Audio-based Head Motion Synthesis for Avatar-based Telepresence Systems

Zhigang Deng, Carlos Busso*, Shri Narayanan*, and Ulrich Neumann
Computer Graphics and Immersive Technology (CGIT) Lab, Dept. of Computer Science
Integrated Media System Center, *Dept. of EE-Systems
University of Southern California, Los Angeles
http://graphics.usc.edu

ABSTRACT

In this paper, a data-driven audio-based head motion synthesis technique is presented for avatar-based telepresence systems. First, head motion of a human subject speaking a custom corpus is captured, and the accompanying audio features are extracted. Based on the aligned pairs between audio features and head motion (audio-headmotion), a K-Nearest Neighbors (KNN)-based dynamic programming algorithm is used to synthesize novel head motion given new audio input. This approach also provides optional intuitive keyframe (key head poses) control: after key head poses are specified, this method will synthesize appropriate head motion sequences that maximally meet the requirements of both the speech and key head poses.

Categories and Subject Descriptors

I.3.7 [**Computer Graphics**]: Three Dimensional Graphics and Realism-Animation, Virtual Reality; I.2.6 [**Artificial Intelligence**]: Learning-Knowledge acquisition; H.5.1 [**Multimedia Information Systems**]: Animations

General Terms

Algorithms, Design, Experimentation, Human Factors

Keywords

Computer Graphics, Facial Animation, Data-driven, Head Motion Synthesis, K Nearest Neighbor, Audio-based, Keyframing Control, Telepresence Systems

1. INTRODUCTION

Humans communicate via two channels [1, 2]: an explicit channel (speech) and an implicit channel (non-verbal gestures). In computer graphics community, significant effort has been made to model the explicit speech channel [3, 4, 5, 6, 7, 8, 9]. However, speech production is often accompanied by non-verbal gestures, such as head motion and eye blinking. The perception study [2] reports that head motion plays a direct role in the perception of speech based on the evaluation of a speech-in-noise task by Japanese subjects. Also, adding head motion can greatly enhance the realism of synthesized facial animation that is being increasingly used in many fields, e.g. education, communication, and entertainment industry.

Because of the complexity of the association between the speech channel and its accompanying head motion, generating appropriate head motion for new audio is a time-consuming and tedious job for animators. They often manually make key head poses by referring to the recorded video of actors reciting the audio/text or capturing the head motion of real actors using motion capture systems. However, it is impossible to reuse the captured data/video for other scenarios without considerable effort. Furthermore, making appropriate head motion for the conversation of multiple humans (avatars) poses more challenging problems for manual approaches and motion capture methods. And automatic head motion is a requirement in many applications, such as autonomous avatars in avatar-based telepresence systems, interactive characters in computer games, etc.

2. PREVIOUS AND RELATED WORK

A comprehensive review on facial animation is beyond this work, and a recent review can be found in [10]. Recent relevant research on non-verbal gestures and head motion is described in this section.

2.1 Non-Verbal Gestures

Pelachaud et al. [11] generate facial expressions and head movements from labeled text using a set of custom rules, based on Facial Action Coding System (FACS) representations [12]. Cassell et al. [13] present an automated system that generates appropriate non-verbal gestures, including head motion, for conversations among multiple avatars, but they address only the "nod" head motion in their work. Perlin and Goldberg [14] develop an "Improv" system that combines procedural and rule-based techniques for behavior-based characters. The character actions are predefined, and decision rules are used to choose the appropriate combinations and transitions. Kurlander et al. [15] construct a "comic chat" system that automatically generates 2D comics for online graphical chatting, based on the rules of comic panel composition. Chi et al. [16] present an EMOTE model by implanting Laban Movement Analysis (LMA) and its efforts and shape components into character animation. It is successfully applied to arm and body movements, but the

ETP'04, October 15, 2004, New York, New York, USA.

applicability of this method to head movements with speech has not been established. Cassell et al. [17] generate appropriate non-verbal gestures from text, relying on a set of linguistic rules derived from non-verbal conversational research. This method works on text, but the possibility of applying this method to audio input is not verified yet.

