Generalized Speech Animation Based on Deep Learning

Xiangyu Wang

University of Southern California

Abstract. In this project, I mainly implemented an autonomous speech generation system from raw audio input based on [1]. It uses a sliding window deep neural network to learn the mapping from phoneme input sequences to mouth movements. The advantage of this system is that it needs very few parameter tuning, can be generalized to any input sequences, compatible with existing animation retargeting approaches and can be run in real time.

Keywords: Neural Network, Speech Animation

1 Introduction

Speech Animation is one important part in realistic character animation. The current trend in the field of speech animation mainly tend toward one of the two extremes. For those large-budget productions, they mainly use high-end performance capture facilities or hire skillful animators to do the keyframing. But the cost is too high. While for those low-budget productions, they just use several simplified viseme lip shapes whatever the input phoneme sequences are. But the quality is usually very low. In recent years, the data-driven methods are becoming popular, most of which are based on the blendshape algorithm. The simplicity of the blending function usually limits the complexity of the dynamics of visual speech that can be modelled. In this project, I use modern machine learning methods, like deep learning, to learn the dynamics directly from the data in a model-free way.

2 Approach

2.1 Overview

The overall structure of the system is shown in Figure 1. It's mainly composed of four components: 1)face feature extraction, 2)speech recognition, 3)Deep Neural Network(DNN) regression, 4)Rig-space retargeting.
In the training session, the feature extractor extracts facial features from each video frame and draws the corresponding phoneme label at the current frame. Such feature pairs will then be thrown into a DNN to train a regressor from the phoneme labels to the corresponding face shape parameters. Then during the

prediction session, the predicted shape parameters will be retargeted to a set of rigging parameters to control the lip movement of the 3d model, thus generating real-time speech animation. in the following parts, I will elaborate on each of them in detail.

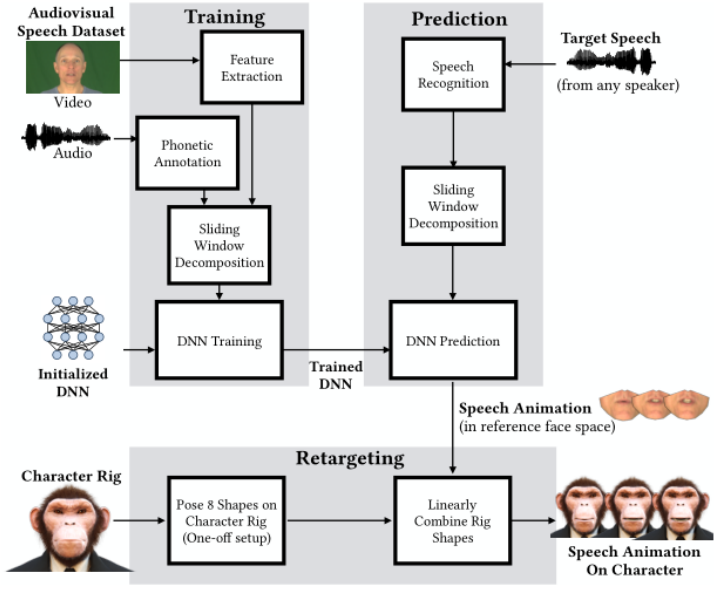

Fig. 1. An Overall Structure of the System

2.2 Face Feature Extraction

The Active Appearance Model(AAM)[2] is used here to extract face features from the video frames. An AAM contains a statistical model of the shape and grey-level appearances of the faces in the training dataset. Here, we only use the shape components of the AAM model because the appearance information doesn't contain any information of the lip movement. As shown in Figure 2, the shape component, **s**, represents 34 vertices of the lower face and jaw, where $s = s_0 + \sum_{i=1}^{m} s_i p_i$. Here we use $m = 15$ modes to capture more than 90% of shape variations. The mean shape is s_0, each s_i is a shape basis vector, and p_i are the shape parameters. In my system, the AAM model is trained with Menpo[1] on the LFPW dataset, which contains 811 face images labelled landmarks.

