of high-level, temporal consistent manipulation of facial animation. In the broader field of animation research, Holden et al. [13] set to tackle a similar set of goals for body animation editing and control. Part of their system is valid for facial animation and can be adapted to our inputs and outputs.

To represent their approach, we first learn a time-convolutional autoencoder with one layer to encode the sequence animation into a latent space and one layer to decode. Then, we learn a fully convolutional network to regress the control parameters to this latent space (see [13] for the details). The regressor is built with only 2 layers as it appeared to give better results in our case. To get a fair comparison, we train one such system by face area, similar to ours (see Section 4.1).

We evaluate the different systems by minimizing the mean square error (MSE) between the input and the output blendshapes weights sequences. For our experiments, we use the regressor with the lowest MSE because the role of the regressor is to accurately regressed the control parameters to the blendshapes weights.

Interestingly, Table 1 shows that Holden et al. [13] performs better than our complete system in term of MSE. However, by looking at the temporal curves of inner lips distance derived from $\mathbf{c}_{gt}$ and ($\mathbf{c}_c$), we realize that their system smooths the motion signal and shows consequent loss of high-frequency components of the mouth and the eyes (Figure 6). While the reconstruction MSE is lower, the corresponding animation is qualitatively less appealing as it misses the key high-frequency communicational cues on the mouth and eyelids. Note that this behavior was probably less an issue in their original application on body animation, as high-frequency components carry less semantic weight in that case as it does for facial motion. In Figure 7, we display two frames extracted from sequences created from the same $\mathbf{c}_{gt}$ with the system of Holden et al. [13] and our system. We can see that, while our system produces an animation with faithful mouth openings and closures, the animation resulting of their system misses these cues due to the smoothing nature of their architecture. Examples of animations using both systems are shown in the supplementary video.

For a more representative quantitative comparison between our system and Holden et al. [13], we propose using a metric that highlights the capacity to accurately retain facial animation cues such as

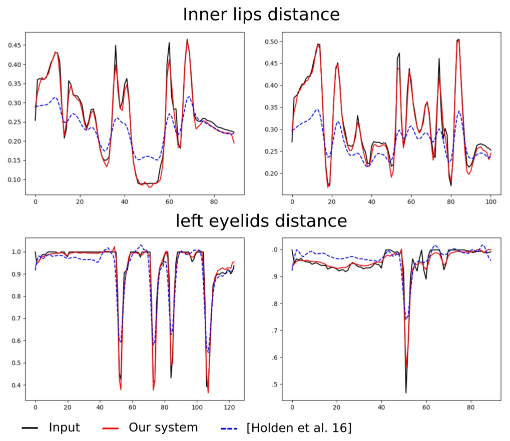

Figure 6: Comparison with Holden et al. [13]: Curves of inner lips distance for different sequences. The body motion system [13] smoothes the output signal loosing the high frequency components.

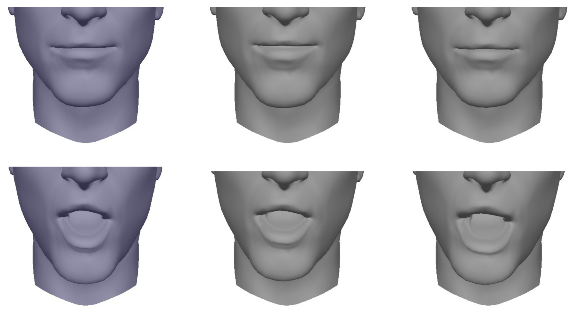

Figure 7: The groundtruth (left). Compare to [13] (middle), our system (right) is able to generate an animation which faithfully respects the input mouth movements and its amplitude.

mouth contacts, closures and eye blinks. To our knowledge, there is no agreed-upon metric in the community for such semantic facial cues, so we suggest measuring a true positive rate (TPR), i.e. ratio of true positive mouth- (respectively eyelid-) closures to the number of actual mouth- (eyelid-) closures, and the false positive ratio (FPR) defined as the ratio of false positive mouth- (eyelid-) closures to the actual mouth- (eyelid-) closures. The TPR measures the capacity of the system to accurately preserve the desired mouth- and eye-related conversational cues. The FPR controls that the system does not hallucinate undesired such movements. On Figure 8, we plot the TPR and the FPR for the mouth and right eyelid closures according to the threshold of detection. We can see that for lower thresholds, only our system creates consistent mouth/eyes closures as its TPR is always the highest. The system of Holden et al. [13] is not capable of producing eyes closures so its FPR is zero for lower thresholds. Meanwhile, we control that our system does not hallucinate motion as its FPR remains low.

