VISUAL SPEECH SYNTHESIS USING QUADTREE SPLINES

Xue-Wen Chen
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213 USA

Jie Yang
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA

ABSTRACT

In this paper, we present a method for synthesizing photo-realistic visual speech using a parametric model based on quadtree splines. In an image-based visual speech synthesis system, visemes are used for generating an arbitrary new image sequence. The images between visemes are usually synthesized using a certain mapping. Such a mapping can be characterized by motion parameters estimated from the training data. With the quadtree splines, we can minimize the number of motion parameters for a given synthetic error. The feasibility of the proposed method has been demonstrated by experiments.

1. INTRODUCTION

Face synthesis plays a very important role in creating an intelligent agent for human computer interaction. Much research has been directed to developing autonomous software agents that can communicate with humans using speech, facial expression, and gesture [3, 4, 5, 6, 7]. Most of those systems are based on a phonemic representation (phoneme or viseme). Typically, the phonemic tokens are mapped onto lip poses and the lips are synthesized from either real images (e.g., Video Rewriting [4]) or graphic approaches (e.g., Baldi [6]). However, different tasks impose different requirements on naturalness (cartoon or realistic face), usability, and real-time implementation. In this research, we are interested in developing a multimodal communication agent for Internet applications [10]. The system has been designed to translate spoken utterances into another language, and to produce an audio-visual output with the speaker's face and synchronized lip movements. The system synthesizes not only lip movements based on translated text, but also eye gaze based on user's location. The system also uses eye blinking and other facial expressions to make the interaction more realistic. Our system is for ordinary users. One of the objectives in the development is to minimize a user's effort in creating a new agent. The current system takes as short as 10 minutes to create a new agent.

One concern in designing Internet applications is the amount of data to be transferred through the network. In order to reduce the database size, our current system only stores small regions of visemes in the database. The images between visemes are synthesized using bi-linear transformation. This could significantly reduce the database size. But sometimes it causes big synthetic errors. In fact, change in images between visemes is nonlinear. Brand attempted to use HMM to learn the representations [3]. Ezzat and Poggio used optical flow to characterize such change [6]. There are several advantages to using motion parameters to represent changes between visemes. First, motion estimation requires little data. This is very attractive for quickly creating a new agent. Second, motion parameters can characterize nonlinear properties between images with much less data than original images. Third, it is very easy to modify the image sequence if needed. However, optical flow is not the best motion model to characterize visemes. There is a lot of redundance in optical flow: many pixels have the same motion. We can characterize motions by patches instead of pixels. In this paper, we propose the use of quadtree splines for synthesizing photo-realistic visual speech. This is motivated by Szeliski and Shum's work [11]. They presented an approach to describe the motion field as a collection of smoothly connected patches of varying size using quadtree splines. With the quadtree spline representation, we can minimize the number of motion parameters for a given synthetic error. The feasibility of the proposed method has been demonstrated by experiments. The organization of this paper is as follows: in Section 2, we describe our current system; in Section 3, we introduce motion estimation using quadtree splines; in Section 4, we present application of quadtree splines to visual speech synthesis; in Section 5, we conclude the paper.

2. FACE TRANSLATION SYSTEM

Face Translation system was developed at the Interactive Systems Lab in Carnegie Mellon University for a language translation task [10]. The system not only can translate a spoken utterance into another language, but also can produce an audio-visual output with the speaker's face and synchronized lip movement. Face Translation uses image processing and morphing technologies to generate images between phonemes. Furthermore, Face Translation synthesizes not only lip movements based on translated text, but also eye gaze based on user's location. Face Translation also uses eye blinking and other facial expressions to make more realistic interaction.

The system is designed for Internet applications. In the initialization phase, the user is asked to read a few sentences. The visemes are selected by phoneme segmentation from speech recognition and then mapped into the target language. Some facial expressions, such as eye gaze and eye blinking, are captured automatically by the system at the same time. These results are stored in a

database. The database is transmitted to the receiving end. At the receiving end, the system can generate visual output based on a few pre-stored images. Face Translation employs the JANUS system [8,9] for speech recognition and translation, and the FESTIVAL Text to Speech System [1] to generate audio speech from text strings. The system outputs the synchronized audio and video. Figure 1 shows the working process of the Face Translation system.