2.2 Head Motion

Researchers have reported that there are strong correlations between speech and its accompanying head movements [18, 19, 20, 21, 2]. For example, Munhall et al. [2] show that the rhythmic head motion strongly correlates with the pitch and amplitude of the subject's voice. Graf et al. [21] estimate the probability distribution of major head movements (e.g. "nod") according to the occurrences of pitch accents. [19, 18] even report that about 80% of the variance observed for fundamental frequency (F0) can be determined from head motion, although the average percentage of head motion variance that can be linearly inferred from F0 is much lower. Costa et al. [20] use the Gaussian Mixture Model (GMM) to model the association between audio features and visual prosody. In their work, only eyebrow movements are analyzed, and the connection between audio features and head motion is not reported. As such, a data-driven synthesis approach for head motion has not been published yet.

In this paper, a data-driven technique is presented to automatically synthesize "appropriate head motion" for given novel speech input. First, head motion is extracted from the captured data of a human subject, speaking a custom corpus with different expressions. Audio is captured simultaneously. All the audio-headmotion pairs are collected into a database indexed by audio features. Next, the audio features of given new speech (audio) input are used to search for their K nearest neighbors in this database. All such K chosen nearest neighbors are put into a nearest neighbor candidate pool, and a dynamic programming algorithm is used to find the optimum nearest neighbor combination by minimizing a cost function. This approach also provides flexible control for animators. If key head poses are specified, this approach will try to maximally satisfy the requests from speech and key head poses.

The main contributions of this approach are:

- It is fully automatic. It can synthesize appropriate head motion directly from audio, and it can be used for many applications, e.g. avatar-based telepresence systems and computer games.
- It also provides optional flexible control. By specifying key head poses, animators can influence the synthesized head motion. Another control is to the ability to adjust "searching weights" (Section 5) to meet the synthesis preference of the animators.

Section 3 describes the acquisition and preprocessing of both audio and visual data. Section 4 describes how to synthesize novel head motion given new speech using a KNN-based dynamic programming technique, and how to search for the optimized weights is discussed (Section 5). Finally, results and conclusions are described Section 6&7 respectively.

3. DATA ACQUISITION

An actress with markers on her face was captured with a Vicon motion capture system [22]. The actress was directed to speak a custom designed corpus composed of about two hundred sentences, each with four expressions (neutral, happy, angry and sad) as naturally as possible. At the same time, the accompanying audio was recorded. This data were captured for a comprehensive research project, and only head motion data is used in this work. Figure 1 illustrates the markers used in this process.

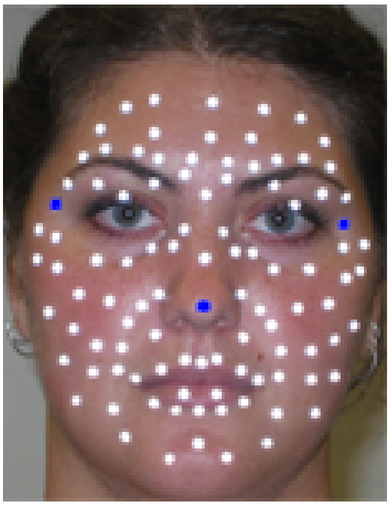

Figure 1: Illustration of the markers used in motion capture. Dark points are three chosen approximately rigid points.

The following procedure is used to extract the transformation of head motion for each frame:

1. A specific nose point is assumed to be the local coordinate center of each frame, and one frame in a neutral pose is chosen as a reference frame.
2. A local coordinate system is defined by three chosen approximately rigid points (the nose point and corner points of the eyes, shown as dark points in Fig 1). The distance between the nose point in each frame and that of the reference frame is its translation vector, and aligning each frame with the reference frame generates its rotation matrix.
3. Since this transformation is only composed of rotation and translation, it can be further decomposed into a six dimensional transformation vector [23]: three Euler angles (converted to "Radians" in this work) and three translational values. As such, a six dimensional transformation vector (T-VEC) is generated.
4. The difference between the T-VECs of two consecutive frames (suppose t_i and t_{i+1}) is the head motion at t_i.

The acoustic information is extracted using the Praat speech processing software [24] with a 30-ms window and 21.6-ms of overlap. The audio-features used are the pitch (F0), the lowest five formants (F1 through F5), 13-MFCC (Mel-Frequency Cepstral Coefficients) and 12-LPC (Linear Prediction Coefficient). These 31 dimensional audio feature vectors are reduced to four dimensions using Principal

Component Analysis (PCA), covering 98.89% of the variation. An audio feature PCA space expanded by four eigenvectors (corresponding to the four largest eigen-values) is also constructed. Note that which audio features enclose most useful information for head motion estimation is still an open question, and audio features used for this work are chosen experimentally.