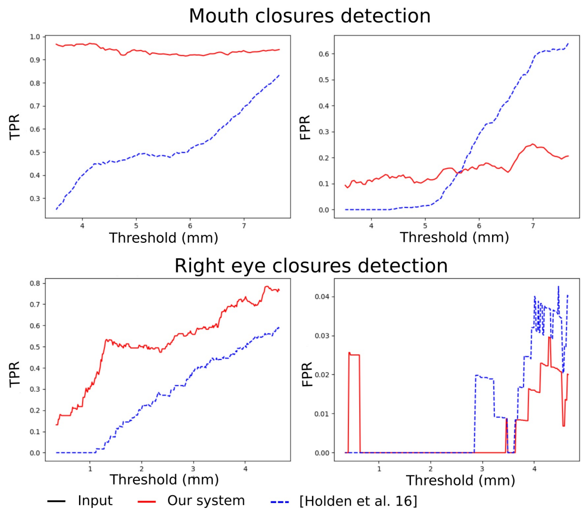

Figure 8: Comparison with [13]: Curves of the TPR and the FPR of the mouth and eyes closures on the testset.

An interesting feature of data-based motion models is the ability to model immobility, that we observe here on the first curve plotting the inner lips distance in Figure 6. Between the 40^{th} frame and the 60^{th} frame, we can observe that our system can cope with no inner lips movements for multiple consecutive frames.

5.2 Data dependency: transfers on another database

As with all data-based approach, it is important to know how the approach depends on the size and content of the dataset. Thus, we test the validity of our model (trained with the B3D(AC)ˆ2 dataset) on the recently released Vocaset database [8]. This dataset is composed of sequences of 12 subjects speaking sentences from the TIMIT corpus. We use the same processing pipeline to get the blendshapes coefficients sequence as in Section 3 except that we do not use 2D information. We downsample the frame rate to 25 fps to match the frame rate of our dataset B3D(AC)ˆ2.

As shown in Table 2, our system trained with only the trainset of the B3D(AC)ˆ2 dataset and applied to the whole Vocaset gives a comparable MSE (0.004) as a one trained with both the Vocaset and B3D(AC)ˆ2 dataset (0.003). The Vocaset content is less diversified, that is why the results obtained using only this dataset are the lowest. Indeed, there is no emotional sequence in this dataset unlike in the B3D(AC)ˆ2 dataset which is one-half composed with emotional sequence. In such sequences, the amplitude of the movements is generally higher compared to neutral sequences. So, at test time, it is easier for a system trained with emotional content to render neutral speech content than in the reverse order. We can see on the supplementary material that our system is suitable to model any new subjects in the Vocaset.

Table 2: Quantitative results of our system trained with the trainset of the B3D(AC)ˆ2 dataset.

Trainset	Testset	MSE (mouth)	TPR	FPR
Vocaset	Vocaset	0.038	0.87	0.06
Vocaset	B3D(AC)ˆ2	0.05	0.81	0.38
B3D(AC)ˆ2	Vocaset	**0.004**	**0.98**	0.22
B3D(AC)ˆ2	B3D(AC)ˆ2	0.008	0.98	0.22
Both	Vocaset	**0.003**	0.95	0.22
Both	B3D(AC)ˆ2	0.01	0.95	0.35