We have applied the system to a travel-planning task where a user communicates with other user(s) via Internet. We call this user "agent" and other user(s) at remote site "client(s)." The agent and client(s) speak different languages. But the agent can talk to the client(s) via the Face Translation system. The client will see the agent's face speaking the translated sentences with synchronized lip movements. The agent's eyes will also look at the client during the conversation. The system works as follows. When the agent speaks to the system, the speech-to-speech translation module translates the spoken utterance into an intermediate language and then maps onto the target language. The string of the translated text is sent to the receiving end. At the receiving end, the system synthesizes synchronized acoustic and visual speech output based on text input. The eye gaze is determined by the location of the client detected by the location detector.

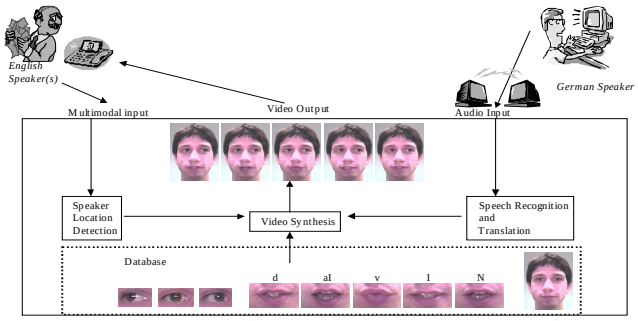


Figure 1. Face Translation system

3. MOTION ESTIMATION USING QUADTREE SPLINES

The motion estimation problem can be formulated as follows: given a sequence of images $I_t(x,y)$ which were formed by locally displacing a reference image $I(x,y)$ with horizontal and vertical displacement fields, i.e.,

$$I_t(x+u_t,y+v_t) = I(x,y),$$

we wish to recover the displacement fields (u_t,v_t) from the reference image $I(x,y)$. The problem is to estimate parameters using different models, such as translation, affine, and projective models. The complexity of motion estimation largely depends on the model used.

A large number of approaches have been proposed to solve this problem. The approaches include optical flow (general motion) estimators, global parametric motion estimators, local parametric motion estimators, constrained motion estimators, stereo and multiframe stereo, and hierarchical (coarse-to-fine) methods. The optic flow field is the 2D distribution of apparent velocities that can be associated with the variation of brightness patterns. The components of optical flow can be considered as describing the velocity of each individual point in the image. Based on the constant-brightness assumption, the optical flow can be computed by:

$$E_x u + E_y v + E_t = 0,$$

where E is the brightness of the image, E_x, E_y, and E_t are the derivatives $\frac{\partial E}{\partial x}$, $\frac{\partial E}{\partial y}$, and $\frac{\partial E}{\partial t}$ respectively, x and y are any orthogonal pair of direction vectors on the surface of the terrain, and t is the number of the image. For the calculation of two-dimensional optical flow, a variety of methods has been used and a good overview can be found in [2].

The optical flow characterizes motion at the pixel level. Figure 2 shows an example of the optical flow pattern from the sequence of images of a rotating Rubik's cube. It is clear that many pixels have the same motion and some pixels don't have any motion at all. Therefore, we can characterize motions by patches instead of pixels. One way to do this is to divide the image into patches and model the motion for each patch.

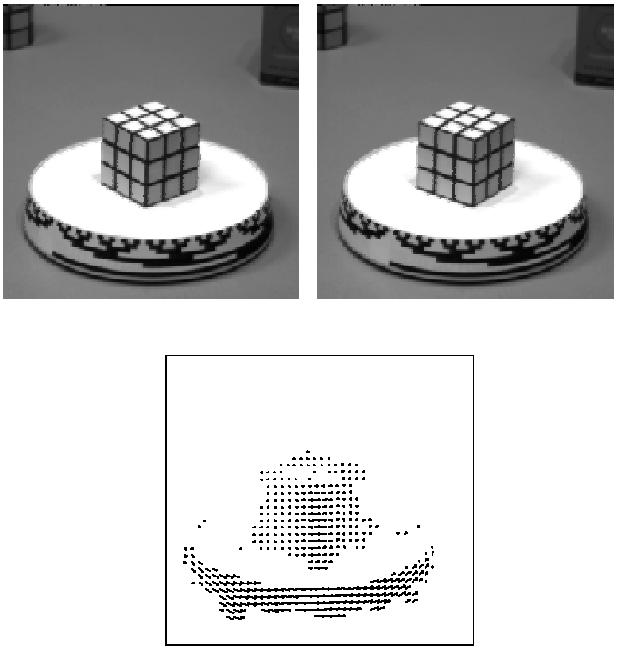


Figure 2. An example of optical flow

The basic idea of motion estimation with quadtree splines is to recursively subdivide an image into square patches of varying size and then match these patches to subsequent

frames in a way that preserves inter-patch motion continuity [11]. The quadtree spline provides a convenient representation to use patches with different sizes for motion estimation, while maintaining inter-patch continuity. Using quadtree splines, we can minimize parameters for modeling the motion. The motion at each pixel will be interpolated by splines.

