A Deep Learning-Based Model for Head and Eye Motion Generation in Three-party Conversations

AOBO JIN, QIXIN DENG, YUTING ZHANG, and ZHIGANG DENG, University of Houston

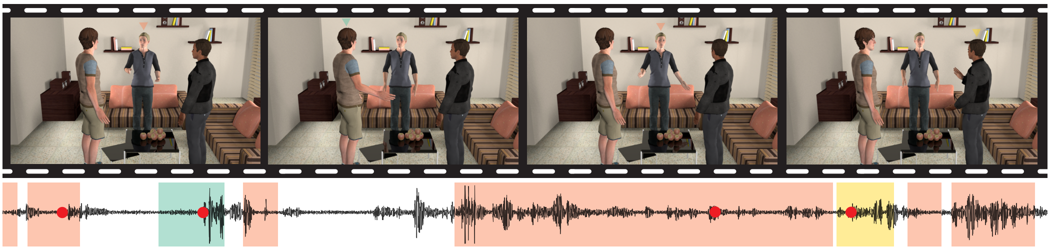


Fig. 1. Three-party conversational animations synthesized by our approach based on novel speech input. Green blocks denote the left interlocutor's speaking time, pink blocks denote the middle interlocutor's speaking time, and yellow blocks denote the right interlocutor's speaking time, and red solid circles denote the four selected frames shown in the top.

In this paper we propose a novel deep learning based approach to generate realistic three-party head and eye motions based on novel acoustic speech input together with speaker marking (i.e., speaking time for each interlocutor). Specifically, we first acquire a high quality, three-party conversational motion dataset. Then, based on the acquired dataset, we train a deep learning based framework to automatically predict the dynamic directions of both the eyes and heads of all the interlocutors based on speech signal input. Via the combination of existing lip-sync and speech-driven hand/body gesture generation algorithms, we can generate realistic three-party conversational animations. Through many experiments and comparative user studies, we demonstrate that our approach can generate realistic three-party head-and-eye motions based on novel speech recorded on new subjects with different genders and ethnicities.

CCS Concepts: • **Computing methodologies** → **Animation**.

Additional Key Words and Phrases: gaze synthesis, multi-party conversation, speech-driven animation, conversational gesture, head motion, multi-agents system

ACM Reference Format:
Aobo Jin, Qixin Deng, Yuting Zhang, and Zhigang Deng. 2019. A Deep Learning-Based Model for Head and Eye Motion Generation in Three-party Conversations. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 2, Article 9 (July 2019), 19 pages. https://doi.org/10.1145/3340250

Authors' address: Aobo Jin, jinaobo1103@gmail.com; Qixin Deng, qxdeng1991@gmail.com; Yuting Zhang, zhangyuting2015@gmail.com; Zhigang Deng, zdeng4@uh.edu, University of Houston.

2577-6193/2019/7-ART9 $15.00
https://doi.org/10.1145/3340250

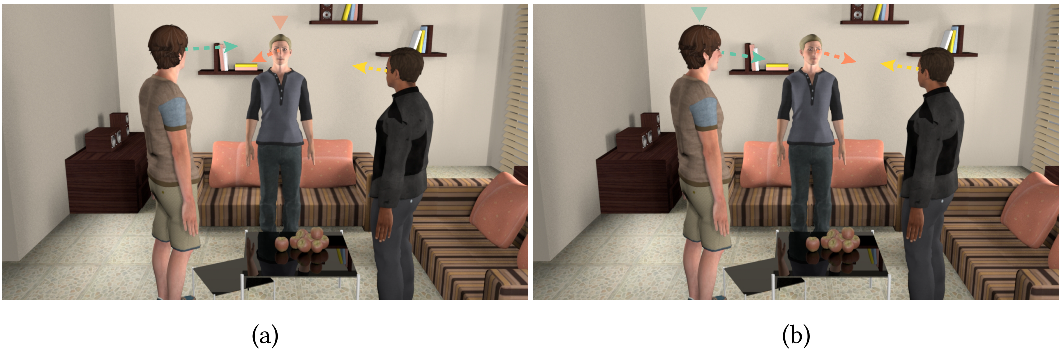


Fig. 2. (a): The middle interlocutor is speaking to the left interlocutor at this frame, both of the two listeners show their interest by looking at the speaker. The triangle above the head points to the current speaker, and the dashed directional lines represent the DFocs of the interlocutors. (b): The left interlocutor is speaking but not looking at any other interlocutor. Meanwhile, the two listeners look at each other and do not show noticeable interest in the speaker.

1 INTRODUCTION

Group conversation is probably one of the most common human-human communication forms in our society. Over the years, its interaction patterns, pathologies, and paradoxes have been extensively studied in many fields including communication, psychology, and behavior sciences [Watzlawick et al. 2011]. Meanwhile, in recent years computationally understanding and modeling multi-party conversations have also attracted increasing attention in the fields of human-computer interaction(HCI) [Ding et al. 2017; Gu and Badler 2006; Otsuka et al. 2007, 2005; Vertegaal et al. 2000] and human-robot interaction [Foster et al. 2012; Johansson et al. 2013; Kondo et al. 2013; Matsuyama et al. 2010; Mutlu et al. 2009] due to its broad applications.

To date relatively few efforts have been attempted to *synthesize* realistic *multi-party* head-and-eye animations, not to mention the whole three-party conversational animations, despite such animation techniques can be potentially applied to a variety of applications, including virtual reality, computer-mediated communication, teleconferencing, and conversational agents. For example, previously researchers have developed a number of approaches to generate conversational hand gestures and other gesture modalities such as facial expression and body movements, using either rule-based algorithms [Cassell et al. 1994, 2001; Marsella et al. 2013] or data-driven schemes [Levine et al. 2010, 2009; Stone et al. 2004]. These approaches demonstrated certain successes of generating the gestures of a single virtual speaker or dyadic conversational animations. However, whether and how they can be effectively applied or extended to generate realistic multi-party conversational head-and-eye animations has not been established. On the other hand, generating realistic multi-party head-and-eye animations from limited user input (e.g., speech) is technically challenging, due to the following main reasons. First, the interaction patterns of multi-party (even three-party) conversations are significantly more complex than the counterparts in dyadic conversations [Goodwin and Heritage 1990; Watzlawick et al. 2011].

Second, regardless the employed algorithms, the generation of realistic multi-party conversational head-and-eye animations would need a delicately recorded, multi-party conversational motion dataset. However, acquiring and processing such a non-trivial dataset itself is unavoidably time-consuming and expensive.

To tackle the above challenges, in this paper we propose a novel deep learning based approach to generate realistic three-party head-and-eye animations based on novel acoustic speech input together with speaker marking. Instead of focusing on the animation generation for any forms of multi-party conversations that could involve with a varying number of interlocutors, our work specifically focuses on the head-and-eye animation synthesis of *three-party conversations*, because: i) three-party conversation is the simplest form of multi-party conversations, and ii) three-party conversations have all the necessary ingredients of multi-party conversations; an effective methodology for three-party conversational animation synthesis would provide a promising first step towards the generation of various general forms of multi-party conversational animations.

Specifically, in our approach we first acquire a high quality, three-party conversational motion dataset. The dataset is multi-modal, since we simultaneously record the facial expressions, head movements, hand gestures, torso movements, and acoustic speech of all the interlocutors in three-party conversations. Then, to generate plausible, speech-synchronized interaction snapshots (an interaction snapshot encodes the combined torso-head-eye directions of all the interlocutors at a particular frame), we train a deep learning based model based on the above acquired dataset. The trained model can automatically predict the interaction snapshots based on speech input as well as speaker marking. After that, we further refine each interaction snapshot by deriving the corresponding eye motion and head/torso motion from the obtained *Direction-of-Focus* (DFoc) of each interlocutor (i.e., DFoc is an individual concept.), which is enclosed in the interaction snapshot. In a nutshell, the DFoc is the combination of torso-head-eye directions. Since the torso direction is represented in the world coordinate system, head direction is defined in the torso coordinate system, and eye rotation angle is based on the head coordinate system, we need to transform them into the same coordinate system to compute the DFoc. For example, if an individual turns the head 10° to left without both torso movement and eye movement, we treat both torso rotation and eye rotation as zero. Furthermore, via the combination of existing lip-sync and speech-driven hand/body gesture generation algorithms, our approach can generate realistic three-party conversational animations. Through many experiments and comparative user studies, we demonstrate that our complete pipeline can generate realistic three-party conversational animations based on novel speech recorded on new subjects with different genders and ethnicities. Figure 1 shows the input and output of our animation synthesis system. Figure 2 shows two examples of interaction snapshots in a synthesized three-party conversation.

