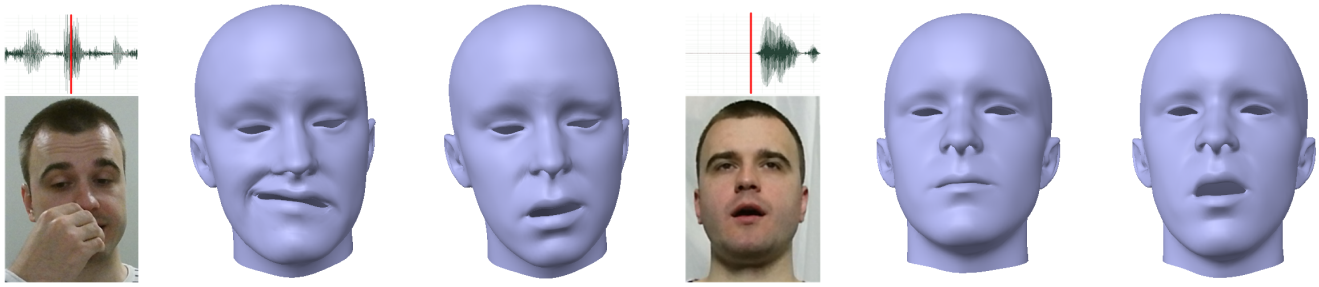
Video-Audio Driven Real-Time Facial Animation

Yilong Liu[1] Feng Xu[1,2*] Jinxiang Chai[3] Xin Tong[2] Lijuan Wang[2] Qiang Huo[2]
[1]Tsinghua University [2]Microsoft Research [3]Texas A&M University



(a) *Input* **(b)** *Visual-only result* **(c)** *Our result* **(d)** *Input* **(e)** *Audio-only Result* **(f)** *Our result*

Figure 1: *Comparing our audio-visual combined solution with visual-only and audio-only solutions. (a-c) show that our combined solution handles occlusion in the mouth region better than the visual-only solution. (d-f) show that when the user is not speaking, our solution reconstructs the mouth shape unlike the audio-only solution. In the audio waveforms, the red line indicates the time position of the input frame.*

Abstract

We present a real-time facial tracking and animation system based on a Kinect sensor with video and audio input. Our method requires no user-specific training and is robust to occlusions, large head rotations, and background noise. Given the color, depth and speech audio frames captured from an actor, our system first reconstructs 3D facial expressions and 3D mouth shapes from color and depth input with a multi-linear model. Concurrently a speaker-independent DNN acoustic model is applied to extract phoneme state posterior probabilities (PSPP) from the audio frames. After that, a lip motion regressor refines the 3D mouth shape based on both PSPP and expression weights of the 3D mouth shapes, as well as their confidences. Finally, the refined 3D mouth shape is combined with other parts of the 3D face to generate the final result. The whole process is fully automatic and executed in real time.

The key component of our system is a data-driven regresor for modeling the correlation between speech data and mouth shapes. Based on a precaptured database of accurate 3D mouth shapes and associated speech audio from one speaker, the regressor jointly uses the input speech and visual features to refine the mouth shape of a new actor. We also present an improved DNN acoustic model. It not only preserves accuracy but also achieves real-time performance.

Our method efficiently fuses visual and acoustic information for 3D facial performance capture. It generates more accurate 3D mouth motions than other approaches that are based on audio or video input only. It also supports video or audio only input for real-time facial animation. We evaluate the performance of our system with speech and facial expressions captured from different actors. Results demonstrate the efficiency and robustness of our method.

*Corresponding author. E-mail:feng-xu@tsinghua.edu.cn

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—animation;

Keywords: real time facial tracking, speech animation, facial animation

1 Introduction

With recent advances in real-time facial tracking and performance capturing techniques, performance-driven facial animation has become available for many consumer-level real-time applications, such as telecommunications, computer games, training and other online interactions. Although state-of-the-art techniques [Weise et al. 2011; Li et al. 2013; Bouaziz et al. 2013; Cao et al. 2013; Cao et al. 2014a] demonstrate relatively accurate 3D tracking results for large-scale facial expressions, it is still difficult for them to capture accurate 3D mouth shapes, especially for fast lip motions when the actor is talking. Since all these methods are based on color or depth sensors, they are also prone to fail when the face is partially occluded or the head is in an extreme pose.

In this paper, we present a real-time facial tracking and animation system that is based on a Kinect sensor with audio and video input. Our method can be applied to any new user without any user-specific training and is robust to occlusions, extreme head pose, and background noise. The key idea of our method is that the face shape, especially the mouth shape can be derived from the actor's facial appearance and speech. Based on this observation, our system captures the color, depth of an actor's face and speech with a Kinect sensor and jointly uses the audio and video input to reconstruct the actor's 3D facial performance in real time.

A key challenge in designing this real-time facial animation system is to find a model for representing both 3D face shapes and their correlated speech data. On one hand, the model should be generic enough so that it can well cover the face shapes and voice variations of difference identities to avoid user-specific training or adaptation.

On the other hand, the model also should be efficient so that it can be evaluated in real time.

In this work, we use three models instead of a unified one to model the face shapes and correlated speech data. We apply a multi-linear model learned from a public face dataset to represent the identity and expression variations of 3D face shapes. We also develop a speaker-independent DNN acoustic model that is learned from a public speech dataset to extract the phoneme state posterior probabilities (PSPP) from speech audio in real time. Our acoustic model not only preserves the accuracy of DNN-based acoustic models for different speakers but also achieves real-time performance by removing the forward dependency. These two generic models allow us to extract user-independent features from input frames in real time and avoid user-specific training.

To reconstruct the 3D mouth shape from joint visual and acoustic features, we design a data-driven lip motion regressor for modeling the correlation between speech data and mouth shapes. The regressor is constructed from a database of 3D mouth shape sequences and synchronized speech audio captured from a speaker in an offline preprocessing step. To this end, we apply the multi-linear model to fit the speaker's face and mouth shapes in each frame and extract the PSPP from the corresponding audio frame, and then use the resulting lip performance, represented as feature positions around the mouth region, and the PSPP to index the corresponding mouth shapes. To better model the speech coarticulation, we associate a subsequence of mouth shapes with the index of the center frame.

At run time, our system first reconstructs 3D facial expressions from color and depth frames with the multi-linear model and then extracts the PSPP from audio frames using the DNN-based acoustic model. Based on lip performance and PSPP as well as their confidences, the lip motion regressor refines the 3D mouth shape. It searches the database to find the first K subsequences whose index best matches the inputs. From all candidates, the regressor selects a mouth shape with the optimal distance to both the input and the subsequences of previous frames. We design a fast search scheme to quickly find the candidates from the database for inputs with varying visual and acoustic confidences. Finally, we combine the refined 3D mouth shape with other parts of the 3D face to generate the final result. The whole process is fully automatic and executed in real time.

With combined visual and acoustic inputs, our system reconstructs better 3D facial shapes than other approaches that are based on audio or video input only. It is robust to occlusions, extreme head poses, and supports video or audio only input for real-time facial animation. We evaluate the performance of our system with speech and facial expressions captured from different actors. Results demonstrate the efficiency and robustness of our method.

2 Related Work

Facial Performance Capturing and Tracking have been extensively studied in both computer graphics and vision. Here we only discuss the real-time 3D facial performance capturing approaches that are most related to our method. Please refer to [Ren et al. 2014] for discussions of recent real-time 2D facial tracking work and [Beeler et al. 2011] for discussions about the latest offline 3D facial performance capturing methods.