5.3 System Robustness: necessity of the autoencoder

Here we evaluate the robustness of our system by its ability to handle inadequate input. It shows that using the regressor alone would be more accurate than the full system in term of MSE as shown in Table 1. However, without the autoencoder, the regressor alone would be too sensitive to user's inputs, leading to unrealistic animation output as soon as input control parameter did not match a realistic animation. The regressor handles the accuracy of mapping from control parameters to blendshapes animation, while the subsequent autoencoder keeps the resulting animation inside the space of plausible animation. Both components are essential for a system aimed at non-specialist users. We show this by inputing different mouth-opening constraints and looking at inner-lips distance at output, as curves on Figure 9 and visually on Figure 10. We can see that the regressor is unstable; as soon as the input constraints constitute an unrealistic facial pattern, the output shapes are unrealistic. The autoencoder cleans up the output animation of the regressor, generating a natural animation. For instance, it projects unrealistic mouth openings to realistic ones when it is required. Note that this is not just a geometric projection operation but a temporal one as well, as our autoencoder models time-windows of animation. More results on full animations are provided in the supplementary video.

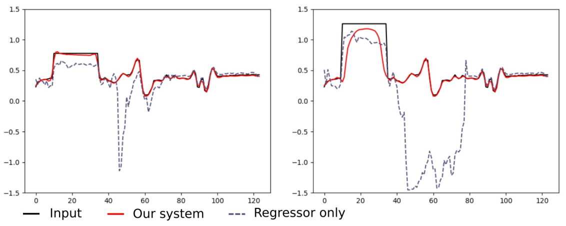

Figure 9: Realistic (left) and unrealistic (right) mouth opening input signal and the corresponding output with our system with and without the autoencoder. We can observe that the regressor alone is too sensitive to the input : unrealistic patterns appear as soon as a unseen input is given.

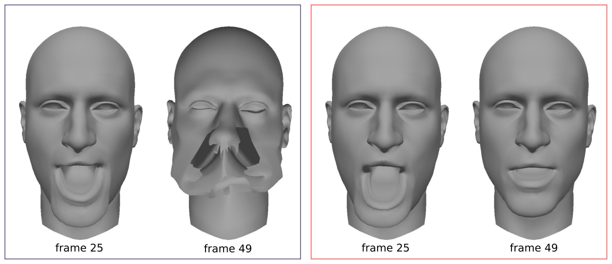

Figure 10: Output animation with an unrealistic mouth opening without (left) and with the autoencoder (right).

5.4 Usability: integration in a traditional facial animation pipeline

Even if our system processes whole sequences of animation, its architecture is light and performs network inference very quickly. This renders interactive uses of such a system imaginable. In this work, we propose an interactive editing tool that is meant to be easily integrated in a facial animation pipeline that would enable non-specialist users to generate quality facial animation. A common modern performance-based facial animation pipeline consists in acquiring sequences of actor performance, tracking his/her facial expressions, retargeting those to blendshapes animation coefficients, and finally manually tuning the obtained animation. Today, real-time face tracking methods enable non-expert to get raw facial animation from simple video feeds, but the animation is often noisy. Moreover, as in professional pipelines the animation must often be edited later on to match the artistic intent. Our tool finds its place at the editing stage of the pipeline. Through an interactive interface, the user can continuously refine the animation to produce the desired animation with low-latency. Indeed, the inference time, time between the moment the user applies its new control parameters and the moment the new final animation is produced is in average less than 0.015s for a typical scene of 8 seconds (202 frames) on CPU.

To showcase this, we use an off-the-shelf real-time face tracking software that outputs blendshapes coefficients. We developed a user interface that enables to visualize temporal curves for our control parameters and edit them via click-and-drag. Our network then runs inference to deliver the edited facial animation at interactive rate. One can for instance change a neutral speech animation sequence by increasing the mouth corners distance, causing the character to smile while speaking. Figure 11 shows a frame with the 2D tracking landmarks, the corresponding animation given by the tracking as well as the final edited animation with a smile. More isolated edits can be performed such as forcing a mouth closure or a blink by acting on the relevant local frames. Dynamic results of such edits are presented in the supplementary video.

6 CONCLUSION & FUTURE WORK

In this paper, we have presented a learning-based editing system that enables easy manipulation of facial animation with simple and intuitive control parameters. This tool can be used by non-specialist users to complete their facial animation pipeline with a tool that can correct and alter animation with no experience in facial animation.