In this way, a database of aligned audio-head-motion is constructed (Fig 2). For simplicity, *AHD* (Audio-Headmotion Database) is used to refer to this database in the remaining sections. Each entry of the AHD is composed of a four dimensional audio feature PCA coefficients (AF-COEF) and a head motion transformation vector (T-VEC). Note that the AHD is indexed by the AF-COEF.

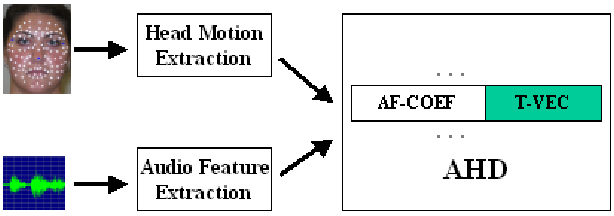

Figure 2: Illustration of the Audio-Headmotion Database (AHD). Each entry in this database is composed of two parts: a AF-COEF (four dimensional audio feature pca coefficients) and a T-VEC (six dimensional head motion transformation vector).

4. SYNTHESIZING HEAD MOTION

After the AHD is constructed (Section 3), the audio features of a given novel speech input are reduced into AF-COEFs by projecting them into the audio feature PCA space (Eq. 1) created in Section 3. Here F is a 31 dimensional audio feature vector, f is its AF-COEF, and M is the eigenvector matrix (31*4 in this case).

$$f = M^T.(F - \bar{F}) \tag{1}$$

Then, these AF-COEFs are used to index the AHD and search for their K nearest neighbors. After these neighbors are identified, a dynamic programming technique is used to find the optimum nearest neighbor combination by minimizing the total cost. Finally, the head motion of the chosen nearest neighbors is concatenated together to form the final head motion sequence. Figure 3 illustrates this head motion synthesis pipeline.

4.1 Find K-Nearest Neighbors

Given an input (inquiry) AF-COEF q, its nearest K neighbors in the AHD are located. In this case, K (the number of nearest neighbors) is experimentally set to 7 (Section 5). The euclidean distance is used to measure the difference between two AF-COEFs (Eq. 2). Here d represents a AF-COEF of an entry in the AHD. In this step, this distance (termed *neighbor-distance* in this paper) is also retained.

$$dist = \sqrt{\sum_{i=1}^{4}(q_i - d_i)^2} \tag{2}$$

Numerous approaches were presented to speed up the K-nearest neighbor search, and a good overview can be found in [25]. In this work, KD-tree [26] is used to speed up this search. The average time complexity of a KD-tree search is $O(\log N_d)$, where N_d is the size of the dataset.

4.2 Dynamic Programming Optimization

After the PCA projection and K nearest neighbors search, for a AF-COEF f_i at time T_i, its K nearest neighbors are found (assume its K nearest neighbors are $N_{i,1}, N_{i,2}, \ldots, N_{i,K}$). Which neighbor should be optimally chosen at time T_i? A dynamic programming technique is used here to find the optimum neighbor combination by minimizing the total "*synthesis cost*" ("synthesis error" and "synthesis cost" are used interchangably in this paper).

The synthesis cost (error) at time T_i is defined to include the following three parts:

- *Neighbor-distance Error (NE)*: the neighbor-distance (Eq. 2) between the AF-COEF of a nearest neighbor, e.g. $c_{i,j}$, and the input AF-COEF f_i (Eq. 3).

$$NE_{i,j} = \|c_{i,j} - f_i\|_2 \tag{3}$$

- *Roughness Error (RE)*: represents the roughness of the synthesized head motion path. Smooth head motion (small RE) is preferred. Suppose V_{i-1} is the T-VEC at time T_{i-1} and $TV_{i,j}$ is the T-VEC of j^{th} nearest neighbor at time T_i. When the j^{th} neighbor is chosen at time T_i, $RE_{i,j}$ is defined as the second derivative at time T_i as follows (Eq. 4):

$$RE_{i,j} = \|TV_{i,j} - V_{i-1}\|_2 \tag{4}$$

- *Away Keyframe Error (AE)*: represents how far away the current head pose is from specified key head pose. Head motion toward specified key head poses decreases the AE. Suppose KP is the next goal of key head pose at time T_i and P_{i-1} is the head pose at time T_{i-1}, then $AE_{i,j}$ is calculated (Eq. 5).