For color and depth input, Weise et al. [2011] construct a user-specific blendshape model as a preprocessing stage and then fit the blendshape weights for each color and depth frame at run time. The resulting weights are then transferred to other 3D avatars for real-time facial animation. Bouaziz et al. [2013] and Li et al. [2013] present solutions to further avoid the preprocessing by jointly optimizing the user-specific expression model and expression parameters at run time. Although these methods can capture the 3D facial performance in real time and are robust to illumination changes, they may fail for faces with large rotation or occlusion. Recently, Hsieh et al. [2015] propose a method which gives uninterrupted facial tracking even with large occlusions. However, it can not recover motions of the occluded regions.

For color input, Cao et al. [2013] present a regression method for real-time 3D facial performance capture that requires user-specific training and calibration. Later, Cao et al. [2014a] propose a general regressor that is learned from a public image dataset for reconstructing 3D facial shapes from video frames.

Different from these methods that use visual input only for capturing 3D facial performance, our method exploits both visual and acoustic input for this task and thus reconstructs more accurate mouth shape than existing methods, especially when the mouth region is occluded.

Speech-driven facial animation has also been studied for a long time in computer graphics and speech synthesis. A set of approaches directly construct the mapping from audio to visual space. Chuang and Bregler [2005] construct a database relating audio pitch to head motion, and use it to drive head motion with audio input. Le et al. [2012] make use of Nonlinear Canonical Correlation Analysis (NCCA) and non-negative linear regression model to further synthesize eye gaze and eyelid motion for live speech, respectively. Brand [1999] models the mapping between vocal and facial dynamics with an HMM and applies a trajectory optimization for synthesizing smooth facial animations. Massaro et al. [1999] use an artificial neural network to map the Mel-Frequency Cepstral Coefficients (MFCC) to visual parameters. Fu et al. [2005] give a comparison of several single HMM-based conversion approaches. Wang et al. [2006] use a single hidden Markov model to realize the mapping between MFCC and Facial Animation Parameters (FAP). Xie and Liu [2007] propose a coupled HMM to realize video realistic speech animation. Zhuang et al. [2010] propose a method using the minimum converted trajectory error criterion to optimize single Gaussian Mixture Model (GMM) training to improve the audio-visual conversion. Although these methods can be used for different speakers, their robustness depends heavily on the scale of the stereo audio-visual database, which usually is very limited, that is used to train such a mapping function.

Other approaches synthesize speech animations from a phone sequence labeled from the audio input. Bregler et al. [1997] create speech video of an new phoneme sequence by using the mouth images in the training footage whose phonemes are matched to input phonemes. Ezzat et al. [2000] convert a phone sequence into smooth speech video by morphing corresponding visemes. Lei et al. [2003] map the phonemes to visemes using a fixed table, where the visemes are modeled by an HMM. King and Parent [2005] create animation for known text by building a facial model and its viseme set. Cao et al. [2005] learn a graph of phoneme nodes with corresponding face motions and emotion tags from the captured speech animation data and use the graph to generate expressive speech animations with a new phoneme and emotion tag sequence. Deng et al. [2006] train a speech coarticulation model and an expression model from the motion capture data and then apply the models for phone-based expressive speech animation generation. Wampler et al. [2007] model expressive speech animations as a multi-linear model of identity, expression and phonemes and apply it for constructing the speech animation of a phoneme sequence for a new 3D face. Sun et al. [2008] use phoneme-based key-frame interpolation for lips animation. Taylor et al. [2012] propose dynamic visemes to better model and synthesize speech coarticulation. In these methods, the phoneme sequence is transformed from the speech signals either by human labelers or by an automatic speech recognizer (ASR) [Ra-

biner and Juang 1993]. While the former is expensive and subject to inconsistency resulting from human disagreement in phoneme labeling, the latter always requires a large buffer of forward frames to determine the current phoneme and thus leads to delay. As a result, all these existing solutions are unacceptable for real-time facial animation.

Instead of converting input speech into phonemes though complete decoding as in an ASR system, we develop a speaker-independent DNN acoustic model for extracting the phoneme state posterior probabilities from the audio input in real time. Our model not only preserves the accuracy and robustness of the DNN-based methods but also achieves real-time performance. The lip motion regressor used in our solution is similar to video rewrite in [Bregler et al. 1997] and also uses K-nearest neighbors for speech animation generation. However, our method is different from video rewrite in several ways. While video rewrite only uses audio for offline speech animation, our method jointly uses audio and video features for real-time facial animation. By combining information from both input channels, our method can generate better 3D mouth motions and is more robust. Moreover, instead of selecting single video frames for result synthesis, our method generates results using overlapped subsequences of the mouth animations and thus better maintains the speech coarticulation. Finally, while video rewrite requires user-specific training footage, our method models the mouth shape of new actors and the ones in the database with user-independent expression weights and avoids user-specific training.

3 Overview

Our system utilizes a single Kinect camera to record color and depth images of any user's facial performance as well as the user's speech data. With the visual and audio input, the system reconstructs the user's 3D facial performance in real time. Our system contains a training stage and an online stage as shown in Fig. 2. In the training stage, we first train a real-time DNN model to extract PSPPs from audio input, and then, by using the DNN model and a pre-trained multilinear model which represents face geometry as a set of identity and expression coefficients, we construct an audio-visual database which is used to model the relationship between the audio-visual input and the final mouth motion. In the online stage, we take the visual and audio input from a Kinect to generate the user's 3D facial performance in real time, with the help of the audio-visual database and the DNN and multilinear models.

Real-time DNN model and multilinear face model In the training stage, we first train a speaker-independent deep neural network (DNN) model, which estimates the *PSPP vector*, **a**, of any user's speech in real time. As the *PSPP vector* is speaker-independent and is highly correlated to speech content, we use it as our speech feature for estimating mouth motion when people speak.

Then we follow the method in [Cao et al. 2014b] and use their FaceWarehouse database (including 3D meshes of 150 identities with 47 pre-defined expressions) to train a multilinear model. This model is used to track facial motion from any user's visual input. Besides, we also use the multilinear model to define a user-independent visual feature. We first choose n vertices on the mouth region of the multilinear face mesh, and then the positions of the vertices, which are synthesized on a neutral identity with the tracked expression, are used as a user-independent visual feature, denoted as the *lip performance vector*, **v**.

Audio-visual database The audio-visual database, aiming to model the relationship between the audio-visual features and the final mouth motion, contains both the audio-visual signal and the corresponding ground truth mouth motion. To construct the database, we record visual and audio data of one actor/actress speaking various speech content with a neutral expression. To record the ground-truth mouth motion, we place dense facial markers on the actor/actress's face and record the 3D positions of those markers. After the data collection, we estimate the *PSPP vector* using the DNN model and the *lip performance vector* using the multilinear model for each frame. At the same time, the ground-truth 3D mouth shape is also reconstructed by marker positions and further fitted by the multilinear model. The obtained expression coefficient is a user-independent mouth shape representation, denoted as the *mouth coefficient vector*, **w**. Thus the database is constructed by **a**, **v** and the corresponding **w** at each frame.