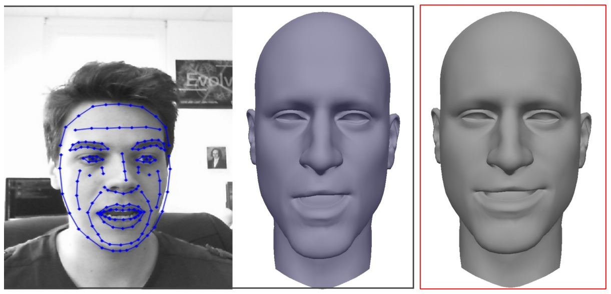

Figure 11: Edition of a video-based animation: example of frame with 2D tracking landmarks, the animation given by the tracking sofware and the final edited animation.

Our method is content-independent and emphasizes robustness, resulting in an editing tool that outputs plausible animation even when given unprecise or unrealistic user inputs. We have studied our systems behavior by evaluating quantitatively on error and semantic metrics versus relevant previous work, and have experimented with different datasets. We have demonstrated the necessity of using resolution-preserving architecture neural network to retain the temporal high-frequency information of facial motion, which architectures from previous work did not address. To be able to train our system and perform quantitative and qualitative evaluation we have reprocessed and augmented the dataset of B3D(AC)ˆ2, and we plan to make this data available for reproducibility.

One important main limitation comes from the quality of this dataset. Indeed, the native capture frame rate of the videos is 25 fps, which is too low to acquire all relevant natural facial cues. Important high-frequency information has already been lost at acquisition time. We also note that the performance of our system strongly depends on the choice of the control parameters. More parameters result in a more accurate but less intuitive system that is harder to manipulate. Conversely, few parameters cause ambiguity in mapping controllers to facial shapes, resulting in less control over the produced animations. As an example, the sidewise motion of the chin is lost due to the lack of dedicated controller (see Figure 13). Moreover, in our current implementation these parameters have to be continuous. An interesting direction of research would be to study the possibility to provide discrete inputs, even semantic ones -such as phonemes-, to control the generated animation.

ACKNOWLEDGMENTS

We thank Emily Guy for her unnamed viewer. We also would like to thank all the members of Dynamixyz for his software Performer. We also thank Renaud Seguier for his supervision.

REFERENCES

[1] Ijaz Akhter, Tomas Simon, Sohaib Khan, Iain Matthews, and Yaser Sheikh. 2012. Bilinear spatiotemporal basis models. *ACM Transactions on Graphics* 31, 2 (April 2012), 1–12. https://doi.org/10.1145/2159516.2159523

[2] Ken Anjyo, Hideki Todo, and J.P. Lewis. 2012. A Practical Approach to Direct Manipulation Blendshapes. *Journal of Graphics Tools* 16, 3 (Aug. 2012), 160–176. https://doi.org/10.1080/2165347X.2012.689747

[3] Volker Blanz and Thomas Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 187–194. https://doi.org/10.1145/311535.311556

[4] Yong Cao, Petros Faloutsos, and Frédéric Pighin. 2003. Unsupervised learning for speech motion editing. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 225–231.

[5] Ozan Cetinaslan, Verónica Orvalho, and John Lewis. 2015. Sketch-Based Controllers for Blendshape Facial Animation.. In *Eurographics (Short Papers)*. 25–28.

[6] Jing Chi, Shanshan Gao, and Caiming Zhang. 2017. Interactive facial expression editing based on spatio-temporal coherency. *The Visual Computer* 33, 6-8 (2017), 981–991.

[7] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *arXiv:1511.07289 [cs]* (Nov. 2015). http://arxiv.org/abs/1511.07289 arXiv: 1511.07289.

[8] Daniel Cudeiro, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael J Black. 2019. Capture, Learning, and Synthesis of 3D Speaking Styles. *Computer Vision and Pattern Recognition (CVPR)* (2019), 11.