$$AE_{i,j} = \|KP - (P_{i-1} + TV_{i,j})\|_2 \tag{5}$$

If the j^{th} neighbor is chosen at time T_i and W_n, W_r, and W_a (assume $W_n \geq 0, W_r \geq 0$, $W_a \geq 0$, and $W_n + W_r + W_a = 1$) are the weights for NE, RE and AE respectively, the synthesis error $err_{i,j}$ (when the j^{th} nearest neighbor is chosen at time T_i) is the weighted sum ofthe above three errors (Eq. 6).

$$err_{i,j} = W_n.NE_{i,j} + W_r.RE_{i,j} + W_a.AE_{i,j} \tag{6}$$

Since the decision made at time T_i only depends on the current K neighbor candidates and the previous state (e.g. the head pose) at time T_{i-1}, a dynamic programming technique is used to solve the optimum nearest neighbor combination.

Suppose $ERR_{i,j}$ represents the accumulated synthesis error from time T_1 to T_i when j^{th} neighbor is chosen at time T_i; $PATH_{i,j}$ represents the chosen neighbor at time T_{i-1} when the j^{th} neighbor is chosen at time T_i. Further assume that all the $NE_{i,j}$, $RE_{i,j}$, $AE_{i,j}$, $ERR_{i,j}$, and $PATH_{i,j}$ are available for $1 \leq i \leq l-1$ and $1 \leq j \leq K$, we move forward to time T_l using the following equations (Eq. 7-9).

$$err_{l,j}^{m} = (ERR_{l-1,m} - W_a.AE_{l-1,m}) + W_r.RE_{l,j} + W_a.AE_{l,j} \tag{7}$$

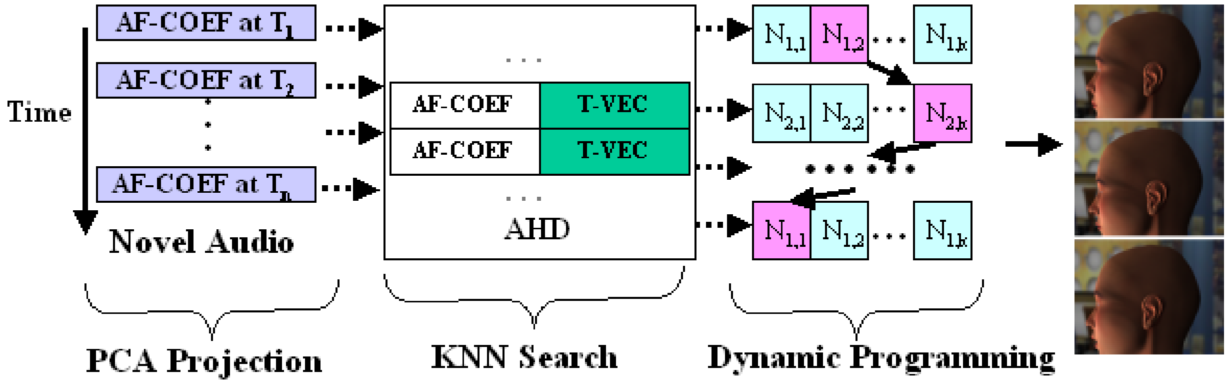

Figure 3: Illustration of the head motion synthesis pipeline. The first step is to project audio features onto the audio feature PCA space, the second step is to find K nearest neighbors in the AHD, and the third step is to solve the optimum combination by dynamic programming.

$$ERR_{l,j} = \min_{m=1...K}(err^{m}_{l,j}) + W_n.NE_{l,j} \tag{8}$$

$$PATH_{l,j} = \arg\min_{m=1...K}(err^{m}_{l,j} + W_n.NE_{l,j}) \tag{9}$$

Note that in Eq. 7, $1 \leq m \leq K$ and $(ERR_{l-1,m} - AE_{l-1,m})$ is used to remove the old AE, because only new AE is useful for current search. $PATH_{l,j}$ retains retracing information about which neighbor is chosen at time T_{l-1} if j^{th} nearest neighbor is chosen at time T_i.