The main contributions of this work include:

- A new speech feature expression for training LSTM model to predict the interaction snapshots; and
- A deep learning based approach to automatically synthesize the interaction snapshots that enclose the DFocs of all the interlocutors in three-party conversations, based on novel acoustic speech input together with speaker marking.

2 RELATED WORK

In this section, we briefly review recent efforts that are most related to our work, including conversational gaze synthesis and head motion synthesis. For a comprehensive review on facial animation synthesis, please refer to recent surveys [Deng and Noh 2008; Lewis et al. 2014; Ruhland et al. 2014].

Conversational Gaze Synthesis. Researchers have pursued a variety of efforts on gaze synthesis [Ruhland et al. 2014]. As pointed out in the existing literature [Lee et al. 2002; Vertegaal et al. 2001], humans typically have different gaze patterns depending on their conversational states such as speaking or listening. As one of the early works in the field, Lee et al. [2002] statistically analyze pre-recorded gaze data of a subject during speaking/listening and then further synthesize novel saccadic movements using the first-order statistics. Vinayagamoorthy et al. [2004] proposed a computational gaze model for two avatars engaging in a dyadic interaction in virtual environment, based on a pre-collected face-to-face gaze dataset. Thiebaux et al. [2009] proposed a gaze controller to realize various manners of gaze through a rich set of input parameters to produce desired expressive gaze animations. Steptoe et al. [2010] proposed a parametric model for both eyelids and blinks based on the reported findings from other fields such as ophthalmology and psychology. Later, along the data-driven direction, Deng and colleagues [Deng et al. 2005; Le et al. 2012; Ma and Deng 2009] developed a series of statistical gaze models from recorded conversational gaze data, including the statistical modeling of gaze-head coupling [Le et al. 2012; Ma and Deng 2009]. In addition, researchers have also used the visual attention on virtual environment to guide the generation of plausible gazes on avatars [Gu and Badler 2006; Khullar and Badler 2001; Pejsa et al. 2016; Peters and O'Sullivan 2003].

Head Motion Synthesis. Many previous works have been proposed to generate realistic head motion on avatars [Munhall et al. 2004]. For example, Graf et al. [2002] studied the conditional probabilities of pitch accents accompanied by primitive head movements such as nodding. Realizing the importance of high quality data for head motion generation, Chuang and Bregler [2005] first create a database of head motion sequences indexed by pitch features and then synthesize new head motions through global optimization that maximizes the match of pitch features in the database. Along this line, based on a pre-recorded audio-visual dataset, researcher have trained different forms of Hidden Markov models (HMMs) for head motion generation [Busso et al. 2007, 2005; Lee and Marsella 2009; Sargin et al. 2008]. Later, Le et al. [2012] developed a Gaussian Mixture Model to learn the probability distribution between kinematic features of head motion (e.g., Euler angles, and angular velocities) and the prosody features of speech for the effective generation of head motion frame-by-frame. Different from the above works that are focused on head movements on talking avatars, Gratch and colleagues developed statistical models to learn a set of rules from an annotated conversation dataset in order to generate gestures including head movements on listening agents [Gratch et al. 2006; Maatman et al. 2005]. In addition, how to effectively synchronize head movement with gaze has also been explored previously [Le et al. 2012; Ma and Deng 2009; Masuko and Hoshino 2006]. Notably, Marsella et al. [2013] proposed a rule-based animation method to generate virtual conversations between two parties, driven by speech as well as annotated texts as the input. However, although their method can generate animated performance for a dyadic conversation, how to apply or extend it for the generation of virtual conversational animations involved with three or more parties has not been demonstrated.

3 OUR APPROACH

Since the turn information is provided as one of the inputs, automatically generating speech-synchronized motions of the eyes, head, and torso for each interlocutor at each frame is the main task. In this work, we solve the synthesis of the above coordinated movements as a two-steps process, by introducing a novel middle-level concept, *interaction snapshots*. Specifically, (i) at the first step, based on our acquired three-party conversational motion dataset (§4), we train two Long Short-Term Memory (LSTM) models, given our novel expression of input speech features: one for the speaker and the other for listeners, to generate an interaction snapshot at current frame (§5), using the interaction snapshot at the previous frame and the acoustic speech features at the current frame as the inputs. Our listener LSTM model implicitly learns the patterns that typically occur in three-party conversations (refer to Figure 4), and such patterns cannot be modeled by any

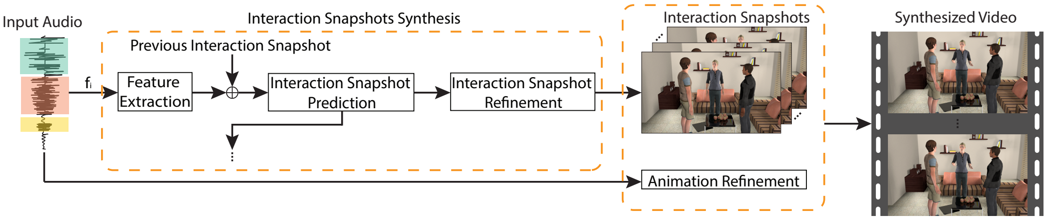

Input Audio
Interaction Snapshots Synthesis
Previous Interaction Snapshot
f_i
Feature Extraction
Interaction Snapshot Prediction
Interaction Snapshot Refinement
Interaction Snapshots
Synthesized Video
Animation Refinement

Fig. 3. Schematic view of our approach. It generates interaction snapshots frame by frame. This figure uses the generation of the i-th frame as an example. The system consists of two main components: interaction snapshots synthesis, and animation refinement. The color blocks on the input audio mark the speaking time of different interlocutors: the pink block for the left interlocutor, the green block for the middle interlocutor, and the yellow block for the right interlocutor.

two-party listener models. (ii) At the second step, we introduce a data-driven scheme to automatically refine each interaction snapshot (enclosing the DFoc of every interlocutor) obtained at the first step by decomposing the DFoc of each interlocutor into two parts (§5.3): *eye gaze*, and the *head-torso combined movement*. The latter is further split into head movement and the upper body movement later. Figure 3 illustrates the schematic view of our approach.

It is noteworthy that, the difference between the interaction snapshot and DFoc in this work is: the former is a collective concept, representing the simultaneous DFocs of all the three interlocutors at a particular time instant (or frame), while the latter is an individual concept. For example, the DFoc of an individual is his/her combined torso-head-and-eye direction at a particular time instant (or frame). In other words, the relation between an interaction snapshot and the DFoc of an individual (enclosed in this interaction snapshot) is the same as that between a member and a set that includes this member.

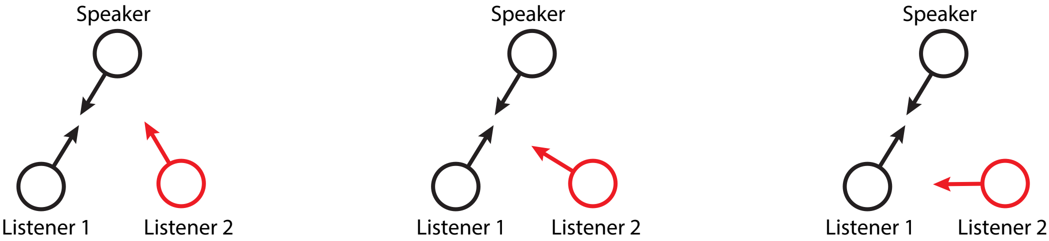


Fig. 4. Top views of some interaction snapshots that could occur in three-party conversations. The interaction snapshot in the left can be synthesized with existing two-party listener models. However, other two interaction snapshots can only be synthesized by our approach.