Online tracking The online stage contains four components. The visual tracking component utilizes the multilinear model to fit the online color and depth input. By doing this, the user's head pose **q**, facial identity $\mathbf{w}^{id}$, facial expression $\mathbf{w}^{exp}$ and the *lip performance vector* are reconstructed at each frame. The audio processing component extracts the *PSPP vector* of the input speech at each frame using the DNN model. By taking both *PSPP vector* and *lip performance vector* as input, the lip motion regression component utilizes our database to robustly and accurately reconstruct the *mouth coefficient vector* at each frame. Finally, by combining the *mouth coefficient vector* with the facial expression estimated by visual tracking, the full face geometry is reconstructed and can be transferred to an avatar in real time.

4 Offline Training

This section describes our training stage, which contains the construction of the real-time DNN model, the multilinear model and the audio-visual database. To build the multilinear model, we exactly follow the method in [Cao et al. 2014b]. Please refer to their paper for details. We will discuss our real-time DNN model and audio-visual database in the following subsections.

4.1 Real-time DNN Model

When people talk, their mouth shapes are highly correlated to their speech content. Based on this observation, many techniques are proposed to explore this correlation and reconstruct mouth shapes from audio signals. MFCC, Perceptual Linear Prediction (PLP) and other low-level audio features are commonly used in these techniques [Zhang et al. 2013; Brand 1999]. However, different speakers may have quite different vocal characteristics, leading to features quite different even when they are pronouncing the same words. Thus these techniques are limited to work on specific speakers. Their models need to be rebuilt for each new speaker.

On the other hand, in speech recognition, speech content can be recognized regardless of the vocal characteristics of different speakers [Seide et al. 2011b]. However, to fully recognize the speech content at any time instance, forward speech data is always required, thus these techniques cannot be performed in real-time. Our goal is to build a real-time model that reconstructs mouth shapes from speech data regardless of speaker. So we require a real-time audio feature that is highly correlated to speech content but is speaker-invariant. To achieve this goal, instead of recognizing input speech as words or phonemes though complete decoding as in an automatic speech recognition (ASR) system, we take the intermediate results which are the PSPPs predicted by DNN models. For the DNN acoustical model training, we adopt the Context-Dependent DNN-HMMs [Yu et al. 2010; Dahl et al. 2012; Seide et al. 2011a], which are a recently very promising and possibly disruptive acoustic model in ASR. The CD-DNN-HMMs model structure is inherited from

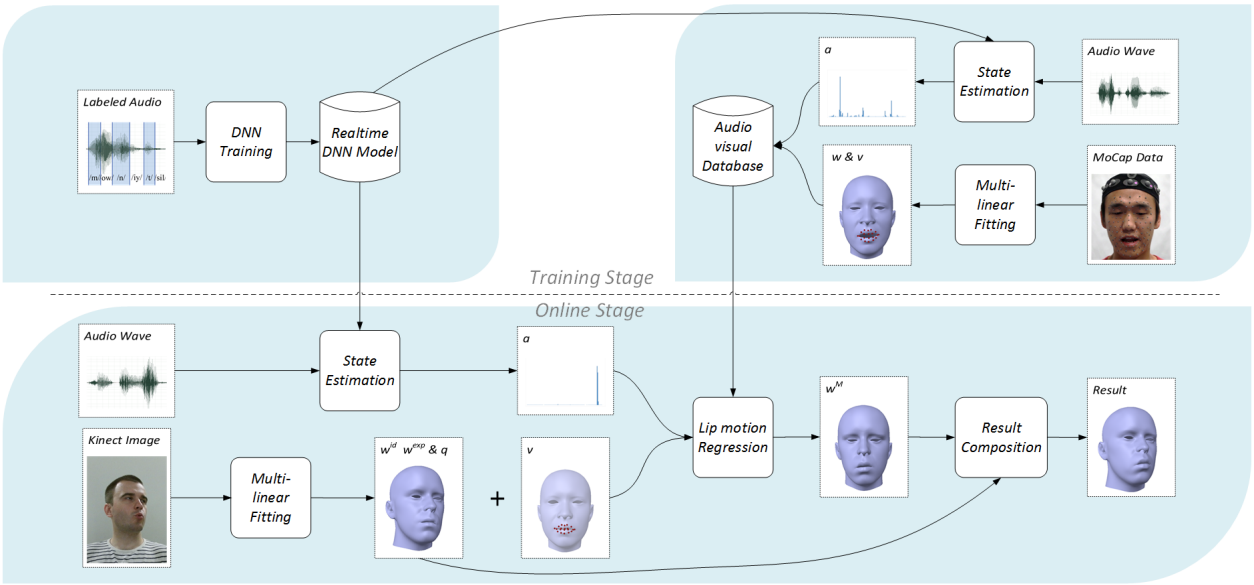


Figure 2: *Overview of our system.*

a matching GMM-HMM model that has been trained on the same data. That model is also used to initialize the class labels through forced alignment. The CD-DNN-HMMs model is trained using the 309-hour Switchboard-I training set [Godfrey and Holliman 1997]. In GMM-HMM model training, the system uses 13-dimensional PLP features with rolling-window mean-variance normalization and up to third-order derivatives with 52 dimensions for each frame. The speaker-independent cross-word triphones use the common 3-state topology and share 9304 CART-tied states. The whole training set is initialized with the tied state ID on alignment by a 60-mixture GMM-HMM. The next step of DNN training is to learn a mapping function between the contextual features and the tied state labels through a multi-layer, Deep Neural Network.

To address the real-time challenge in the desired online conversion system, we propose to use single-sided contextual features as DNN input. It is known that using both left and right long-span context features with derivatives can help improve DNN classification accuracy, but it also introduces time delay as the system needs to wait until all the contextual frames arrive to start the prediction of the current frame. For the same reason, we further remove all of the feature derivatives and use only the static feature alone in preparing the DNN inputs. As a result, we only use the static 13th-order PLP features on the current frame and its ten previous frames, to guarantee real-time performance in the conversion stage. The DNN model is trained with 7 data sweeps, consisting of 13x11 dimensions in the input layer, 7 layers of 2k hidden nodes and 9304 states in the output layer.

From our experiments, the accuracy of the PSPPs estimated by our real-time DNN model is a little bit lower than the original DNN model in [Seide et al. 2011b], which is trained with both forward and backward speech data. However, when judging from the accuracy of the mouth shape, our real-time model is comparable to the model used in [Seide et al. 2011b]. The validation of our real-time DNN model is detailed in the experiment section.

4.2 Audio-Visual Database

To robustly reconstruct mouth shapes from speech input, we require the database to cover the whole speech space. To satisfy this, we use designed content with 594 English sentences [Zhang et al. 2013], which contain most of the speech variations in the English language. Before recording, we attach dense facial IR markers (about 90-110 markers) on the actor/actress's face to reconstruct the ground-truth mouth shape. In the recording, with the actor/actress reading those sentences, we record both the speech data and the 3D positions of the facial markers.

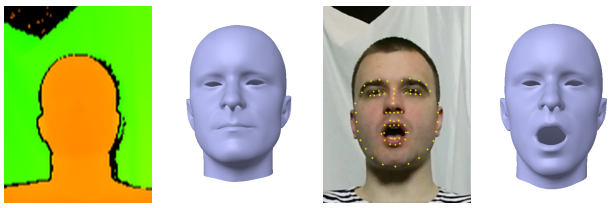
For the speech data, we extract the 132 dimensional *PSPP vector* at 100 FPS, using the DNN model trained in the previous subsection. The *PSPP vector* describes the speech content and is designed to be speaker-independent [Seide et al. 2011b]. We use $\mathbf{a}_i$ to denote the *PSPP vector* at frame i. For the marker position data, we first choose one recorded frame with a neutral expression and align the markers of this frame to a scanned neutral face of the actor/actress. Then we reconstruct the actor/actress' 3D facial motions at each frame by performing Laplacian deformation [Botsch and Sorkine 2008] on the scanned face mesh driven by the facial markers. As we have attached dense markers on the face, the deformed mesh can be treated as the ground-truth face mesh. The reconstructed meshes are further fitted by the multilinear model, and thus the identity-independent mouth motion is expressed as the expression weights of the multilinear model. As we only use it to represent mouth shape, we call it the *mouth coefficient vector* and denote it as $\mathbf{w}_i$ for frame i.