Finally, the optimum nearest neighbor combination is determined by Equation 10-11. Assume s_i represents the nearest neighbor optimally chosen at time T_i.

$$s_n = \arg\min_{j=1...K} ERR_{n,j} \tag{10}$$

$$s_{i-1} = PATH_{i,s_i} \qquad 2 \leq i \leq n \tag{11}$$

Suppose $TV_{i,j}$ is the T-VEC of j^{th} nearest neighbor at time T_i, the final head pose $HeadPos_i$ at time T_i $(1 \leq i \leq n)$ is calculated in Eq. 12.

$$HeadPos_i = \sum_{j=1}^{i} TV_{j,S_j} \tag{12}$$

The time complexity of this KNN-based dynamic programming synthesis algorithm is $O(n.\log N_d + n.K^2)$, where K is the number of nearest neighbors, N_d is the number of entries in the AHD, and n is the number of input AF-COEF, for example, if 30 head motion frames per second is synthesized and t is the total animation time (second), then $n = t * 30$.

5. CHOOSING THE OPTIMUM WEIGHTS

As described in Section 4, the dynamic programming synthesis algorithm uses three weights $\vec{W}(W_n,\ W_a,\ W_r)$ to influence the outcome of the chosen nearest neighbors. What are the optimum weights for this head motion synthesis algorithm? Since it is assumed that $W_a \geq 0$,$W_n \geq 0$, $W_r \geq 0$, and $W_a + W_n + W_r = 1$. The searching space can be illustrated as Fig. 4.

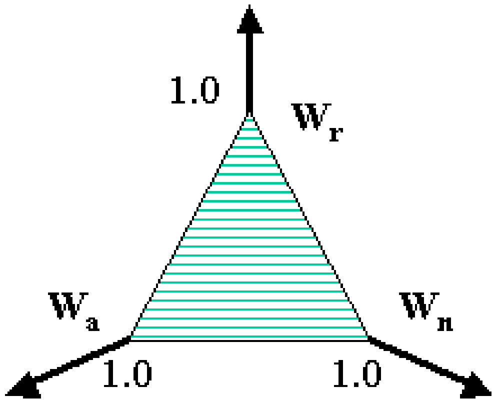

Figure 4: Illustration of the search space of the weights $\vec{W}(W_a,\ W_n,\ W_r)$.

Several speech segments (from the captured data, not those used for constructing the AHD in Section 3) are used for cross-validation [27]. For each speech segment, the key head poses at the start time and the ending time are specified as the same as the original captured head poses. For a specific weight configuration, *Total Evaluation Error* (TEE) is defined as follows (Eq. 13):

$$TEE(W_n, W_a, Wr) = \sum_{i=1}^{N}\sum_{j=1}^{6}(\hat{V}_i^j - V_i^j)^2 \tag{13}$$

Where N is the number of total cross-validation head motion frames, $\hat{V}_i$ is the synthesized head pose at frame i, and V_i is the ground-truth head pose at frame i.

A variant of gradient-descent method and non-sequential random search [28] are combined to search the global minimum TEE (its weights are the optimum weights) (Eq. 14-15). Here only four basic directions are considered: $\vec{e}_1 = (\alpha, 0, -\alpha)$, $\vec{e}_2 = (-\alpha, 0, \alpha)$, $\vec{e}_3 = (0, \alpha, -\alpha)$, and $\vec{e}_4 = (0, -\alpha, \alpha)$. α is the step size (experimentally set to 0.05 in this work)

$$j = \arg\min_{i=1..4} TEE(\vec{W}_t + \vec{e}_i) \tag{14}$$

$$\vec{W}_{t+1} = \vec{W}_t + \vec{e}_j \tag{15}$$

The initial weight $\vec{W}_0$ is generated as follows: W_a is randomly sampled from the uniform distribution [0..1], then W_n is randomly sampled from uniform distribution [0...1-W_a], and W_r is assigned $1 - W_a - W_n$.

Non-sequential random Search [28] is used to avoid getting stuck at a local minimum in the weight space: a given number of initial weights are generated at random, then each initial weight performs an independent search, and finally, the winner among all the searches is the optimum weights. Fig 5 illustrates the search result after 20 initial weights are used. The resultant optimum weights $\vec{W}$= $[W_a = 0.31755, W_n = 0.15782, W_r = 0.52463]$.