4 DATA ACQUISITION AND PROCESSING

To acquire high-quality three-party conversational motion data that need to be used in this work, we specifically built an in-house, hybrid data acquisition system, described below. A ten-cameras VICON optical motion capture system was installed in a controlled lab environment to capture the motions of three interlocutors. Each interlocutor was asked to wear a mocap suit attached with optical markers to record their upper body motions, and facial markers were also glued on their faces (see Figure 5). Three Canon HD cameras were used to record the close-up facial video of the three interlocutors, respectively. Their voice was recorded by a wired microphone in the middle

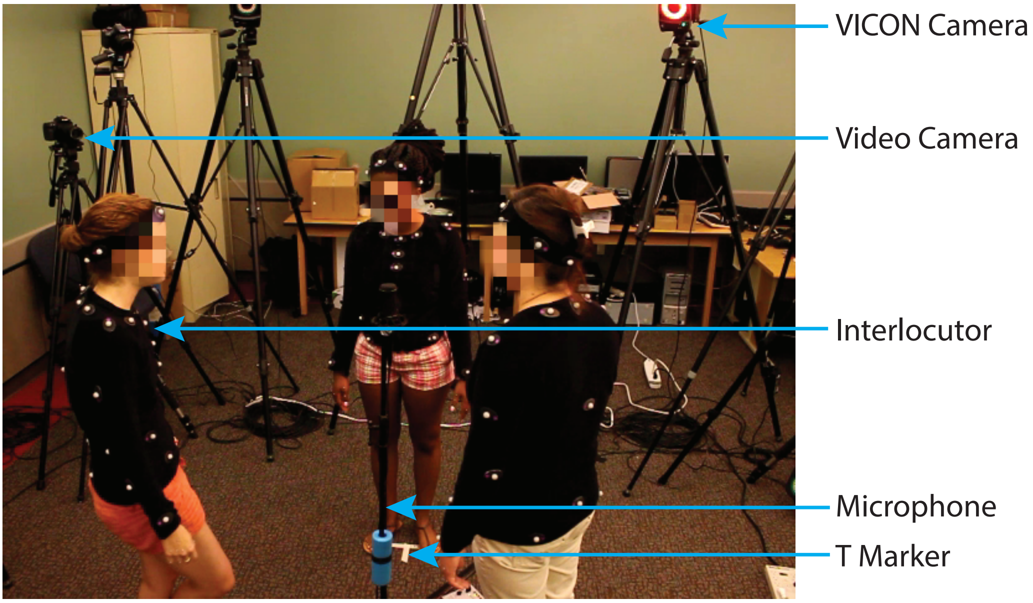


Fig. 5. A snapshot of the in-house built, data acquisition system for three-party conversation motion capture.

of them. All the acquired data (i.e., 3D mocap data, video data, and speech data) were temporally aligned and later down-sampled to 30 frames per second. Note that the originally recorded mocap data is 120 frames per second, and video data is 30 frames per second. In addition, from the recorded 3D mocap data, we calculated the joint angles of the upper skeleton for each interlocutor, with the aid of inverse kinematics. In this work, we recorded three-party conversations of two groups: the first group contains three males and the second contains three females.

During the data acquisition, three interlocutors were instructed to stand at the three places of T markers, formed an equilateral triangle with a distance of 1 meter to each other. Before the recording of each session, interlocutors were asked to stay at their original locations (i.e., the T markers) as much as possible to minimize the influence of the lower body on the motion of the upper body; however, they were allowed to naturally move their upper bodies, hands, and heads during the conversation. They were instructed to choose any topics of interest for group discussion in a normal speed, with natural non-verbal behaviors. To this end, we recorded a total of 10 three-party conversation sessions: 5 for three males and 5 for three females, each of which lasts from 8 to 10 minutes. The total number of frames in our dataset is about 144K after the dataset was re-sampled to 30 frames per second. At the post-processing step, we first removed those sporadic conversational segments that contain overlapping speech (i.e., two or more interlocutors are speaking simultaneously), since speech data was recorded by the microphone placed in the middle of three interlocutors and we cannot accurately extract each speaker's speech in the case of overlapping speech.

In order to collect natural eye motions during conversations, all the interlocutors did not wear eye tracking devices. Instead, from the recorded close-up face video of each interlocutor, we recovered his/her 3D eye gazes through a combination of two existing methods [Le et al. 2012; Wang et al. 2016]. Our work does not consider eyelid motion, so we did not collect any eyelid motion data.

Also, we predicted the gaze directions using the gaze tracking algorithms [Le et al. 2012; Wang et al. 2016] when the eyes are closed.

Based on the recorded acoustic speech, its speech transcription, timing, and the speaking time of each interlocutor were obtained manually by a language expert with the aid of the Praat tool [1]. We also used Praat to extract the pitch and intensity features from acoustic speech. For a small percentage of frames that do not have pitch values, we generated uniformly distributed random values to fill the gap.

5 INTERACTION SNAPSHOTS SYNTHESIS

As aforementioned, the core element of our approach, which is also the key contribution of this work, is to synthesize interaction snapshots and further derive the eye gazes and head/torso movements of all the interlocutors in a three-party conversational animation. Which speech features are good for the training of a machine learning model has always been a non-trivial, widely open problem. In this work, we design new speech feature representations (described below) to train LSTM models, and then we describe how to synthesize an interaction snapshot at each frame using the trained LSTM models.

5.1 Data Representation for LSTM Training

To train the LSTM model, we first need to transform our data to representations that are suitable for LSTM training. For acoustic speech, we use its pitch, P, and intensity, I, as its features with a window size of 33.3 ms. Instead of adopting MFCC features that have been widely used in speech recognition and lip-sync applications, we choose pitch/intensity features since prosodic features (including pitch, intensity, etc.) have been demonstrated their strong relevance to head motion in the literature [Busso et al. 2007, 2005; Chuang and Bregler 2005].

In this work, instead of directly using the pitch and intensity at current frame, we use their differences between the current frame and the previous frame (i.e., $P_i - P_{i-1}$, and $I_i - I_{i-1}$) as the speech features for LSTM training. This is inspired by random forest models, where the difference of image features is often used to eliminate the illumination effect.

Also, we further decompose the calculated DFoc of an interlocutor into three Euler angles (i.e., yaw α, pitch β, and roll γ). Since in three-party conversation scenarios, γ typically stay at zero. Therefore, in this work we ignore γ and focus on α and β.

In addition, since we need to model and generate the DFocs of the three interlocutors (one is the speaker, and the other two are listeners) that together form an interaction snapshot, we will train two different types of LSTM models: one for the speaker (called the *speaker LSTM model*), and the other for the listeners (called the *listener LSTM model*).

To train the speaker LSTM model, the input feature vector at a data frame i is the following 4-dimensional vector:

$$X_i = (P_i - P_{i-1}, I_i - I_{i-1}, \alpha_{i-1}, \beta_{i-1}) \tag{1}$$

The first two elements in the above Eq. 1 are the pitch and intensity of the speech between the current frame and the previous frame, respectively, and the last two elements are the two rotational angles of interest at the previous frame $i - 1$.

By contrast, to train the listener LSTM model, the input feature vector at frame i is the following 6-dimensional vector.

[1] http://www.fon.hum.uva.nl/praat/

$$X_i = \begin{cases} (P_i - P_{i-1}, I_i - I_{i-1}, 0, 0, \alpha_{i-1}, \beta_{i-1}) & \text{speaker on left} \\ (0, 0, P_i - P_{i-1}, I_i - I_{i-1}, \alpha_{i-1}, \beta_{i-1}) & \text{speaker on right} \end{cases} \tag{2}$$

In the above Eq. 2, if the speaker is on the left side of this listener, we add two zero elements after the two speech feature elements; otherwise, we add them before the speech feature elements. Note that the three interlocutors stood at the vertices of an equilateral triangle during data capture (refer to §4). Therefore, from the perspective of a listener, the speaker would be either on the left or on the right.

For the above input feature vector at a training frame i, its corresponding expected output (also called the *label* vector in this work) is a 2-dimensional vector: $Y_i = (\alpha_i, \beta_i)$ of the speaker (for the speaker LSTM model) or of the listener (for the listener LSTM model). Here (α_i, β_i) represents the DFoc of this particular interlocutor at current frame.

We further carefully chopped the pre-recorded long conversation data into short subsequences with the same length. The duration of each subsequence is 1 second (i.e., 30 frames). To this end, we obtained 3033 subsequences for the speaker LSTM model, and 4421 subsequences for the listener LSTM model. Note that we do not differentiate the two listeners at any moment, which enables us to combine their data together in this work.