As the multilinear model is fitted to the recorded marker positions, the *lip performance vector* is obtained by synthesizing the positions of the predefined n vertices on the neutral identity. The *lip performance vector* is denoted as $\mathbf{v}_i$ for frame i.

After a syncing step, we construct the database as $\mathbf{D} = \{\mathbf{d}_i = (\mathbf{a}_i, \mathbf{v}_i, \mathbf{w}_i)|, i = 1, ..., N\}$ where N denotes the total number of frames in the database. In general, the sentences are read for more than 30 minutes, thus N is about 0.2 million.

5 Online Tracking

This section describes our online tracking algorithm in detail. The visual tracking step (Sec. 5.1) utilizes depth and color images to estimate $\mathbf{v}_t$ for the input frame t. The audio processing step (Sec. 5.2) takes an audio waveform as input to estimate $\mathbf{a}_t$ for the same input frame t. Then the lip motion regression step (Sec. 5.3) takes $\mathbf{v}_t$ and $\mathbf{a}_t$ as input to reconstruct the final mouth shape at frame t. Finally,



(a) (b) (c) (d)

Figure 3: *Visual Tracking. (a) Depth with neutral expression. (b) Result of identity fitting. (c) One color frame with detected features. (d) Result of expression fitting.*

the mouth shape is composed with the global motion and facial expression obtained in the visual tracking step to synthesize the final facial animation (Sec. 5.4).

5.1 Visual Tracking

Our visual tracking component estimates the user's facial motions from online depth and color input in real time. The visual tracking is user-independent, so it works for any input user. We achieve this by using a multilinear model [Cao et al. 2014b]. For any 3D face $\mathbf{M}$, the multilinear model represents it as a set of identity weights $\mathbf{w}^{id}$ and a set of expression weights $\mathbf{w}^{exp}$, denoted as:

$$\mathbf{M} = R\left(C_r \times_2 \mathbf{w}^{id} \times_3 \mathbf{w}^{exp}\right) + T,$$

where C_r is the reduced core tensor. And we use $\mathbf{q} = [R, T]$ to denote the global motion parameters. In our visual tracking, we ask users to start from a neutral expression to perform identity fitting. In particular, we fix $\mathbf{w}^{exp}$ to neutral and estimate 3D head pose $\mathbf{q}$ and $\mathbf{w}^{id}$ iteratively by fitting the multilinear model to the depth (shown in Fig. 3(a,b)) captured by a consumer depth sensor like the new Kinect. The correspondence is obtained by Iterative Closest Point (ICP) [Besl and McKay 1992]. After identity fitting on the first frame, we perform expression fitting for each of the following frames in real time, i.e. we fix $\mathbf{w}^{id}$ and iteratively estimate 3D head pose $\mathbf{q}_t$ and $\mathbf{w}_t^{exp}$. Notice that in the expression fitting, we only fit sparse facial feature points tracked from the color image [Ren et al. 2014] (shown in Fig. 3(c,d)) and map them to the depth image to get the 3D positions. Notice that in the latest literature, there are some visual-based face tracking techniques which refine the user-specified facial expression space by online updates [Li et al. 2013; Bouaziz et al. 2013; Cao et al. 2014a]. We believe these techniques can also be used in our system to pursue better visual tracking results.

The visual tracking technique estimates the overall expressions of the user. However, it is difficult to reconstruct the detailed mouth shape robustly and accurately, especially for situations with occlusion, lighting change and fast motion. As formulated in Sec. 3, we will combine the mouth shape obtained by the visual tracking with audio information to better reconstruct the mouth shape in the following sections. To be consistent with the representation in the database, we still use the *lip performance vector* to represent the speaker-independent mouth shape obtained by the visual tracking. To be specific, for each input frame t, we use $\mathbf{w}_t^{exp}$ and $\mathbf{w}_{neu}^{id}$ to synthesize a mesh and use the pre-defined n vertices around the mouth region. The position of the n vertices is denoted as $\mathbf{v}_t$ for input frame t.

5.2 Audio Processing

Besides color and depth images being recorded online, the audio signal is also recorded by the new Kinect simultaneously. After syncing the recorded audio stream with the images, we estimate the *PSPP vector* for each frame t using the DNN model, denoted as $\mathbf{a}_t$, which will be combined with $\mathbf{v}_t$ to reconstruct the mouth shape of the user in the following section.

5.3 Lip Motion Regression

We use $\{\mathbf{a}_t, \mathbf{v}_t\}$ to reconstruct the mouth shape $\{\mathbf{w}_t\}$ for each input frame t. As mentioned before, the reason we combine $\mathbf{a}_t$ and $\mathbf{v}_t$ is because they each may have ambiguities in determining the mouth shape in certain situations. For example, visual tracking always has failure cases especially when there is occlusion, large lighting change or pose change. Audio-based tracking has no way to predict mouth shape when the user is not speaking. Based on these observations, we propose a confidence-based retrieval scheme, which utilizes the confidence of the visual and audio information to define a combined distance based on the *PSPP vector* and *lip performance vector*, which is further used to extract reasonable samples from the database to synthesize the final mouth shape. In this way, we combine the visual and audio information together and let them help each other to overcome their own drawbacks. Furthermore, the coarticulation effects indicate that the mapping from audio to mouth shapes is not a one-to-one mapping, because with different speech content, the same pronounced sound may be produced using different mouth motions. As a consequence, we extract sub-sequences instead of isolated frames from the database and we extract K nearest sub-sequences (KNN search) as candidates to cover all possible motions for the current frame. By considering the content information, we further distinguish the best sub-sequences for all input frames.

To be specific, we first define a distance measure between an input frame t and a database frame k:

$$d(t,k) = \sum_{j=-f}^{0} \|\mathbf{a}_{t+j} - \mathbf{a}_{k+j}\|_2^2 + \beta \|\mathbf{v}_t - \mathbf{v}_k\|_2^2. \qquad (1)$$

Here, for considering the audio coarticulation, we involve backward neighbors (f is set to 2 for all our experiments) to calculate the distance of *PSPP vectors*. Notice that we cannot involve forward neighbors for real-time applications. In the online stage, β is dynamically changed with the confidence of audio and visual information:

$$\beta = 2.7\, exp(-\mathbf{c}_t^a / \mathbf{c}_t^v).$$

The audio confidence $\mathbf{c}_t^a$ is only decided by a silence detector as:

$$\mathbf{c}_t^a = \begin{cases} 1 & \text{if non-silence} \\ 0 & \text{if silence} \end{cases},$$

for the reason that it is impossible to infer mouth shapes from audio when the user is not speaking. The silence detector is implemented by setting a threshold (0.4 in all the experiments) on the silence element in $\mathbf{a}_t$. The visual confidence $\mathbf{c}_t^v$ is calculated by the fitting error of the mouth features in the visual tracking step. To be specific,

$$\mathbf{c}_t^v = exp(-\sum_{i=1}^{n} \|\mathbf{v}_t^{fit} - \mathbf{v}_t^{depth}\|_2^2), \qquad (2)$$

where $\mathbf{v}_t^{fit}$ denotes the *lip performance vector* defined by the fitting result while $\mathbf{v}_t^{depth}$ denotes the *lip performance vector* defined on the input depth. From our experiments, we see that $\mathbf{c}_t^v$ decreases when there is occlusion, large pose or lighting change.