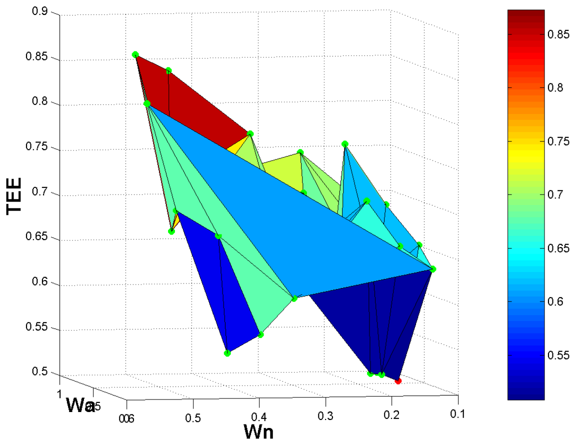

Figure 5: Plot of the search result after 20 initial weights are used (K=7). The global minimum is the red point, corresponding to the weights: W_a=0.31755, W_n=0.15782, and W_r= 0.52463.

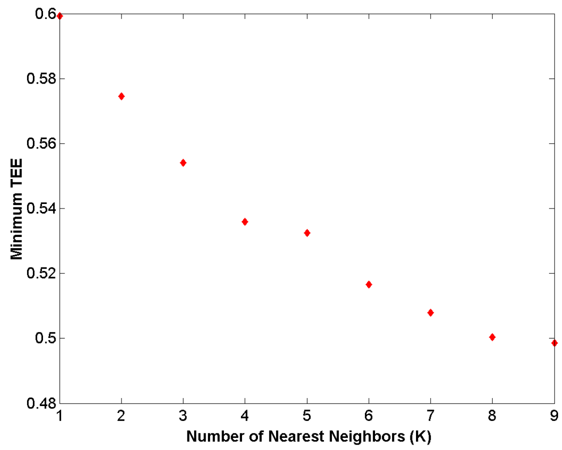

Figure 6: Plot of minimum TTE versus K. For each K, 20 iterations of non-sequential random search are used.

We argue that the optimum weights may depend on the subject, since the audio-headmotion mapping reflected in the constructed AHD may capture the head motion personality of the captured subject. Further investigation is needed to compare the optimum weights of different subjects.

Since the number of nearest neighbors is discrete, unlike the continuous weight space, we experimentally set the optimized K to 7 using the following experiments: after K is set to a fixed number, the above searching method was used to search the minimum TEE. Figure 6 illustrates the minimum TTE with respect to different K.

6. RESULTS AND APPLICATIONS

6.1 Ground-Truth Comparison

To evaluate this approach, ground-truth head motion is compared to the synthesized head motion. A speech segment that was not used for training and cross-validation is used for comparisons, and approriate key head poses are also specified (only start head pose and ending head pose). Figure 7 illustrates the trajectory comparisons of synthesized head motion and ground-truth one.

6.2 Applications without Keyframes

In many applications, such as avatar-based telepresence systems and computer games, automated head motion is required. This approach can be applied to these applications by simply setting W_a to zero. Therefore, the head motion is guided only by the roughness and neighbor-distance criterias. In some cases, staying in the initial head pose is preferred, for example, the avatar speaking and paying attention only to one specific human subject, e.g. the user. By automatically setting key head poses to the initial head pose, the system can simulate these scenarios. Figure 8 illustrates some frames of synthesized head motion.

6.3 Applications with Keyframes

Although various automatic approaches were presented, keyframing is still a useful tool for animators. For example, in the case of the conversation of multiple avatars, head motion often accompanies turn-takings. Therefore, animators can specify the appropriate key head poses, corresponding to the turn-taking time. This approach will automatically fill in the head motion gaps. If animators want the synthesized head motion to more closely follow key head poses, animators just need to increase the weight W_a.

7. CONCLUSIONS AND FUTURE WORK

In this paper, a data-driven audio-based approach is presented for automatically synthesizing appropriate head motion for avatar-based telepresence systems. The audio-headmotion mapping is stored in a database (AHD), constructed from the captured head motion data of a human subject. Given novel speech (audio) input and optional key head poses, a KNN-based dynamic programming technique is used to find the optimized head motion from the AHD, maximally satisfying the requirements from both audio and specified key head poses. Keyframe control provides flexibility for animators without the loss of the naturalness of synthesized head motion.