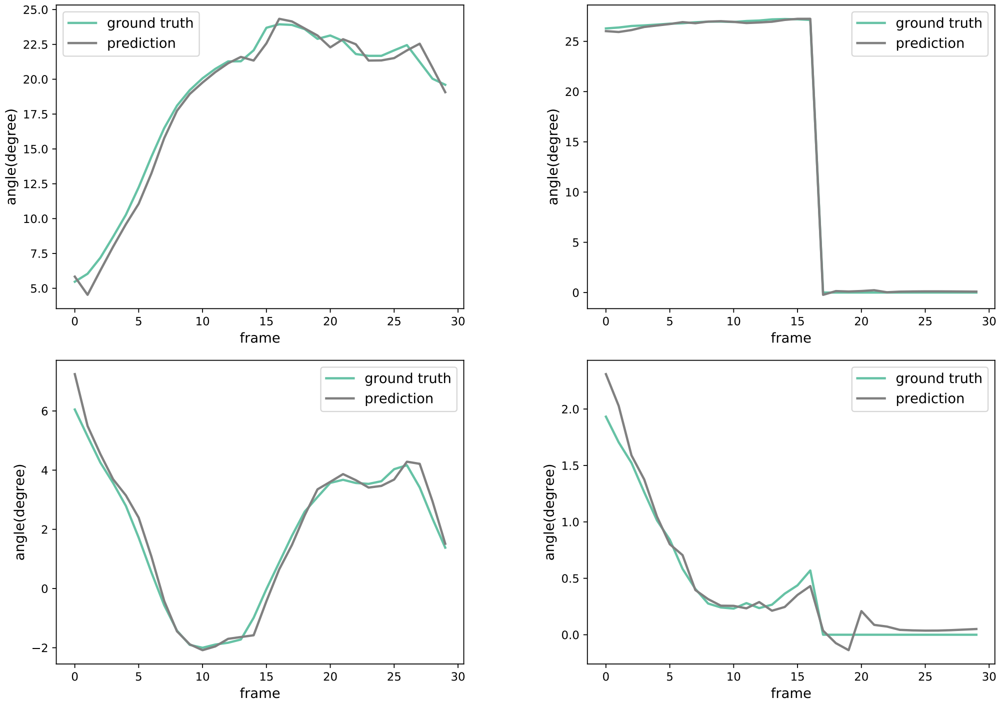
5.2 LSTM Models

LSTM is a recurrent neural network (RNN) model designed to handle sequence problems [Hochreiter and Schmidhuber 1997]. There are three gates in a LSTM cell: a forget gate f, an input gate i, and an output gate o. By controlling these gates, LSTM can learn to remember some important information and forget some unnecessary information from the previous state. LSTM has a major advantage to handle long sequences that a standard RNN cannot handle. During the back-propagation, the standard RNN typically has a vanishing or exploding gradients problem. With the three gates, LSTM can better handle these problems.

The LSTM model can have different formats, mainly classified as four types: one-to-one, one-to-many, many-to-one, and many-to-many. Two different types exist in many-to-many LSTM: in the first one, the length of the input sequence is the same as that of the output one; in the other one, the length of the input sequence does not need to be the same as that of the output. Since in this work we have a feature vector (input) and a label vector (output) at each data frame, so we choose many-to-many LSTM as our model.

In this work, we employ a one-layer LSTM instead of a two-layers LSTM, since we found the one-layer LSTM can generate dresired results, and the training efficiency of the one-layer LSTM is much more faster than that of the two-layers LSTM. We randomly split the dataset to the training data (90%) and the test data (10%) for both the speaker LSTM model and the listener LSTM model. In the trained LSTM model, the number of units is set to 512 for each layer. In order to overcome the over-fitting problem, the dropout rate is set to 0.5. The start learning rate for the model is 0.001 using Adam optimizer, and the decayed learning rate is used, which is 0.95 for every 2000 steps. In order to train the LSTM model, normalization also need to be done on both the feature vector sequences and the label vector sequences. For all the sequences, we divide the 95% max value for each dimension to make the range of the training data from -1 to 1 and remove outliers. The maximal number of training steps is set to 40000 for both the speaker LSTM model and the listener LSTM model. We trained both of the LSTM models on GPU to speed-up the efficiency.

After the two LSTM models are trained, we use the 10% retained test data to evaluate their effectiveness and accuracy. Figure 6 shows the experiment results when the trained LSTM models are applied to two randomly chosen test sequences for the speaker LSTM model and the listener LSTM model, respectively. As shown in this figure, the predicted DFocs by our LSTM models are



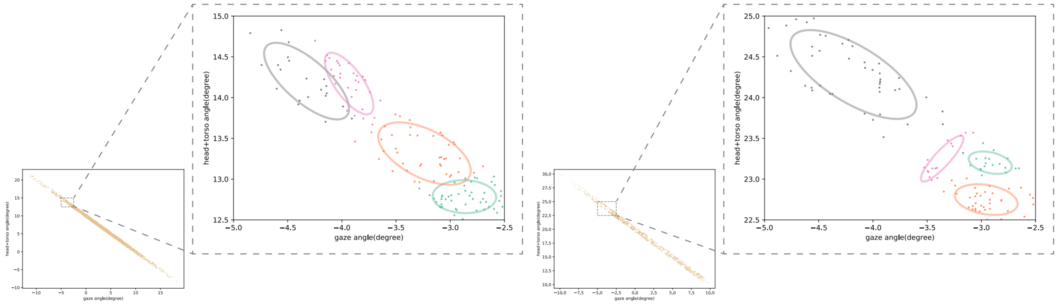
(a)

(b)

Fig. 6. (a) A test result for the speaker LSTM model: the top panel is the yaw angle α, the bottom is the pitch angle β, and (b): A test result for the listener LSTM model, the top panel is the yaw angle α, the bottom panel is the pitch angle β.

reasonably close to the ground-truth. Also, at frame #17 and after, there is a clear discrepancy between the top and the bottom panels in Figure 6(b). The main reason is, the LSTM model can somehow "remember" previous states and still utilize the previous states for prediction. In the case of Figure 6(b), the input speech becomes silent since frame #17. During the LSTM model training, we labeled the corresponding output as zeros for the silence speech part. Therefore, in the bottom panel of Figure 6(b), the LSTM model continues to use the remembered previous states for prediction, instead of immediately stopping at zero. By contrast, in the top panel of Figure 6(b), its remembered previous states before frame #17 have been flat already and thus its predictions are flat at frame #17 and after.

In addition, for each of the two LSTM models, we ran all the test sequences and then computed the mean square error (MSE): 0.83 and 0.57 for the yaw and pitch angles (in degree), respectively, in the case of the speaker LSTM model; 0.41 and 0.43 for the yaw and pitch angles (in degree), respectively, in the case of the listener LSTM model.



(a)

(b)

Fig. 7. (a): Samples in a bin (10°-11°) of the Yaw angle α, and (b): Samples in a bin (19°-20°) of the Pitch angle β.

5.3 Interaction Snapshots Refinement

After the above interaction snapshot at current frame is predicted, we need to further decompose the predicted DFoc of each interlocutor into (i) eye gaze, (ii) head movement, and (iii) torso movement before transferring these movements to 3D human models. Our strategy is to first decompose a DFoc to (1) eye gaze and (2) the head-torso combined movement, and then the latter is further split into head movement and the torso movement.

Specifically, we decompose each predicted DFoc in the space of the yaw angle α and in the space of the pitch angle β, respectively, using the same methodology (described below), since the yaw/pitch angles can be regarded independent of each other. Recall that in §5.1, the DFoc of each interlocutor at every frame of our acquired dataset has been converted to the representation of (α, β). Actually, when computing (α, β) at each data frame for each interlocutor, we also pre-computed and saved its corresponding yaw and pitch angles for the eyes and the head+torso combination, respectively: (α_e, β_e) and (α_{h+t}, β_{h+t}).

Quantization: Without the loss of generality, take the decomposition of α as an example, we first quantize the α angles of all the frames in our dataset to a number of bins. The bin size is 1° and the total number of bins is 60 (i.e., −30° to 30°, determined by the min/max among all the α angles in our dataset). In this way, each bin includes the α angles of some frames, called *α samples* in this writing. Specifically, each α sample is a two-dimensional vector (α_e, α_{h+t}).

GMM fitting for each bin: Then, we use a Gaussian Mixture Model (GMM) to fit all the α samples in each bin B_i (examples are shown in Figure 7). After that, given any (α_e, α_{h+t}) we can straightforwardly calculate its probability based on the fitted GMM: A higher probability value means this specific combination (α_e, α_{h+t}) has a higher chance to occur to achieve the combined α angle in B_i; and vice versa.