To model the local fields, we use a 2D projective transformation defined as follows:

$$u(x,y) = \frac{m_0 x + m_1 y + m_2}{m_6 x + m_7 y + 1} - x\,,$$

$$v(x,y) = \frac{m_3 x + m_4 y + m_5}{m_6 x + m_7 y + 1} - y\,.$$

Thus, for each patch, we require only 8 parameters. These parameters can be estimated using as few as four-pixel flows in the patch. To decide whether to split a patch further into four smaller patches, we then calculate the sum of the square of the difference (SSD) between the flow vectors computed from the motion estimation algorithm and the flow vectors estimated from the transformation model over all pixels in the patch, and compare it to a preset threshold. If SSD is less then the threshold, the parametric motion model is valid inside this patch; otherwise, this patch will be further subdivided. Starting with the whole image, we subdivide recursively until either SSDs for all patches fall below the threshold or the smallest patch size is reached. In the next section, we will show how to apply this method to visual speech synthesis.

4. VISUAL SPEECH SYNTHESIS

In this section, we discuss the application of a parametric model using quadtree splines to visual speech synthesis. The parameters of the model are estimated from the training data. The model is then used to synthesize new image sequences.

Figure 3 shows the procedure of the training process. The motion parameters are estimated from recorded video sequences. The training data include both audio and video: the audio track is used for phoneme segmentation; the associated time labels are then used to segment the video track as well. Optical flows from each viseme image to every other viseme are computed and are stored in a small parametric space -- the quadtree splines estimated from motion. The method has been described in detail in section 3. This will significantly reduce the size of data to be transferred through networks.

Figure 4 shows two-end visemes. Figure 5 represents the corresponding optical flows with the first viseme as reference image. The motion is represented by arrowed flow vectors. As we can see, most motions occur in the region below the speaker's nose, especially around the speaker's mouth and neck. Figure 6 is the quadtree spline based representation. Apparently, most regions are uniform.

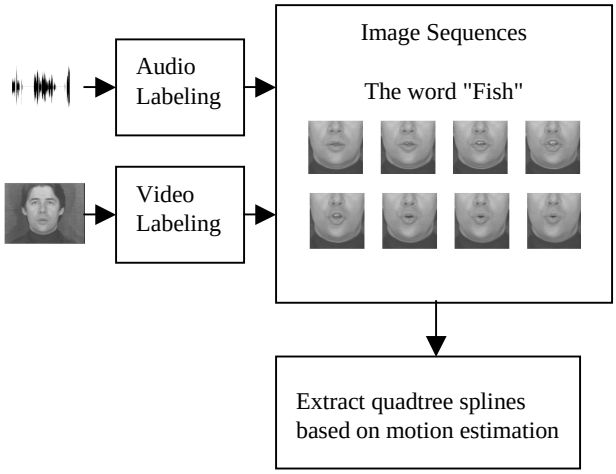
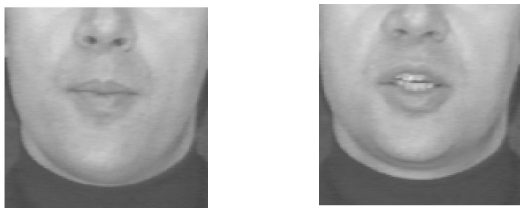


Figure 3. Overview of the training process

.



(a) (b)

Figure 4. Two-end visemes:
(a) the starting viseme, and (b) the ending viseme

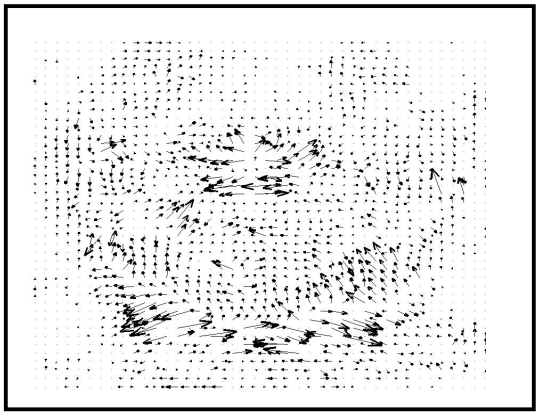


Figure 5. Optical flows

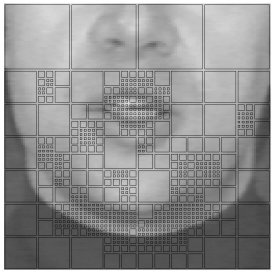


Figure 6. The quadtree spline representation

The key frames and quadtree splines are used to synthesize the new image sequence. We use image-based morphing techniques [12] for producing transitions between images. These techniques combine 2D interpolations of shape and color to create dramatic special effects. Morph transformation allows to generate images that are strikingly lifelike and visually convincing.