This approach can be applied to many scenarios where automated head motion is required, such as automated head motion and conversations among multiple avatars. Flexibly tuning the weights used in the algorithm and specifying

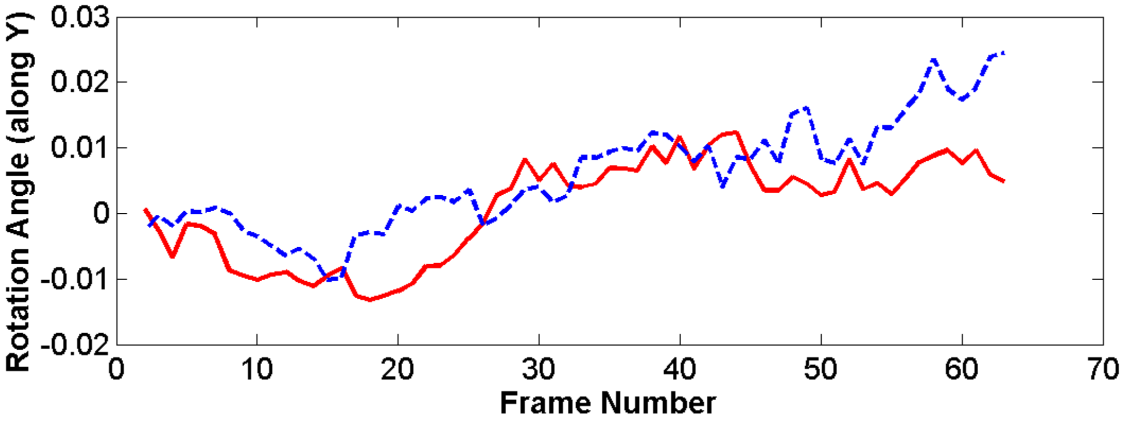

Figure 7: Comparison of ground-truth head motion (red solid curve) and the synthesized head motion (dashed blue curve), for she neutrally pronounce utterance "Do you have an aversion to that"? Note that the motion tendency at most places is similar.

appropriate key head poses will generate various styles of synthesized head motion. It also can be used as a fast tool for making initial head motion. Comparing with making animation from scratch, refining the generated initial head motion saves much time for animators.

A limitation of this data-driven approach is that it is difficult to anticipate in advance the amount of training data needed for specific applications. For example, if the specified key head poses are beyond the training data, the performance of this approach will degrade, since there are not enough "matched" head motion entries in the AHD to achieve the specified key head poses. But after some animation is generated, it is easy to evaluate the variety and appropriateness of synthesized head motion and obtain more data if necessary. Designing a database to achieve greater degree application independence is a topic for open research.

We are aware that head motion is not an independent part of the whole facial motion. Since it may strongly correlate with eye motion, e.g. head motion-compensated gaze, appropriate eye motion will greatly enhance the realism of synthesized head motion. The linguistic structure of the speech also plays an important role in the head motion of human subjects [11]. We plan to combine the linguistic structure into this approach: a combination of linguistic (e.g. syntactic and discourse) and audio features will be used to drive the head motion.

We also plan to investigate the possibility of combining this approach with human body animation, as in the case of a human speaking while walking/running, since the head motion composition involved may not just be a simple addition.

8. ACKNOWLEDGMENTS

This research has been funded by the Integrated Media System Center/USC, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152. Special Thanks go to Murtaza Bulut and J.P.Lewis for data capture and insightful discussions, Hiroki Itokazu and Bret St. Clair for model preparation, Pamela Fox for proof reading. We also appreciate many valuable comments from other colleagues in the CGIT Lab/USC.

9. REFERENCES

[1] Banse, R. and Scherer, K. Acoustic profiles in vocal emotion expression. *Journal of Personality and Social Psycology*, 70, 3, 1996, 614-636.

[2] Munhall,K. G., Jones, J. A., Callan, D. E., Kuratate,T., and Bateson, E.V. Visual Prosody and speech intelligibility: Head movement improves auditory speech perception. *Psychological Science*, 15, 2 (Feb 2004), 133-137.

[3] Bregler, C., Covell, M., and Slaney, M. Video Rewrite: Driving Visual Speech with Audio, In *Proceedings of ACM SIGGRAPH'97*, 1997, 353-360.

[4] Brand, M. Voice Puppetry, In *Proceedings of ACM SIGGRAPH'99*, 1999, Los Angeles, 21-28.

[5] Noh, J. Y., and Neumann, U. Expression Cloning, In *Proceedings of ACM SIGGRAPH'01*, 2001, Los Angeles, 277-288.

[6] Ezzat, T., Geiger, G., and Poggio, T. Trainable Videorealistic Speech Animation. *ACM Trans. On Graphics (Proc. of ACM SIGGRAPH'02)*, 21, 3, 2002, 388-398.