Decomposition: Now we describe how to decompose any predicted (α^k, β^k) at frame k into its corresponding eye and head+torso parts: (α_e^k, β_e^k) and (α_{h+t}^k, β_{h+t}^k). We use the decomposition of α^k as a specific example and the decomposition of β^k is done similarly. Since such a decomposition at the previous frame k-1 (i.e., (α_e^{k-1}, α_{h+t}^{k-1})) has already been obtained, we need to compute this decomposition at current frame k as follows. First, α^k is mapped to a specific bin (assuming the mapped bin is B_i) using the above same quantization step. Then, we use the following formula to select the optimal (α_e^k, α_{h+t}^k):

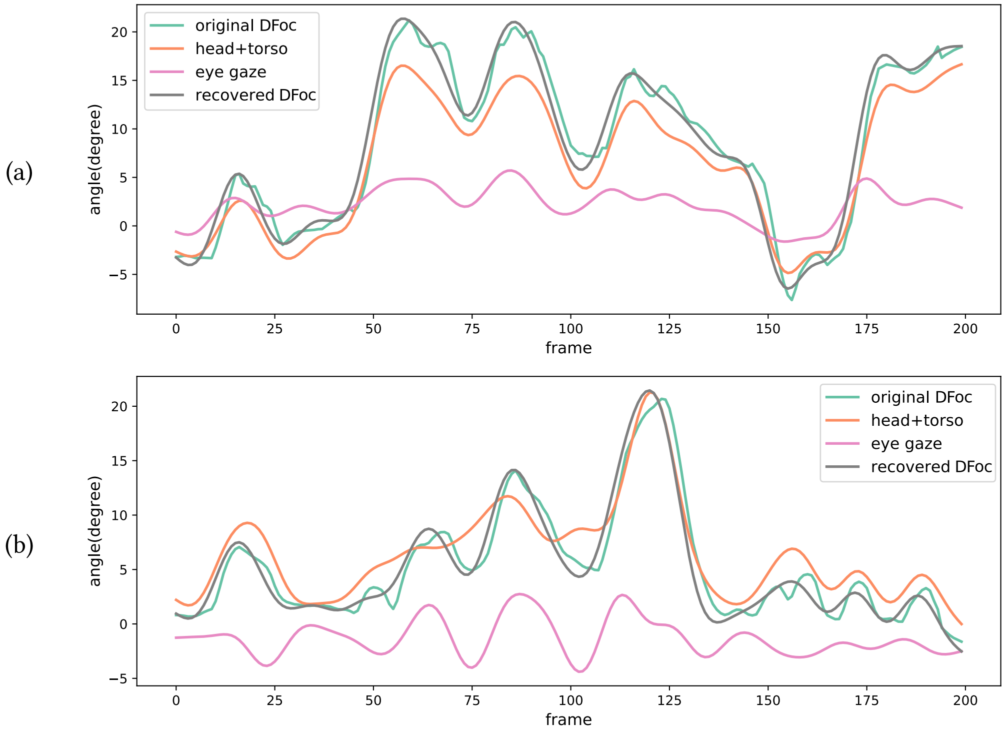


Fig. 8. Example results of the DFoc decomposition in our method. (a) The decomposed Yaw angles α by our method. (b) The decomposed Pitch angles β by our method.

$$(\alpha_e^k, \alpha_{h+t}^k) = \arg \min_{(\alpha_e^*, \alpha_{h+t}^*) \in B_i} ||(\alpha_e^*, \alpha_{h+t}^*)^T - (\alpha_e^{k-1}, \alpha_{h+t}^{k-1})^T||^2 / P^* \tag{3}$$

Here $(\alpha_e^*, \alpha_{h+t}^*)$ is any α sample in B_i, $(\alpha_e^{k-1}, \alpha_{h+t}^{k-1})$ is the known decomposition at the previous frame k-1, and P^* is the calculated probability value of $(\alpha_e^*, \alpha_{h+t}^*)$ according to the fitted GMM of B_i. The underlying rationale in Eq. 3 is to give a large penalty if the difference of the α samples between the current frame and the previous frame is large, or the calculated probability of the α sample is small. For the first frame, we simply choose the sample in B_i that has the largest probability since the information of the previous frame is unavailable.

We also evaluated the interaction snapshot refinement step. Given a sequence of DFocs for one interlocutor (called the original DFoc sequence), we decomposed the sequence into a sequence of eye gazes and a sequence of head+torso movements, as shown in Fig. 8. Then, we added them together to recover the DFoc sequence and compared the recovered DFoc sequence with the original DFoc sequence. As shown in this figure, our recovered DFoc sequence is reasonably close to but a little more smoother than the original one.

6 ADD OTHER ANIMATION DETAILS

To demonstrate the effectiveness of our approach, we also need to add other animation details of all the interlocutors including the lip-sync and hand/body movements of the speaker, and the hand/body movements of the two listeners. Below we describe how these animations are added onto our work.

Lip-sync of the speaker. We employ the Lipsync Tool (Annosoft Inc., 2018) to generate lip-sync based on speech input. This tool uses pre-defined 17 visemes to control mouth movements.

Hand/body gesture of the speaker. We directly implemented the gesture controllers by [Levine et al. 2010] to generate hand/body gesture of the speaker based on acoustic speech input. However, the difference between our implementation and the original work [Levine et al. 2010] is that we selectively focus on the motion synthesis of the upper body (i.e., hands, arms, and torso), while ignoring the head motion part.

Hand/Body gesture of listeners. We choose to extend the texture synthesis based eye motion synthesis approach [Deng et al. 2005] to generate natural hand/body movements on listeners. Specifically, from our acquired dataset, we select some frames as the hand/body motion examples in our synthesis process. Each texel in this case includes all the joint angles of the upper body except the head. Since in our work the lengths of to-be-synthesized listener gesture sequences vary significantly, instead of choosing a fixed patch size as in the work of [Deng et al. 2005] that is the critical parameter in the patch-based sampling algorithm, we employ an adaptive patch size, which is $\lfloor n/10 \rfloor + 1$. Here n denotes the length of the to-by-synthesized gesture sequence.

Head movement extraction. As described above, for each interlocutor (speaker or listener) in a three-party conversation, we can obtain his/her upper body motion. Meanwhile, we obtain the head-torso combined movement at current frame (§5.3). Therefore, we can straightforwardly obtain the head movement at current frame by subtracting the torso movement from the head-torso combined movement.

Test Audio	Audio Len. (frame)	Spe. DFoc (second)	Lis. DFoc (second)	Spe. Gestures (second)	Lis. Gestures (second)	Total (second)
No. 1	25	2.48	0.53	137.10	0.04	140.15
No. 2	73	2.99	1.04	406.76	0.08	410.87
No. 3	374	4.60	4.13	1955.27	0.26	1964.26

Table 1. Runtime statistics of our approach for several test cases, Spe. stands for Speaker and Lis. stands for Listener.

7 RESULTS AND EVALUATIONS

Validation of our divide-and-conquer methodology. In order to validate our divide-and-conquer methodology (i.e., synthesize the motion of head+eye and the motion of hand+torso separately), we calculated the correlation coefficient between the combined directions of head+gaze and the directions of the torso in the space spanned by the yaw and pitch angles. The range of the calculated correlation coefficient is between -1 and 1; 0 represents “no correlation” and +/- 1 represents “positive/negative linear correlation”. Figure 9 shows the visualized result of randomly selected, 10-minutes samples in our dataset. As shown in this figure, the calculated correlation coefficients between the directions of head+gaze and the directions of the torso are -0.178 and 0.018 for the yaw angle and the pitch angle, respectively. These coefficients show that the directions of head+gaze and the directions of the torso do not have statistically strong correlations. In this way, this helps to justify our divide-and-conquer strategy.