Eq. 1 is then used to extract candidate frames from the database, followed by collecting their neighbors to form candidate sequences. However, in practice, the time varying β makes it impossible to pre-build KD-trees for the database for fast search. Trading off

Figure 4: *Candidate extraction. Left: Input frame with **a** and **v**. Right: Candidates obtained by the two KD-tree based method (top) and the full search method (bottom). Notice that this frame has low audio confidence ($\beta = 2.7$). The first two candidates of the two KD-tree based method are obtained by visual nearest neighbors, while the third candidate is obtained by audio nearest neighbors.*

performance, we pre-build two KD-trees for all $\mathbf{a}_i$ and $\mathbf{v}_i$ in the database, respectively. In the online stage, for each input frame t, we extract the K nearest frames for both $\mathbf{a}_t$ and $\mathbf{v}_t$, which are denoted as $\{\mathbf{a}_t^k | k = 1, ...K\}$ and $\{\mathbf{v}_t^k | k = 1, ...K\}$. If K is set to a reasonable value ($K = 20$ in all our experiments), the $2K$ candidates should include samples that are close to the input $\{\mathbf{a}_t, \mathbf{v}_t\}$ with respect to Eq. 1. Notice that when the audio and visual confidences are both very low, it is not reasonable to use Eq. 1 to extract candidates. In this case, we simply apply a sequence with neutral expression as the only candidate. Here the silence detector is again used to determine low audio confidence while a threshold (0.5 in all the experiments) for Eq. 2 is used to detect low visual confidence.

To compare the two KD-tree based method and the full search method, we visualize some candidates extracted by the two methods (shown in Fig. 4). We see that the two KD-tree based method extracts reasonable candidates which are the same as the full search method (the first two candidates). Even if it includes some candidates (the last candidate) obtained from a low-confidence signal, the final result is comparable (shown in the accompanying video) because Eq. 1 is still used to choose the best candidate in the following steps.

After the KNN search, we get all candidate sequences for frame t, denoted as

$$\{(\tilde{\mathbf{a}}_t^k, \tilde{\mathbf{v}}_t^k, \tilde{\mathbf{w}}_t^k) | k = 1, ..., 2K\},$$

where

$$(\tilde{\mathbf{a}}_t^k, \tilde{\mathbf{v}}_t^k, \tilde{\mathbf{w}}_t^k) = \{(\mathbf{a}_t^{k-f}, \mathbf{v}_t^{k-f}, \mathbf{w}_t^{k-f}), ..., (\mathbf{a}_t^{k+f}, \mathbf{v}_t^{k+f}, \mathbf{w}_t^{k+f})\}$$

which stands for candidate sequences with $2f + 1$ frames.

The next step is to choose the best candidate sequence for each frame to reconstruct the final mouth shape. To solve this problem, our key observation is that the desired candidate sequences of all input frames should be consistent with each other in their overlapping regions. Based on this observation, we use dynamic programing to chose one candidate sequence for each input frame.

In particular, we construct a graph where each candidate k of frame t is represented by a node $N_{t,k}$. And each node belonging to frame t is connected to all nodes of its neighboring frames $t - 1$ and $t + 1$ by edges. After constructing the graph, we define the distance of an edge as follows:

$$D(N_{t,p}, N_{t+1,q}) = d(t, p) + d(t + 1, q) + \alpha\, dis(\tilde{\mathbf{w}}_t^p, \tilde{\mathbf{w}}_{t+1}^q). \quad (3)$$

Figure 5: *Comparing greedy search with dynamic programming. Top: Ground-truth sequence. Middle: Result of greedy search. Bottom: Result of dynamic programming.*

The first two terms measure how much the two candidate sequences match their corresponding input. $dis(\tilde{\mathbf{w}}_t^p, \tilde{\mathbf{w}}_{t+1}^q)$ measures how much the two candidate sequences are consistent to each other in mouth shape. To measure this, we again use the pre-defined n mesh vertices around the mouth region from the multilinear model. After using the weights $\tilde{\mathbf{w}}_t^p$ and $\tilde{\mathbf{w}}_{t+1}^q$ to reconstruct the position of the vertices on the neutral identity, the average Euclidean distance of the vertices on overlapping frames is used to measure the mouth shape distance:

$$dis(\tilde{\mathbf{w}}_t^p, \tilde{\mathbf{w}}_{t+1}^q) = \frac{1}{2fn} \sum_{j=-f}^{f-1} \sum_{i=1}^{n} \|\mathbf{v}_i(\mathbf{w}_t^{p+j+1}) - \mathbf{v}_i(\mathbf{w}_{t+1}^{q+j})\|_2^2.$$

After defining the distance of each edge, dynamic programming is able to find a path from the first frame to the last frame with minimum total distance among all possible paths in the graph. The parameter α (set to 5 in all our experiments) is used to balance the smoothness and the similarity to the input, which yields a plausible result $\tilde{\mathbf{w}}_t^{DP}$ after dynamic programming. Finally, we use the middle frame in sequence $\tilde{\mathbf{w}}_t^{DP}$, which corresponds to the input frame t, as the output of frame t, denoted as $\mathbf{w}_t^M$.

Dynamic programming is able to find the path with the globally minimum energy. However, it requires processing all the input frames, which is not suitable for our real-time application. To overcome this drawback, we use a greedy search algorithm to replace dynamic programming. To be specific, for the first input frame, after deriving all the candidate sequences, we use the candidate with the minimum value of Eq. 1 as the output. Then for each new input frame, we calculate Eq. 3 with fixed p for the reason that the output of the previous frame is already decided. The candidate q with minimum Eq. 3 among all candidates is taken as the output of the current frame.

The greedy search algorithm seeks a local minimum of the energy which is an approximation of dynamic programming. To measure how accurate the approximation is, we record an input audio accompanied with facial IR makers to reconstruct ground-truth mouth motion, and then we compare the two algorithms though the results of our system. Numerically, the two methods give comparable results (shown in Tab. 1 and Fig. 6). This is consistent with the visual result shown in Fig. 5 and the accompanying video. Both of the

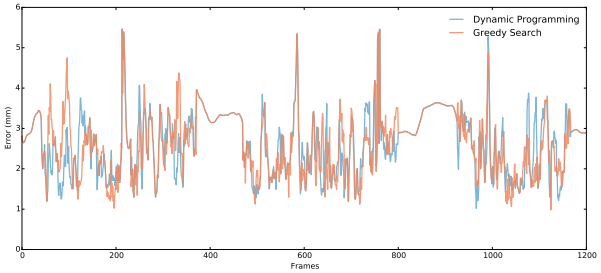


Figure 6: *Error curves of greedy search and dynamic programming. The Error here is the average vertex distance used to calculate Avd in Tab. 1.*

Method	AvE	Avd (cm)	Cor
Greedy search	1.1171	0.2771	0.4404
Dynamic programming	0.9710	0.2874	0.4658

Table 1: *Comparison of greedy search and dynamic programming. AvE measures the average per-frame energy of the obtained path. Avd measures the average vertex distance from the reconstructed result to the ground truth obtained by IR markers. Cor measures the average correlation between the reconstructed vertex trajectory and the ground-truth trajectory.*

algorithms reconstruct motions that are consistent with the input, even though the mouth shapes are still different from the ground truth.