[1] http://www.menpo.org/

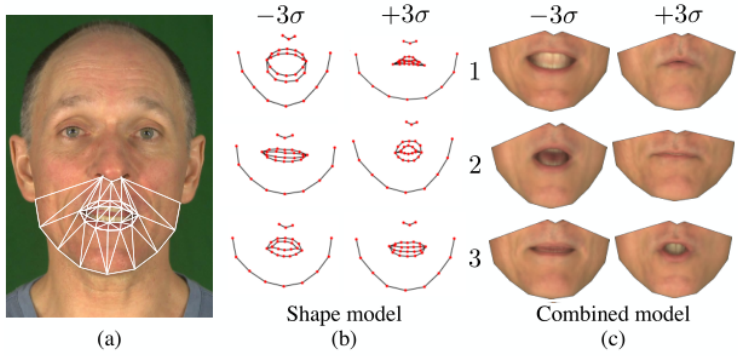

Fig. 2. a)The 34 vertices of the AAM shape component. b)The first three shape modes in the AAM shape component shown at ±3 standard deviations from the mean. c)The first three modes in the AAM combined model shown at ±3 standard deviations from the mean.

2.3 Speech Recognition

In this part, I used off-the-shell software Penn Phonetics Lab Forced Aligner(P2FA) [3], which is based on the HTK toolbox[4]. It mainly converts raw audio input to individual phonemes, each span a certain time interval. Then each phoneme is represented as a 40-dimension one-hot vector and fed into the neural network for training.

2.4 Deep Learning Sliding Window Regression

In the system, I used a fully connected feed-forward network with a sliding window input layer(Figure 3), three hidden layers and a final output layer. Each hidden layer consists of 3000 hidden units. The input of the network is the phoneme one-hot vector, while the output is the corresponding AAM shape parameter. Here I use the sliding window DNN instead of the sequential models such as LSTM or RNN because it may need much more data to train them to capture the localized coarticulation effects between the phonemes. Here the input window length is set to be 11 and the output window length is set to be 5. The neural network is implemented with Tensorlfow[2], with the batch size to be 100 and the dropout rate to be 50%. The training data comes from the KB-2K dataset, which consists of 2543 phonetically diverse sentences.

[2] https://www.tensorflow.org/

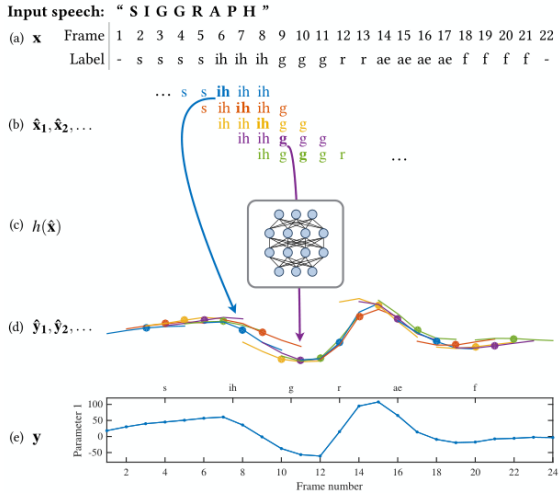

Fig. 3. The Deep Learning Sliding Window Pipeline

2.5 Rig-space Retargeting

For retargeting, I used the first four shape parameters to describe the more than 90% of the variation of the motions. At each component, I took $\pm 3\sigma$ to get 8 reference faces, as is shown in Figure 4. For each reference face, I got the corresponding rigging parameter by fitting the FLAME model[5]. Then the 8 bases are used to span the whole rigging space. We map from the 4-dimension shape parameter $\mathbf{p}$ to the 8-dimension weights $\mathbf{w}$ with the relation below

$$w_{2i-1} = max((-1)^i \frac{p_i}{3\sigma_i}, 0), \quad i = 1 \cdots 4$$

Then the final mapped rigging parameter R_t can be represented as

$$R_t = \sum_{i=1}^{8} w_i (R_i - r_0) + r_0$$

where R_i are the rigging parameter at the reference shapes and r_0 is that at the neutral shape.