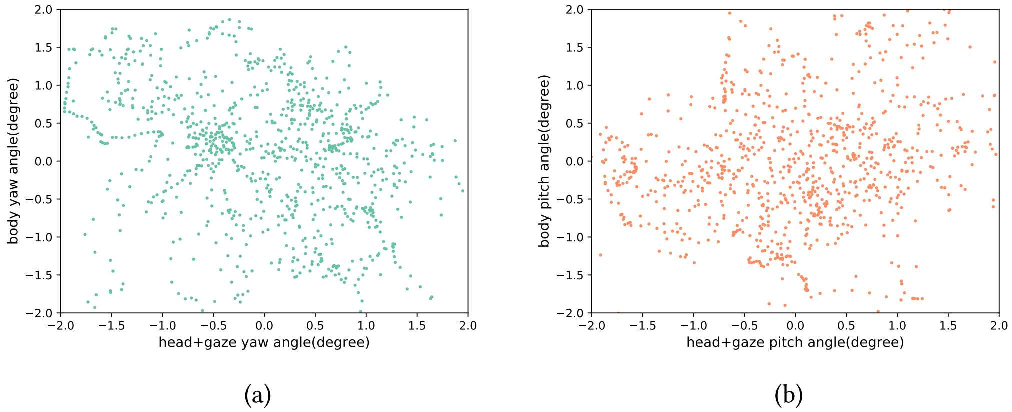


Fig. 9. Plotting of randomly selected, 10-min data samples (the combined directions of the head+gaze vs. the directions of the torso) in our dataset. (a) samples of the yaw angle α, and (b) samples of the pitch angle β.

Effectiveness of our method. To evaluate the effectiveness of our method, we applied it to many test speech clips for conversational animation generation. Note that some test conversational speech clips were recorded on new subjects uninvolved with our data acquisition process (§4). Before our method is called for animation generation, we need to manually mark who is the speaker in which period of the input speech. Figure 10 visualizes the resulting curves of head+torso and gaze angles (including the yaw angle and the pitch angle) of a synthesized three-party conversational animation example. In particular, its left panel shows the angle curves as well as the speaking time of each interlocutor, and its right panel shows the snapshots of the corresponding interlocutors at selected frames. Note that the bottom interlocutor at frame #110 and the top interlocutor at frame #600 in Figure 10 show the situations where a listener does not always look at the speaker (also refer to Figure 4). By contrast, in existing two-party listener models the listener is typically assumed to look at the speaker. For animation results, please refer to the supplemental demo video.

Runtime statistics. In all of our experiments, we ran our system on an off-the-shelf desktop computer with the following configuration: Intel i7-6700 CPU @3.4 GHz, 16GB Memory, and NVIDIA Geforce GTX 1070 GPU. The runtime statistics of the main steps of our system are shown in Table 1, where the speaker gesture generation module [Levine et al. 2010] consumes the majority (i.e., more than 99.7%) of the computational time in our current system. By contrast, the main contribution of this work (i.e., the DFoc synthesis for all the interlocutors) is much more efficient at runtime after the LSTM models are trained.

7.1 Perceptual User Studies

We further conducted perceptual user studies to evaluate the synthesized animation results by our approach. Since several different aspects (including head motion, eye motion, hand gesture/body motion, etc.) can potentially influence the visual perception of our synthesized animations, our user studies consists of two studies: i) a paired comparison study for head+eye motion, and ii) a paired comparison study for overall motion. Inspired by the demonstrated effectiveness of paired comparison studies for graphics/animation applications [Le et al. 2012; Ledda et al. 2005; Ma and Deng 2009], our two paired comparison studies are designed to perceptually compare the results by our method, the results by a selected baseline method (i.e., a state of the art method), and

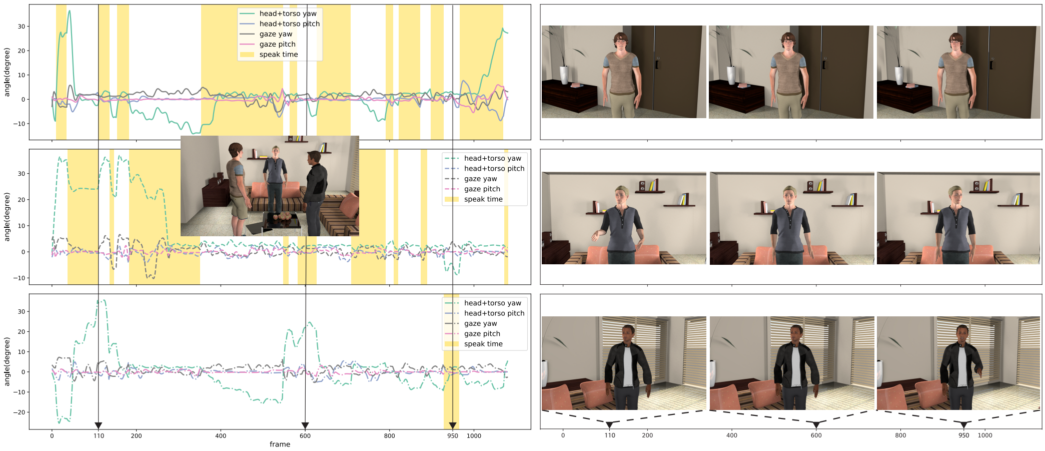


Fig. 10. The visualization of a synthesized three-party conversation example by our approach: the head+torso and gaze angles (including the yaw angle α and the pitch angle β) with their speak time. Each panel (from top to bottom) shows the information of one interlocutor. The snapshots in the right show the rendered corresponding interlocutor at three selected time frames.

the captured motion (ground-truth). We choose [Le et al. 2012] as the baseline method since it is the state of the art method to automatically generate coordinated head+eye motion on a virtual speaker based on acoustic speech input.

Specifically, in both studies, we randomly selected several three-party conversational speech segments from our retained test dataset, and then produced their corresponding three-party conversational animations using three different approaches, respectively: our approach, directly using the original pre-recorded motion (ground-truth), and the enhancement of [Le et al. 2012] (the baseline method). Since the original method by [Le et al. 2012] only considers the automated generation of the head+eye motion on the speaker, in our enhancement we added three modules used in our method: lip-sync, the generation of speaker hand+torso gestures (except head+eye motion), and the generation of listener gestures, onto the work of [Le et al. 2012] to ensure a fair comparison.

When producing animation clips with audio for our user studies, the same 3D scene, characters, and video resolution (1920×1080 pixels) were used. Participants were asked to sit in the same controlled environment, with a 0.6m distance away from a LCD monitor with 1920 × 1080 resolution. We did not impose maximal viewing time for each clip (in other words, participants can view the same clip many times before making a choice). Participants were asked to select the perceptually more realistic one between two side-by-side animation clips in a pair. They can optionally choose the undecided choice if they had difficulty to tell which one is more realistic between the two clips in a pair.

We recruited a total of 20 student volunteers in a university (16 males, 4 females; from 22 to 35 years old) to participate in our user studies. Note that both the order of displaying animation pairs for each participant and the left/right positions of the two clips in a pair were randomized.

i) Head+eye motion paired comparison study. We randomly selected 5 audio clips (each lasts about 15 seconds) from our retained test dataset, and then generated corresponding three-party conversational animations using our approach and the aforementioned enhancement of [Le et al. 2012], respectively. Also, we generated the corresponding ground-truth animation clips based on the pre-recorded motion data. The same pre-recorded mouth motion data was used to animate the mouth in all of the three comparison methods (ground-truth, our method, and the enhancement of [Le et al. 2012] - for convenience in the remaining writing we will just use [Le et al. 2012] to refer to its enhanced version). We produced a total of 15 animation clips with audio and formed 10 different pairs (our method vs. ground-truth, and our method vs. [Le et al. 2012]). Finally, as described previously, participants were asked to select the perceptually more realistic one between the two side-by-side clips in each pair.

Figure 11 shows the user voting result, where the symbol * denotes p-value < 0.05 for the comparison in the row through two-tailed independent paired t-test. As shown in this figure, our approach collected statistically significantly more votes than [Le et al. 2012] based on the speaker's head+eye motion (speaker based) and the overall head+eye motion of the three interlocutors (overall motion based). This is not surprising, since the model in [Le et al. 2012] is trained based on a pre-collected dataset of two-party conversational motions and it is not specifically designed for animating three-party conversations. On the other hand, when only considering listeners' head+eye motion (listeners based), our method obtained close votes as [Le et al. 2012], besides a large number of undecided votes (56). The main reason is that the same listener motion synthesis module is used in both our method and [Le et al. 2012], as mentioned above. Finally, compared with the original captured head+eye motion, our method collected close votes (all the p-values > 0.05). Note that our method even collected 4 more votes in the 'listeners based' and 'overall motion based' paired comparisons than the original captured motion, arguably due to the subjective nature of user studies.