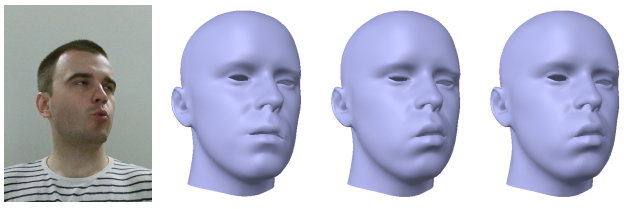
5.4 Result Composition

In this subsection, we use $\mathbf{w}_t^{exp}$ and $\mathbf{w}_t^M$ to synthesize the final output $\mathbf{w}_t$. Ideally, the synthesized face $\mathbf{w}_t$ should match $\mathbf{w}_t^M$ in the mouth region and match $\mathbf{w}_t^{exp}$ in the non-mouth region. This leads to the following linear equation:

$$\mathbf{B}\,\mathbf{w}_t = M\,\mathbf{B}\,\mathbf{w}_t^M + (I - M)\,\mathbf{B}\,\mathbf{w}_t^{exp}.$$

Here, $\mathbf{B}$ is the face expression basis of the multilinear model and M is a $3P \times 3P$ (P is the total number of vertices on the multilinear model) diagonal matrix with 1 denoting the vertices on the mouth region and 0 denoting other vertices. As B and M are predefined and do not change with t, the linear equation can be solved with the following online linear operation:

$$\mathbf{w}_t = \mathbf{B}^{-1}\,M\,\mathbf{B}\,\mathbf{w}_t^M + \mathbf{B}^{-1}\,(I - M)\,\mathbf{B}\,\mathbf{w}_t^{exp}.$$



(a) (b) (c) (d)

Figure 7: *Result Composition. (a) Input image of frame t. (b) Result with $\mathbf{w}_t^{exp}$. (c) Result with $\mathbf{w}_t^M$. (d) Result with $\mathbf{w}_t$.*

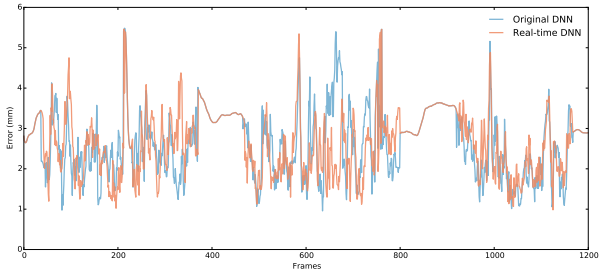


Figure 8: *Error curves of the two DNN models. The Error here is the average vertex distance used to calculate Avd in Tab. 2.*

With $\mathbf{w}^{id}$, $\mathbf{w}_t$ and the global pose $\mathbf{q}_t$, we generate the tracking results given the online visual and audio input as shown in Fig. 7. Also, if we have an avatar with the same defined expression basis, we can transfer $\mathbf{w}_t$ and $\mathbf{q}_t$ to the avatar to generate animations on new characters in real time.

6 Results and Discussion

In this section, we first examine the key components of our system, and then show our results on different speakers, with different speech content and motions. Finally, we discuss the limitations of our system.

Performance Our system is implemented on a computer with a 3.20 GHZ four core CPU, 16G RAM and NVIDIA Geforce GTX 680 graphics card. In the training stage, the real-time DNN model takes 36 hours for training with 309 hours of audio data. Our audio-visual database requires about 1 hour for data recording and about 4 hours for post-processing. In the online stage, our system takes 15-20 ms for estimating the *PSPP vector* and 3ms for estimating the *lip performance vector*. The regression algorithm takes 8ms to get the lip motion and another 1 ms for synthesizing the final result and rendering. So the system runs at about 30 FPS on average.

6.1 Evaluation

We evaluate two key components of our system. The first is our real-time DNN model, which achieves user-independent PSPP estimation in real time. The second is our lip motion regression, which combines audio and visual information together to achieve more accurate and robust facial tracking compared with audio-only and visual-only solutions.

6.1.1 Real-time DNN Model

We have trained a real-time DNN model which estimates the *PSPP vector* of an input audio stream in real time. The phoneme accuracy of our model and the original DNN model is compared in Tab. 2 by $accuracy$. We see that the accuracy of our model drops by 10% compared with the original DNN model. Besides the phoneme accuracy, we also compare the two models on the final output of our system. A numerical comparison is also shown in Tab. 2 by Avd and Cor. Also, the error curves of a short sequence are compared in Fig. 8. Even though the original DNN is slightly better than our DNN on this numerical comparison, judging from the visual comparison shown in our accompanying video, our DNN is comparable to the original DNN. The underlying reason is that our system only utilizes the *PSPP vector* to extract candidates from the database. By adjusting the number of candidates, we can guarantee that reasonable candidates are always extracted by our real-time DNN model,

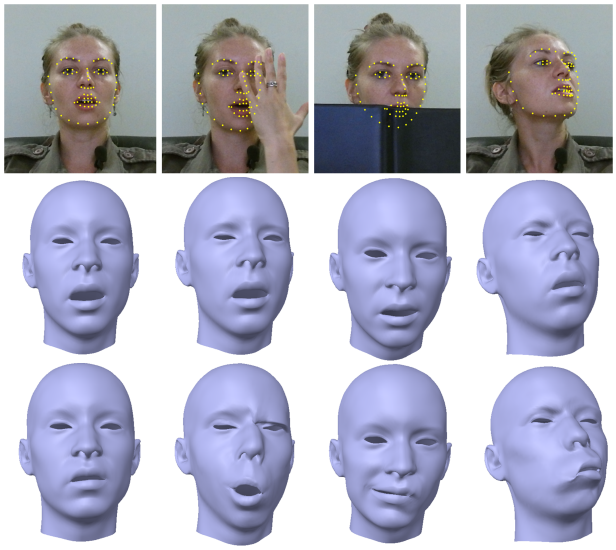


Figure 9: *Comparison with visual-only solution. Top: Reference images. Middle: Results of our solution. Bottom: Results of visual-only solution.*

even though its phoneme accuracy is lower. With our lip motion regression model, reasonable lip motion can still be synthesized.

Method	*Accuracy* (%)	*Avd* (cm)	*Cor*
Original DNN	69.87	0.2673	0.4675
Real-time DNN	59.90	0.2771	0.4404

Table 2: *Comparison of the two DNN models. The phoneme with maximum probability is treated as the estimated phoneme of the two DNN models. Accuracy is calculated with respect to the ground truth. Avd measures the average vertex distance from the reconstructed result to the ground truth obtained by IR markers. Here, we still use the positions of n pre-defined vertices on the mouth region to judge the accuracy of the reconstruction of mouth shapes. Cor measures the average correlation between the reconstructed vertex trajectory and the ground-truth trajectory.*

6.1.2 Lip motion regression

Our method jointly utilizes visual and audio information to determine the lip motion of an input user. Here, we compare it with visual-only and audio-only solutions. Fig. 9 shows cases where visual tracking fails to locate correct feature point positions on images due to fast motion, large occlusion and extreme head pose. In these situations, audio information helps to extract reasonable candidates from the database and the greedy search algorithm chooses from those candidates and synthesizes plausible results. In other situations with normal facial motion and speech, the visual tracking works well and the audio information only slightly improves the final tracking result as shown in Tab. 3 (Visual comparison is shown in the accompanying video.). Here, we do not use facial IR markers to get 3D ground-truth positions as they may lead to an unfair comparison by influencing visual tracking. Instead, we manually define ground-truth landmark positions in the image domain to measure tracking accuracy. In Tab. 3, the improvement is limited because in these situations, the visual information by itself is almost enough to determine the mouth shapes. As speech is determined not only by mouth shape, it only helps to generate plausible mouth shapes but not exactly the ground truth.