[9] Dynamixyz. 2019. Performer. http://www.dynamixyz.com/performer-single-view/

[10] Gabriele Fanelli, Juergen Gall, Harald Romsdorfer, Thibaut Weise, and Luc Van Gool. 2010. A 3-D Audio-Visual Corpus of Affective Communication. *IEEE Transactions on Multimedia* 12, 6 (Oct. 2010), 591–598. https://doi.org/10.1109/TMM.2010.2052239

[11] Michael Gleicher. 1997. Motion editing with spacetime constraints. In *Proceedings of the 1997 symposium on Interactive 3D graphics - SI3D '97*. ACM Press, Providence, Rhode Island, United States, 139–ff. https://doi.org/10.1145/253284.253321

[12] Ikhsanul Habibie, Daniel Holden, Jonathan Schwarz, Joe Yearsley, Taku Komura, Jun Saito, Ikuo Kusajima, Xi Zhao, Myung-Geol Choi, and Ruizhen Hu. 2017. A Recurrent Variational Autoencoder for Human Motion Synthesis. *IEEE Computer Graphics and Applications* 37 (2017), 4.

[13] Daniel Holden, Jun Saito, and Taku Komura. 2016. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics* 35, 4 (July 2016), 1–11. https://doi.org/10.1145/2897824.2925975

[14] Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. 2015. Learning motion manifolds with convolutional autoencoders. ACM Press, 1–4. https://doi.org/10.1145/2820903.2820918

[15] Pushkar Joshi, Wen C Tien, Mathieu Desbrun, and FrÃľdÃľric Pighin. 2003. Learning Controls for Blend Shape Based Realistic Facial Animation. *SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), 187–192.

[16] Lucas Kovar, Michael Gleicher, and FrÃľdÃľric Pighin. 2002. Motion graphs. In *ACM SIGGRAPH*. ACM, 482.

[17] Manfred Lau, Jinxiang Chai, Ying-Qing Xu, and Heung-Yeung Shum. 2009. Face poser: Interactive modeling of 3D facial expressions using facial priors. *ACM Transactions on Graphics* 29, 1 (Dec. 2009), 1–17. https://doi.org/10.1145/1640443.1640446

[18] J P Lewis and Ken-ichi Anjyo. 2010. Direct Manipulation Blendshapes. *IEEE Computer Graphics and Applications* 30, 4 (July 2010), 42–50. https://doi.org/10.1109/MCG.2010.41

[19] Hao Li, Bart Adams, Leonidas J. Guibas, and Mark Pauly. 2009. Robust Single-view Geometry and Motion Reconstruction. In *ACM SIGGRAPH Asia 2009 Papers (SIGGRAPH Asia '09)*. ACM, New York, NY, USA, 175:1–175:10. https://doi.org/10.1145/1661412.1618521

[20] Hao Li, Jihun Yu, Yuting Ye, and Chris Bregler. 2013. Realtime Facial Animation with On-the-fly Correctives. *ACM Trans. Graph.* 32, 4 (2013), 42:1–42:10. https://doi.org/10.1145/2461912.2462019 OK.

[21] Qing Li and Zhigang Deng. 2008. Orthogonal-Blendshape-Based Editing System for Facial Motion Capture Data. *IEEE Computer Graphics and Applications* 28, 6 (Nov. 2008), 76–82. https://doi.org/10.1109/MCG.2008.120

[22] L. Ma and Z. Deng. 2018. Real-Time Facial Expression Transformation for Monocular RGB Video: Real-Time Facial Expression Transformation for Monocular RGB Video. *Computer Graphics Forum* (Oct. 2018). https://doi.org/10.1111/cgf.13586

[23] Xiaohan Ma, Binh Huy Le, and Zhigang Deng. 2009. Style learning and transferring for facial animation editing. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 123–132.

[24] Julieta Martinez, Michael J. Black, and Javier Romero. 2017. On human motion prediction using recurrent neural networks. *arXiv:1705.02445 [cs]* (May 2017). http://arxiv.org/abs/1705.02445 arXiv: 1705.02445.