Fig. 4. The 8 reference shapes defined at $\pm 3\sigma$ in each shape component

3 Results

Figure 5 shows an example of the word "impulses".

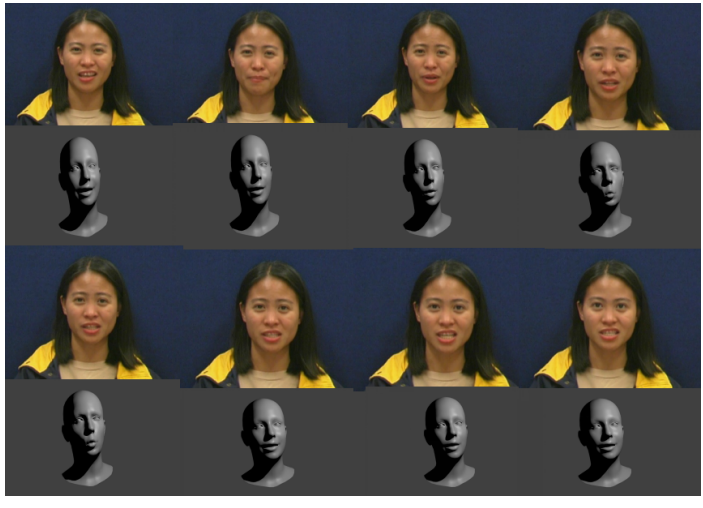

Fig. 5. The ground truth and the predicted animation for the word "impulses", which consists of 8 phonemes: ih,m,p,ah,l,s,ah,z

Figure 6 shows another example of the word "major".

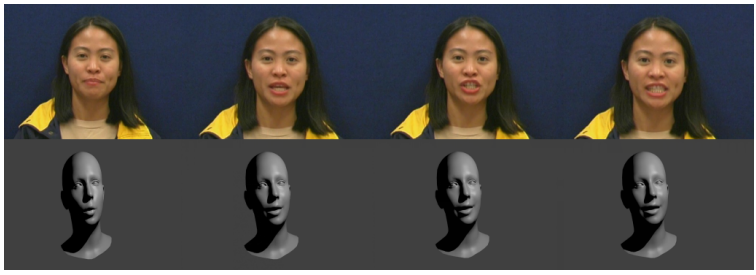

Fig. 6. The ground truth and the predicted animation for the word "major", which consists of 4 phonemes: m,ey,jh,er

4 Future Work

Since there's limited time for the project, the result may not seem very good. To make the pipeline work better, it may need a set of more accurately selected reference shapes. I may also add more linguistically motivated features as input besides the one-hot vector, such as whether the two neighboring phonemes belong to a certain consanant-vowel pair. What's more, it may improve if I train separate AAM for pixels outside the mouth area and those inside the mouth. Last but not least, it may need a more robust 2d to 3d mapping mechanism.

5 Conclusion

In this project, I basically implemented a real-time end-to-end system that can generate speech animation directly from raw audio input system. From the result, it reacts correctly to most of the phonemes although there are still some small artifacts. Some feature engineering work may improve the results. At a more general stage, such technique may find its way in other fields, such as music visualization and human computer interaction.

References

1. Taylor, S., Kim, T., Yue, Y., Mahler, M., Krahe, J., Rodriguez, A.G., Hodgins, J., Matthews, I.: A deep learning approach for generalized speech animation. ACM Transactions on Graphics (TOG) **36**(4) (2017) 93
2. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. IEEE Transactions on pattern analysis and machine intelligence **23**(6) (2001) 681–685
3. Yuan, J., Liberman, M.: Speaker identification on the scotus corpus. Journal of the Acoustical Society of America **123**(5) (2008) 3878
4. Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., et al.: The htk book. Cambridge university engineering department **3** (2002) 175
5. Li, T., Bolkart, T., Black, M.J., Li, H., Romero, J.: Learning a model of facial shape and expression from 4d scans. ACM Transactions on Graphics (TOG) **36**(6) (2017) 194