On the other hand, the audio-only solution meets difficulties when

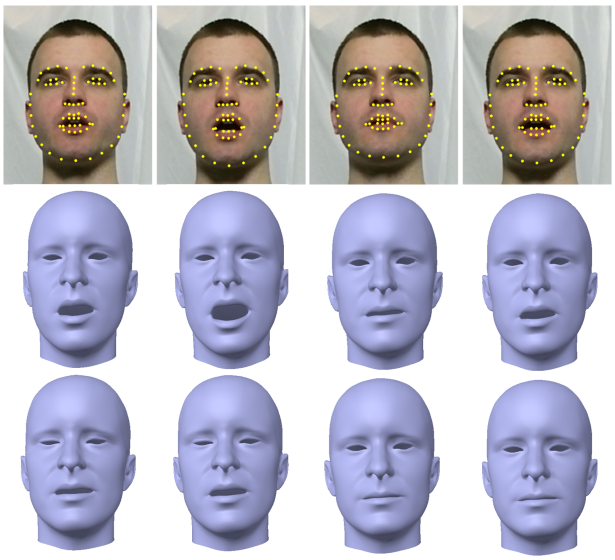


Figure 10: *Comparison with audio-only solution. Top: Reference images. Middle: Results of our solution. Bottom: Results of audio-only solution.*

the speaker is silent. Furthermore, as different people may have different mouth shapes when speaking the same speech content, the audio-only solution can only output a plausible result whose overall motion matches the input speech but not exactly match the ground truth. Fig. 10 shows a sequence result of our solution and the audio-only solution when a user is talking. We see that our solution matches the input better than the audio-only solution. Notice that the leftmost two results are obtained when the user is talking, and the audio-only solution may fail to output accurate mouth shapes sometimes. The rightmost two results are obtained when the user is not talking, so the audio-only solution outputs a closed mouth shape.

To evaluate the effectiveness of jointly using visual and speech features as input in the regression, we compare our method to simply blending audio-only and visual-only results by confidence, as shown in the accompanying video. We see that naive blending leads to less accurate mouth shapes in some frames and temporal flickering caused by sudden confidence change, while our method does not suffer from these kinds of limitations.

Method	*Avd* (pixel)	*Stdev*
Visual-only	4.79	1.46
Audio visual-combined	4.74	1.37

Table 3: *Reconstruction errors in the image domain.*

6.2 Results

In this subsection, we show more results of our system on different users with different speech and motion (Fig 11). To better view our result, please refer to our accompanying video which integrates the input audio tracks. Notice that our real-time demo is recorded in an office with background noise and the users' motion contains large occlusion and extreme poses.

6.3 Limitations

Our system currently has several limitations. First of all, our user-independent visual tracking technique is not the state-of-the-art in the literature. As discussed before, the online update technique [Li

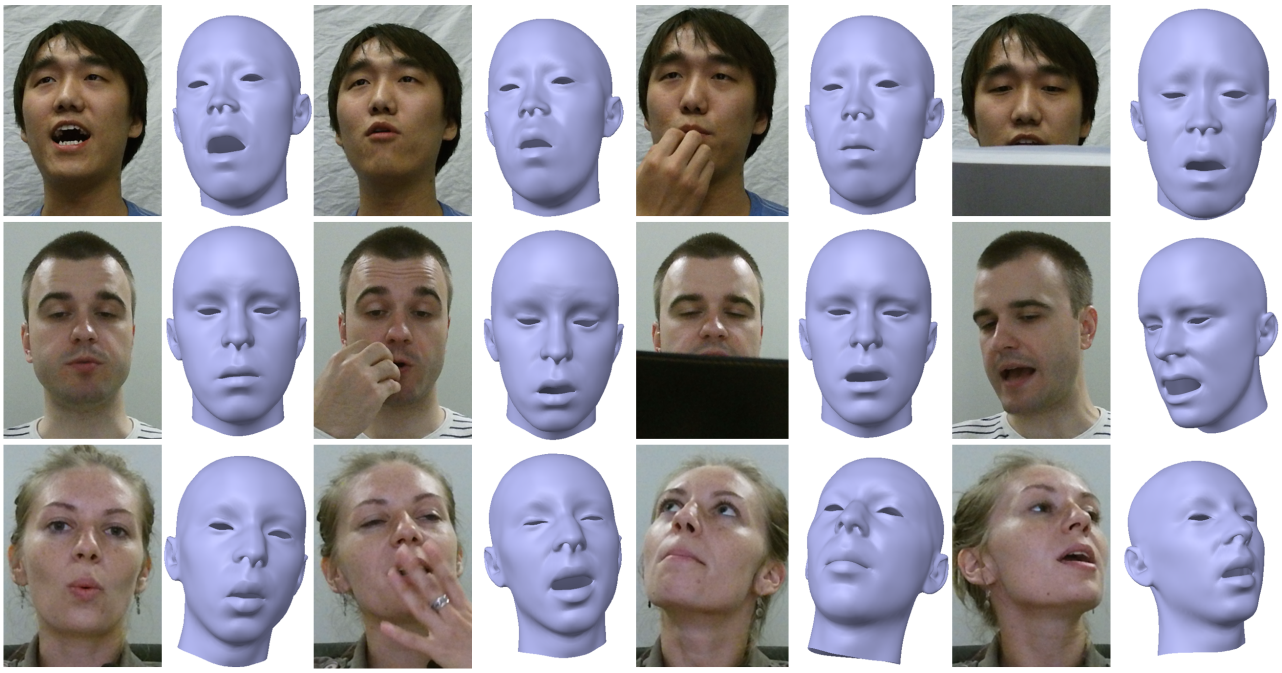


Figure 11: *Results for different users with different speech and motions.*

et al. 2013; Bouaziz et al. 2013; Cao et al. 2014a] can be used here to obtain better visual tracking results, which will definitely improve the accuracy of our whole system. Secondly, we assume that the correlation between audio and mouth shape is user-independent, thus after extracting user-independent visual and audio features, we synthesize the result for any input user from a database of one specific person. However, different people still have their own characteristics when pronouncing the same speech, and these differences are ignored by our technique. Furthermore, we have not considered the correlation between emotion and mouth shape when people talk, so we use a database recorded with a neutral expression. From the results, we have not seen any artifact caused by this. But in theory, involving emotion in building the relationship between speech and mouth shape should be more reasonable. Finally, our system considers audio and visual confidence in the lip motion regression, which gives more robust facial motion tracking compared with audio-only or visual-only solutions. However, when the two kinds of signals are both of low confidence, our system cannot give a correct result, e.g. when there is an extremely large occlusion in the scene and the user is not talking.

7 Conclusion

We propose a facial tracking and animation system for capturing 3D facial performance in real time. Based on a data driven lip motion regressor, our system can reconstruct more accurate 3D mouth motions from combined video and speech audio information and thus is more robust than video based real-time facial animation methods. We also present a real-time speaker-independent DNN based acoustic model for automatically extracting PSPP from a speech audio. We tested our system with live audio and video sequences captured from different actors and also transferred our reconstruction results to other 3D characters.