[25] Roger Blanco i Ribera, Eduard Zell, J. P. Lewis, Junyong Noh, and Mario Botsch. 2017. Facial retargeting with automatic range of motion alignment. *ACM Transactions on Graphics* 36, 4 (July 2017), 1–12. https://doi.org/10.1145/3072959.3073674

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention âĂŞ MICCAI 2015*, Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi (Eds.). Vol. 9351. Springer International Publishing, Cham, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

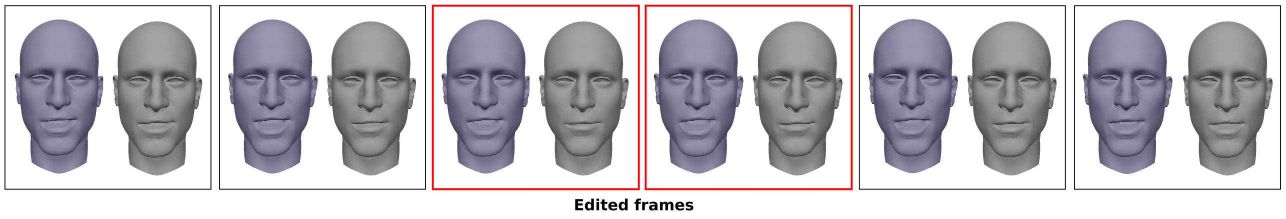

Figure 12: Few frame edition : enforce the mouth closure.

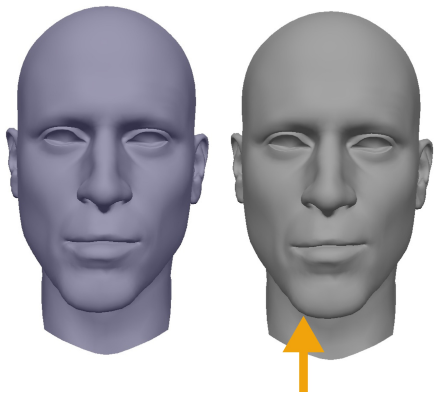

Figure 13: Limitation of our system: some motion such as the sidewise motion of the chin of the groundtruth (left) is lost at the output of our system (right).

[27] Yeongho Seol, J. P. Lewis, Jaewoo Seo, Byungkuk Choi, Ken Anjyo, and Junyong Noh. 2012. Spacetime expression cloning for blendshapes. *ACM Transactions on Graphics (TOG)* 31, 2 (2012), 14.

[28] Robert W. Sumner and Jovan Popović. 2004. Deformation transfer for triangle meshes. *ACM Trans. Graph.* 23, 3 (2004), 399–405. https://doi.org/10.1145/1015706.1015736 OK.

[29] Sarah Taylor, Taehwan Kim, Yisong Yue, Moshe Mahler, James Krahe, Anastasio Garcia Rodriguez, Jessica Hodgins, and Iain Matthews. 2017. A Deep Learning Approach for Generalized Speech Animation. *ACM Trans. Graph.* 36, 4 (July 2017), 93:1–93:11. https://doi.org/10.1145/3072959.3073699

[30] J. Rafael Tena, Fernando De la Torre, and Iain Matthews. 2011. Interactive Region-based Linear 3D Face Models. In *ACM SIGGRAPH 2011 Papers (SIGGRAPH '11)*. ACM, New York, NY, USA, 76:1–76:10. https://doi.org/10.1145/1964921.1964971 event-place: Vancouver, British Columbia, Canada.

[31] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. 2011. Realtime performance-based facial animation. In *ACM transactions on graphics (TOG)*, Vol. 30. ACM, 77.

[32] Li Zhang, Noah Snavely, Brian Curless, and Steven M Seitz. 2004. Spacetime Faces: High Resolution Capture for Modeling and Animation. *ACM Trans. Graph.* 23 (2004), 548–558.

[33] Qingshan Zhang, Liu ,Zicheng, and Heung-Yeung Shum. 2003. Geometry-driven photorealistic facial expression synthesis. *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), 177–186. https://doi.org/10.1109/TVCG.2006.9

[34] Yang Zhou, Zhan Xu, Chris Landreth, Evangelos Kalogerakis, Subhransu Maji, and Karan Singh. 2018. Visemenet: Audio-driven animator-centric speech animation. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 161.