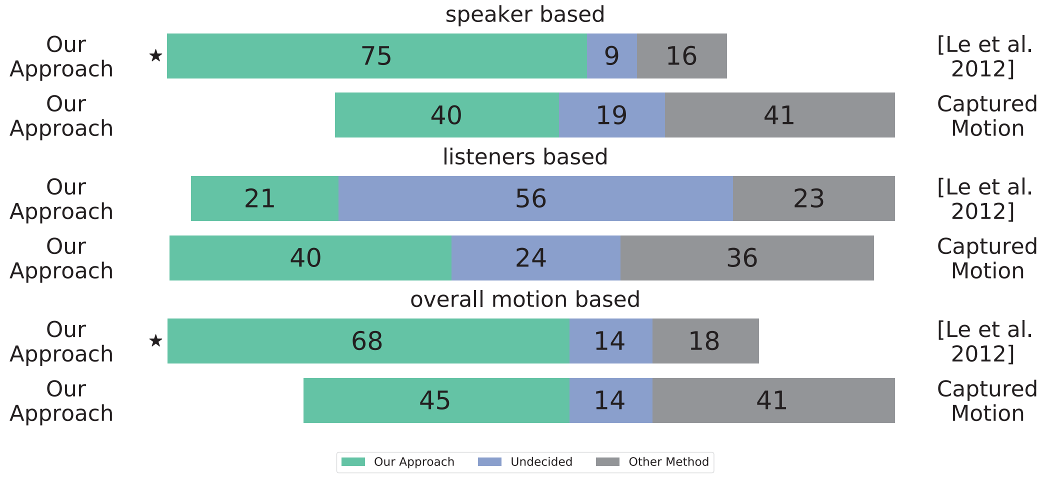


Fig. 11. The obtained vote counts in our head+eye motion paired comparison study based on the speaker's head+eye motion (speaker based), listeners' head+eye motion (listeners based), and the overall head+eye motion of the three interlocutors (overall motion based). Star indicates the computed statistical significance according to two-tailed independent paired t-test with p-value < 0.05.

ii) Overall motion paired comparison study. We randomly selected 5 audio clips (each lasts about 15 seconds) from our retained test dataset to generate corresponding conversational animations (including head motion, eye motion, hand gesture, body motion, etc.) by directly using the original captured motion, by our method, and by [Le et al. 2012], respectively. Just like the above study, we produced a total of 15 animation clips with audio and formed 10 different pairs (our method vs. ground-truth, and our method vs. [Le et al. 2012]), and participants were asked to select the perceptually more realistic one between the two side-by-side clips in each pair.

Figure 12 illustrates the obtained vote counts from this study. Our approach received statistically significant more votes than [Le et al. 2012], regardless the participants did the perceptual judgments based on the speaker's motion (speaker based), listeners' motion (listeners based), or the overall motion of all the three parties (overall motion based). One interesting observation is even though we generated the same motion for listeners in both our method and [Le et al. 2012], our approach received 14 more votes than [Le et al. 2012]. One plausible explanation is that the participants' overall visual perception on the animated conversations was, to a certain extent, affected by the speaker's motion, regardless the given study instructions.

On the other hand, compared with the original captured motion, our approach received 17 fewer votes based on the speaker's motion (speaker based) and 7 fewer votes based on the overall motion of the three interlocutors (overall motion based). Also, the calculated p-values for both the speaker based paired comparison and the overall motion based paired comparison > 0.05, which means the numbers of the votes received by our method and by the captured motion are not statistically significantly different.

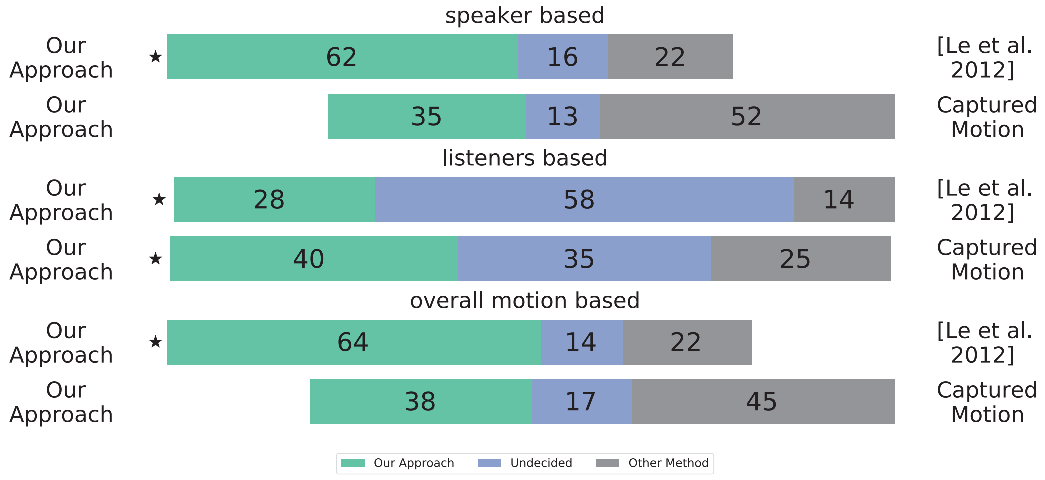


Fig. 12. The user vote counts in our overall motion paired comparison study, based on the speaker's motion (speaker based), listeners' motion (listeners based) and the overall motion of the three interlocutors (overall motion based). Star indicates the computed statistical significance according to a two-tailed independent paired t-test with p-value < 0.05.

8 DISCUSSION AND CONCLUSION

In this paper we present a deep learning based approach to synthesize realistic head and eye animations in three-party conversations. Its novel and core element is a LSTM-based machine learning model to predict the interaction snapshots (i.e., the DFocs of all the three interlocutors at any moment). Since our system is data-driven, we particularly acquired a high-quality, three-party conversational motion dataset for this work. Through many experiments as well as perceptual user studies, we found that our approach is capable of generating plausible head-and-eye motions for three-party conversational animations.

As the first work in a new direction, our current approach has several limitations. First, the computational efficiency of the whole system need to be further improved. Although, as aforementioned, the major computational bottleneck of our system is the generation of speaker gestures. In the future, we plan to design more efficient runtime algorithms (e.g., deep learning-based algorithms) to replace the current speaker gesture generation module. Second, our current approach is specifically focused on the automation, while ignoring the affective states of the interlocutors and the semantic aspects of the conversation content. Therefore, to produce more realistic three-party (or even multi-party) conversational animations, as the next step, we would like to explore the idea of extracting useful semantic features from the conversation content and the emotional features of the conversation, and then further exploit them for more realistic motion synthesis. Third, other physical arrangements of interlocutors (e.g., a semi-circle configuration) and detailed finger motions for conversations are not considered in the current work. Finally, since only one microphone was used to record acoustic speech during our data acquisition process, we cannot accurately extract the prosody for each speaker when overlapping speech occurs.

As the future work, we would like to explore how to extend or generalize the current three-party conversational animation approach for general multi-party conversational animation synthesis. In addition, the current approach can only be used for off-line applications. We are interested in developing highly efficient algorithms and pipelines to online generate multi-party conversational animations based on live speech input. We also plan to extract the emotional content from acoustic speech and/or facial expressions and exploit such informations to improve the current system.

ACKNOWLEDGMENTS

This work is in part supported by NSF IIS-1524782.

REFERENCES

Carlos Busso, Zhigang Deng, Michael Grimm, Ulrich Neumann, and Shrikanth Narayanan. 2007. Rigid head motion in expressive speech animation: Analysis and synthesis. *IEEE Transactions on Audio, Speech, and Language Processing* 15, 3 (2007), 1075–1086.

Carlos Busso, Zhigang Deng, Ulrich Neumann, and Shrikanth Narayanan. 2005. Natural head motion synthesis driven by acoustic prosodic features. *Computer Animation and Virtual Worlds* 16, 3-4 (2005), 283–290.

Justine Cassell, Catherine Pelachaud, Norman Badler, Mark Steedman, Brett Achorn, Tripp Becket, Brett Douville, Scott Prevost, and Matthew Stone. 1994. Animated conversation: rule-based generation of facial expression, gesture & spoken intonation for multiple conversational agents. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques.* ACM, 413–420.

Justine Cassell, Hannes Högni Vilhjálmsson, and Timothy Bickmore. 2001. Beat: the behavior expression animation toolkit. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques.* ACM, 477–486.

Erika Chuang and Christoph Bregler. 2005. Mood swings: expressive speech animation. *ACM Transactions on Graphics (TOG)* 24, 2 (2005), 331–347.

Zhigang Deng, John P Lewis, and Ulrich Neumann. 2005. Automated eye motion using texture synthesis. *IEEE Computer Graphics and Applications* 25, 2 (2005), 24–30.