In future work, we would like to explore other regression methods for modeling the correlation between speech and lip motion, and develop more efficient techniques for mouth shape refinement. Our current system is based on the Kinect sensor. We also want to investigate how to use our method to enhance other video-based real-time facial tracking solutions. Another interesting future directionis to jointly use the captured audio and video information for other speech processing tasks.

Acknowledgements

We wish to thank the reviewers for their constructive feedback, Kai Chen for training the real-time DNN model, and Stephen Lin for proofreading. We also thank the GAPS lab of Zhejiang University for sharing the FaceWarehouse database for our multilinear model training.

References

BEELER, T., HAHN, F., BRADLEY, D., BICKEL, B., BEARDSLEY, P., GOTSMAN, C., SUMNER, R. W., AND GROSS, M. 2011. High-quality passive facial performance capture using anchor frames. *ACM Transactions on Graphics (TOG) 30*, 4, 75.

BESL, P. J., AND MCKAY, N. D. 1992. Method for registration of 3-d shapes. In *Robotics-DL tentative*, International Society for Optics and Photonics, 586–606.

BOTSCH, M., AND SORKINE, O. 2008. On linear variational surface deformation methods. *Visualization and Computer Graphics, IEEE Transactions on 14*, 1, 213–230.

BOUAZIZ, S., WANG, Y., AND PAULY, M. 2013. Online modeling for realtime facial animation. *ACM Transactions on Graphics (TOG) 32*, 4, 40.

BRAND, M. 1999. Voice puppetry. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 21–28.

BREGLER, C., COVELL, M., AND SLANEY, M. 1997. Video rewrite: Driving visual speech with audio. In *Proceedings of the 24th annual conference on Computer graphics and interactive*

techniques, ACM Press/Addison-Wesley Publishing Co., 353–360.

CAO, Y., TIEN, W. C., FALOUTSOS, P., AND PIGHIN, F. 2005. Expressive speech-driven facial animation. *ACM Trans. Graph. 24*, 4 (Oct.), 1283–1302.

CAO, C., WENG, Y., LIN, S., AND ZHOU, K. 2013. 3d shape regression for real-time facial animation. *ACM Trans. Graph. 32*, 4 (July), 41:1–41:10.

CAO, C., HOU, Q., AND ZHOU, K. 2014. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on Graphics (TOG) 33*, 4, 43.

CAO, C., WENG, Y., ZHOU, S., TONG, Y., AND ZHOU, K. 2014. Facewarehouse: a 3d facial expression database for visual computing. *Visualization and Computer Graphics, IEEE Transactions on 20*, 3, 413–425.

CHUANG, E., AND BREGLER, C. 2005. Mood swings: Expressive speech animation. *ACM Trans. Graph. 24*, 2 (Apr.), 331–347.

DAHL, G. E., YU, D., DENG, L., AND ACERO, A. 2012. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on 20*, 1, 30–42.

DENG, Z., NEUMANN, U., LEWIS, J. P., KIM, T., BULUT, M., AND NARAYANAN, S. 2006. Expressive facial animation synthesis by learning speech coarticulation and expression spaces. *IEEE Trans. Vis. Comput. Graph. 12*, 6, 1523–1534.

EZZAT, T., AND POGGIO, T. 2000. Visual speech synthesis by morphing visemes. *International Journal of Computer Vision 38*, 1, 45–57.

FU, S., GUTIERREZ-OSUNA, R., ESPOSITO, A., KAKUMANU, P. K., AND GARCIA, O. N. 2005. Audio/visual mapping with cross-modal hidden markov models. *Multimedia, IEEE Transactions on 7*, 2, 243–252.

GODFREY, J. J., AND HOLLIMAN, E. 1997. Switchboard-1 release 2. *Linguistic Data Consortium, Philadelphia*.

HSIEH, P.-L., MA, C., YU, J., AND LI, H. 2015. Unconstrained realtime facial performance capture. In *Computer Vision and Pattern Recognition (CVPR)*.

KING, S., AND PARENT, R. 2005. Creating speech-synchronized animation. *Visualization and Computer Graphics, IEEE Transactions on 11*, 3 (May), 341–352.

LE, B., MA, X., AND DENG, Z. 2012. Live speech driven head-and-eye motion generators. *Visualization and Computer Graphics, IEEE Transactions on 18*, 11 (Nov), 1902–1914.

LEI, X., DONGMEI, J., RAVYSE, I., VERHELST, W., SAHLI, H., SLAVOVA, V., AND RONGCHUN, Z. 2003. Context dependent viseme models for voice driven animation. In *Video/Image Processing and Multimedia Communications, 2003. 4th EURASIP Conference focused on*, vol. 2, IEEE, 649–654.

LI, H., YU, J., YE, Y., AND BREGLER, C. 2013. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graph. 32*, 4, 42.

MASSARO, D. W., BESKOW, J., COHEN, M. M., FRY, C. L., AND RODGRIGUEZ, T. 1999. Picture my voice: Audio to visual speech synthesis using artificial neural networks. In *AVSP'99-International Conference on Auditory-Visual Speech Processing*.

RABINER, L. R., AND JUANG, B.-H. 1993. *Fundamentals of speech recognition*, vol. 14. PTR Prentice Hall Englewood Cliffs.

REN, S., CAO, X., WEI, Y., AND SUN, J. 2014. Face alignment at 3000 fps via regressing local binary features. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, IEEE, 1685–1692.

SEIDE, F., LI, G., CHEN, X., AND YU, D. 2011. Feature engineering in context-dependent deep neural networks for conversational speech transcription. In *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, IEEE, 24–29.

SEIDE, F., LI, G., AND YU, D. 2011. Conversational speech transcription using context-dependent deep neural networks. In *Interspeech*, 437–440.

SUN, N., SUIGETSU, K., AND AYABE, T. 2008. An approach to speech driven animation. In *Intelligent Information Hiding and Multimedia Signal Processing, 2008. IIHMSP'08 International Conference on*, IEEE, 113–116.

TAYLOR, S. L., MAHLER, M., THEOBALD, B.-J., AND MATTHEWS, I. 2012. Dynamic units of visual speech. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '12, 275–284.

WAMPLER, K., SASAKI, D., ZHANG, L., AND POPOVIĆ, Z. 2007. Dynamic, expressive speech animation from a single mesh. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '07, 53–62.

WANG, G.-Y., YANG, M.-T., CHIANG, C.-C., AND TAI, W.-K. 2006. A talking face driven by voice using hidden markov model. *Journal of information science and engineering 22*, 5, 1059.

WEISE, T., BOUAZIZ, S., LI, H., AND PAULY, M. 2011. Real-time performance-based facial animation. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2011) 30*, 4 (July).

XIE, L., AND LIU, Z.-Q. 2007. A coupled hmm approach to video-realistic speech animation. *Pattern Recognition 40*, 8, 2325–2340.

YU, D., DENG, L., AND DAHL, G. 2010. Roles of pre-training and fine-tuning in context-dependent dbn-hmms for real-world speech recognition. In *Proc. NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

ZHANG, X., WANG, L., LI, G., SEIDE, F., AND SOONG, F. K. 2013. A new language independent, photo-realistic talking head driven by voice only. In *INTERSPEECH*, 2743–2747.

ZHUANG, X., WANG, L., SOONG, F., AND HASEGAWA-JOHNSON, M. 2010. A minimum converted trajectory error (mcte) approach to high quality speech-to-lips conversion.