Given the two-end visemes $\boldsymbol{I_0}$ and $\boldsymbol{I_1}$, our goal is to synthesize intermediate images. The first step is to compute the correspondence map, which can be decoded from the quadtree spline based motion parameters. By multiplying the flow vectors with a scale factor α between 0 and 1, a series of warped intermediate images $\boldsymbol{I_0}^{warped}(\alpha)$, which approximate the transitions between the starting viseme and the ending viseme, is produced. Obviously, this warping cannot model the difference between the starting viseme and the ending viseme, e.g., teeth in Fig. 4(b). To synthesize more realistic images, we need to forward warp $\boldsymbol{I_0}$ and to backward warp $\boldsymbol{I_1}$ along the associated correspondences to the intermediate images. Suppose that the warped intermediate images are $\boldsymbol{I_0}^{warped}(\alpha)$ and $\boldsymbol{I_1}^{warped}(1\text{-}\alpha)$ which approximate the transformation between $\boldsymbol{I_0}$ and $\boldsymbol{I_1}$. We then add the weighted warped-intermediate images with respective cross-dissolve or blending parameters to produce the final morphed image $\boldsymbol{I}^{morph}$ [7]:

$$\boldsymbol{I}^{morph} = (1\text{-}\alpha)\, \boldsymbol{I_0}^{warped}(\alpha) + \alpha\, \boldsymbol{I_1}^{warped}(1\text{-}\alpha)\,.$$

The parameter α is between 0 and 1. As the intermediate frame moves away from the starting viseme, α increases gradually from 0 to1.

Figure 7. Synthesized intermediate frames

Figure 7 shows an example of the synthesized image sequence. It is clear that synthesized intermediate frames of images are remarkable realistic.

5. CONCLUSION

We have presented a new parametric model for synthesizing photo-realistic visual speech using quadtree splines. With the quadtree splines, we can minimize the number of motion parameters for a given synthetic error. The new model can efficiently and effectively characterize the non-linear changes in visual speech synthesis. The parameters of the model can be learned from the training data. We are currently working on real-time implementation of the proposed approach.

REFERENCES

[1] A. W. Black, P. Taylor, and R. Caley. Festival. *www.cstr.ed.ac.uk/projects/festival.html*, The Centre for Speech Technology Research (CSTR) at the University of Edinburgh,1998.

[2] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *Int. J. of Computer Vision*, 12(1):43-77, 1994.

[3] M. Brand. Voice puppetry In *Computer Graphics Proceedings*. P. 463, 1999

[4] C. Bregler, M. Covell, and M. Slaney. Video rewrite: Driving visual speech with audio. In *Computer Graphics Proceedings*, pages 353-360, 1997.

[5] E. Cosatto, H. P. Graf. Photo-realistic talking-heads from image samples. In *IEEE Transactions on Multimedia*, vol.2, no.3, pages 152-163, 2000.

[6] M. Cohen, J. Beskow, and D. W. Massaro. Recent developments in facial animation: An inside view. In *Proceedings of Auditory-Visual Speech Processing (AVSP 98)*, 1998.

[7] T. Ezzat and T. Poggio. Visual speech synthesis by morphing visemes. In *MIT A.I Memo No. 1658*, May 1999.

[8] A. Lavie, A. Waibel, L. Levin, M. Finke, D. Gates, M. Gavald, T. Zeppenfeld, and P. Zhan. Janus-iii: Speech-to-speech translation in multiple languages. In *Proceedings of ICASSP*, 1997.

[9] L. Osterholtz, C. Augustine, A. McNair, I. Rogina, H. Saito, T. Sloboda, J. Tebelskis, A. Waibel, and M. Woszczyna. Testing generality in Janus: A multi-lingual speech translation system. *In Proceedings of ICASSP*, 1992.

[10] M. Ritter, U. Meier, J. Yang, and A. Waibel. Face translation: A multimodal translation agent. In *Proceedings of Auditory-Visual Speech Processing (AVSP 99)*, 1999.

[11] R. Szeliski and H.-Y. Shum. Motion estimation with quadtree splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.18, no.12 pages 1199-210, 1997

[12] G. Wolberg, Digital Image Warping, *IEEE Computer society Press*, Los Alamitos, CA, 1990.