Zhigang Deng and Junyong Noh. 2008. Computer facial animation: A survey. In *Data-driven 3D facial animation.* Springer, 1–28.

Yu Ding, Yuting Zhang, Meihua Xiao, and zhigang Deng. 2017. A Multifaceted Study on Eye Contact based Speaker Identification in Three-party Conversations. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems.* ACM, 3011–3021.

Mary Ellen Foster, Andre Gaschler, Manuel Giuliani, Amy Isard, Maria Pateraki, and Ronald Petrick. 2012. Two people walk into a bar: Dynamic multi-party social interaction with a robot agent. In *Proceedings of the 14th ACM international conference on Multimodal interaction.* ACM, 3–10.

Charles Goodwin and John Heritage. 1990. Conversation analysis. *Annual review of anthropology* 19, 1 (1990), 283–307.

Hans Peter Graf, Eric Cosatto, Volker Strom, and Fu Jie Huang. 2002. Visual prosody: Facial movements accompanying speech. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on.* IEEE, 396–401.

Jonathan Gratch, Anna Okhmatovskaia, Francois Lamothe, Stacy Marsella, Mathieu Morales, Rick J van der Werf, and Louis-Philippe Morency. 2006. Virtual rapport. In *IVA*, Vol. 6. Springer, 14–27.

Erdan Gu and Norman Badler. 2006. Visual attention and eye gaze during multiparty conversations with distractions. In *Intelligent Virtual Agents*. Springer, 193–204.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

Martin Johansson, Gabriel Skantze, and Joakim Gustafson. 2013. Head pose patterns in multiparty human-robot team-building interactions. In *International Conference on Social Robotics*. Springer, 351–360.

Sonu Chopra Khullar and Norman I Badler. 2001. Where to look? Automating attending behaviors of virtual human characters. *Autonomous Agents and Multi-Agent Systems* 4, 1-2 (2001), 9–23.

Yutaka Kondo, Kentaro Takemura, Jun Takamatsu, and Tsukasa Ogasawara. 2013. A gesture-centric android system for multi-party human-robot interaction. *Journal of Human-Robot Interaction* 2, 1 (2013), 133–151.

Binh H Le, Xiaohan Ma, and Zhigang Deng. 2012. Live speech driven head-and-eye motion generators. *IEEE transactions on visualization and computer graphics* 18, 11 (2012), 1902–1914.

Patrick Ledda, Alan Chalmers, Tom Troscianko, and Helge Seetzen. 2005. Evaluation of tone mapping operators using a high dynamic range display. In *ACM Transactions on Graphics (TOG)*, Vol. 24. ACM, 640–648.

Jina Lee and Stacy Marsella. 2009. Learning a model of speaker head nods using gesture corpora. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 289–296.

Sooha Park Lee, Jeremy B Badler, and Norman I Badler. 2002. Eyes alive. In *ACM Transactions on Graphics (TOG)*, Vol. 21. ACM, 637–644.

Sergey Levine, Philipp Krähenbühl, Sebastian Thrun, and Vladlen Koltun. 2010. Gesture controllers. In *ACM Transactions on Graphics (TOG)*, Vol. 29. ACM, 124.

Sergey Levine, Christian Theobalt, and Vladlen Koltun. 2009. Real-time prosody-driven synthesis of body language. In *ACM Transactions on Graphics (TOG)*, Vol. 28. ACM, 172.

John P Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Frédéric H Pighin, and Zhigang Deng. 2014. Practice and Theory of Blendshape Facial Models. *Eurographics (State of the Art Reports)* 1, 8 (2014).

Xiaohan Ma and Zhigang Deng. 2009. Natural eye motion synthesis by modeling gaze-head coupling. In *Virtual Reality Conference, 2009. VR 2009. IEEE*. IEEE, 143–150.

RM Maatman, Jonathan Gratch, and Stacy Marsella. 2005. Natural behavior of a listening agent. In *International Workshop on Intelligent Virtual Agents*. Springer, 25–36.

Stacy Marsella, Yuyu Xu, Margaux Lhommet, Andrew Feng, Stefan Scherer, and Ari Shapiro. 2013. Virtual character performance from speech. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 25–35.

Soh Masuko and Junichi Hoshino. 2006. Generating head–eye movement for virtual actor. *Systems and Computers in Japan* 37, 12 (2006), 33–44.

Yoichi Matsuyama, Hikaru Taniyama, Shinya Fujie, and Tetsunori Kobayashi. 2010. Framework of Communication Activation Robot Participating in Multiparty Conversation.. In *AAAI Fall Symposium: Dialog with Robots*.

Kevin G Munhall, Jeffery A Jones, Daniel E Callan, Takaaki Kuratate, and Eric Vatikiotis-Bateson. 2004. Visual prosody and speech intelligibility: Head movement improves auditory speech perception. *Psychological science* 15, 2 (2004), 133–137.

Bilge Mutlu, Toshiyuki Shiwa, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. 2009. Footing in human-robot conversations: how robots might shape participant roles using gaze cues. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*. ACM, 61–68.

Kazuhiro Otsuka, Hiroshi Sawada, and Junji Yamato. 2007. Automatic inference of cross-modal nonverbal interactions in multiparty conversations: who responds to whom, when, and how? from gaze, head gestures, and utterances. In *Proceedings of the 9th international conference on Multimodal interfaces*. ACM, 255–262.

Kazuhiro Otsuka, Yoshinao Takemae, and Junji Yamato. 2005. A Probabilistic Inference of Multiparty-conversation Structure Based on Markov-switching Models of Gaze Patterns, Head Directions, and Utterances. In *Proceedings of the 7th International Conference on Multimodal Interfaces (ICMI '05)*. ACM, New York, NY, USA, 191–198. https://doi.org/10.1145/1088463.1088497

Tomislav Pejsa, Daniel Rakita, Bilge Mutlu, and Michael Gleicher. 2016. Authoring directed gaze for full-body motion capture. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 161.

Christopher Peters and Carol O'Sullivan. 2003. Attention-driven eye gaze and blinking for virtual humans. In *ACM SIGGRAPH 2003 Sketches & Applications*. ACM, 1–1.

Kerstin Ruhland, Sean Andrist, Jeremy Badler, Christopher Peters, Norman Badler, Michael Gleicher, Bilge Mutlu, and Rachel Mcdonnell. 2014. Look me in the eyes: A survey of eye and gaze animation for virtual agents and artificial systems. In *Eurographics State-of-the-Art Report*. 69–91.

Mehmet E Sargin, Yucel Yemez, Engin Erzin, and Ahmet M Tekalp. 2008. Analysis of head gesture and prosody patterns for prosody-driven head-gesture animation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 8 (2008), 1330–1345.

William Steptoe, Oyewole Oyekoya, and Anthony Steed. 2010. Eyelid kinematics for virtual characters. *Computer animation and virtual worlds* 21, 3-4 (2010), 161–171.

Matthew Stone, Doug DeCarlo, Insuk Oh, Christian Rodriguez, Adrian Stere, Alyssa Lees, and Chris Bregler. 2004. Speaking with hands: Creating animated conversational characters from recordings of human performance. *ACM Transactions on Graphics (TOG)* 23, 3 (2004), 506–513.

Marcus Thiebaux, Brent Lance, and Stacy Marsella. 2009. Real-time expressive gaze animation for virtual humans. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 321–328.

Roel Vertegaal, Robert Slagter, Gerrit Van der Veer, and Anton Nijholt. 2001. Eye gaze patterns in conversations: there is more to conversational agents than meets the eyes. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 301–308.

Roel Vertegaal, Gerrit van der Veer, and Harro Vons. 2000. Effects of gaze on multiparty mediated communication. In *Graphics Interface*. 95–102.

Vinoba Vinayagamoorthy, Maia Garau, Anthony Steed, and Mel Slater. 2004. An eye gaze model for dyadic interaction in an immersive virtual environment: Practice and experience. In *Computer Graphics Forum*, Vol. 23. Wiley Online Library, 1–11.

Congyi Wang, Fuhao Shi, Shihong Xia, and Jinxiang Chai. 2016. Realtime 3d eye gaze animation using a single rgb camera. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 118.

Paul Watzlawick, Janet Beavin Bavelas, and Don D Jackson. 2011. *Pragmatics of human communication: A study of interactional patterns, pathologies and paradoxes*. WW Norton & Company.