[7] Kshirsagar, S., and Thalmann, N. M. Visyllable based Speech Animation. *Computer Graphics Forum (Proc. of Eurographics'03)*, 22, 3, 2003, 631-640.

[8] Blanz, V., Busso, C., Poggio, T., and Vetter, T. Reanimating Faces in Images and Video. *Computer Graphics Forum (Proc. of Eurographics'03)*, 22, 3, 2003, 641-650.

[9] Deng, Z., Bulut, M., Neumann, U., Narayanan, S. Automatic Dynamic Expression Synthesis for Speech Animation. In *Proceedings of IEEE 17th International Conference on Computer Animation and Social Agents (CASA) 2004*, Geneva, Switzerland, July 2004, 267-274.

[10] Parke, F. I., and Waters, K. *Computer Facial Animation.* A K Peters, Wellesey, Massachusetts, 1996.

Figure 8: Some frames of synthesized head motion sequence, driven by the recorded speech "By day and night he wrongs me; every hour He flashes into one gross crime or other..." from a Shakespere's play.

[11] Pelachaud, N., and Badler, N., and Steedman, M. Generating Facial Expressions for Speech. *Cognitive Science*, 20, 1, 1994, 1-46.

[12] Ekman, P. and Friesen and W. V. *Unmasking the Face: A Guide to Recognizing Emotions from Facial Clues.* Prentice-Hall, 1975.

[13] Cassell, J., Pelachaud, C., Badler, N., Steedman, M., Achorn, B., Bechet, T., Douville, B., Prevost, S., and Stone, M. Animated Conversation: Ruled-based Generation of Facial Expressions Gesture and Spoken Intonation for Multiple Conversational Agents. In *Computer Graphics (Proceedings of ACM SIGGRAPH'94)*, 1994, 413-420.

[14] Perlin, K., and Goldberg, A. Improv: A System for Scripting Interactive Actors in Virtual Worlds. In *Proceedings of ACM SIGGRAPH'96*, 1996, 205-216.

[15] Kurlander, D., Skelly, T., and Salesin, D. Comic Chat. In *Proceedings of ACM SIGGRAPH'96*, 1996, 225-236.

[16] Chi, D., Costa, M., Zhao, L., and Badler, N. The Emote Model for Effort and Shape. In *Proceedings of ACM SIGGRAPH'00*, 2000, 173-182.

[17] Cassell, J., Vilhjalmsson, H., and Bickmore, T. Beat: The Behavior Expression Animation Toolkit. In *Proceedings of ACM SIGGRAPH'01*, 2001, Los Angeles, 477-486.

[18] Kuratate, T., Munhall, K. G., Rubin, P. E., Bateson, E. V., and Yehia, H. Audio-Visual Synthesis of Talking Faces from Speech Production Correlation. In *Proceedings of Eurospeech'99*, 1999.

[19] Yehia, H., Kuratate, T., and Bateson, E. V. Facial Animation and Head Motion Driven by Speech Acoustics. In *5th Seminar on Speech Production: Models and Data.* 265-268.

[20] Costa, M., Chen, T., and Lavagetto, F. Visual Prosody Analysis for Realistic Motion Synthesis of 3D Head Models. In *Proceedings of International Conference on Augmented, Virtual Environments and Three-Dimensional Imaging*, 2001, 343-346.

[21] Graf, H. P., Cosatto, E., Strom, V., and Huang, F. J. Visual Prosody: Facial Movements Accompanying Speech. In *Proceedings of IEEE International Conference on Automatic Faces and Gesture Recognition*, 2002, 381-386.

[22] http://www.vicon.com

[23] http://skal.planet-d.net/demo/matrixfaq.htm

[24] Boersma, P. and Weenink, D. *Praat Speech Processing Software*, Institute of Phonetics Sciences of the University of Amsterdam. http://www.praat.org.

[25] Dasarathy, B. V. *Nearest Neighbor Pattern Classification Techniques*, IEEE Computer Society Press, 1991.

[26] Friedman,J. H., Bentley, J. L., and Finkel, R. A. An algorithm for finding best matches in logarithmic expected time. *ACM Transaction on Mathematical Software*, 3, 3, 1977, 209-226.

[27] Hastie, T., Tibshirani,R., Friedman, J. *The elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer-Verlag, 2001.

[28] Pierre, D. A. *Optimization Theory With Applications.* General Publishing Company, 1986