LOW BIT-RATE VIDEO STREAMING FOR FACE-TO-FACE TELECONFERENCE

Zhen Wen[*], *Zicheng Liu*[†], *Michael Cohen*[‡], *Jin Li*[§], *Ke Zheng*[¶], *Tomas Huang*[‖]

ABSTRACT

Face-to-face video teleconferencing is very important for real time communication. Current teleconferencing application uses standard video codec, such as MPEG1/2/4, for the compression of face video. It either requires high bandwidth for high quality video transmission, or the transmitted face video be blurred at low bit-rate. In this paper, we present a system for real-time coding of face video at low bit-rate. There are two main contributions. First, we improve the technique of long term memory prediction by selecting frames into the database in an optimal way. A new frame is selected into the database only when it is significantly different from those frames which are already in the database. In this way, the database can cover a wider range of images. Second, we incorporate the prior knowledge about faces into the long term memory prediction framework. The prior knowledge includes: (1) facial motions are repetitive such that most of them can be reconstructed from multiple reference frames; and (2) different components of the face and the background could tolerate different level of error because of different perceptual importance. Experiments show that at similar PSNR the proposed system works much faster and achieves better visual quality than standard H.264/JVT codec.

1. INTRODUCTION

Face-to-face video communication is an important component for real time communication. Current commercial video teleconference solutions (such as $Polycom^{TM}$ [10]) usually require high bandwidth, thus limit their usage. Meanwhile, much research has been done for low bit-rate face video coding. One class of methods is model-based coding, such as MPEG-4 face animation standard [5]. In these methods, facial image is coded as changes of face model parameters so that very low bit-rate can be achieved. However, it is difficult to make the synthesized face model look natural and match the input video. Compared to model-based techniques, traditional waveform based coding techniques (such as H.26X, MPEG-1/2/4 etc) are fully automatic and robust. However, the quality of low bit-rate video teleconference is usually not good enough. The computational complexity for sophisticated waveform video coder, such as H.264, can also be very high.

In this paper, we present a low bit-rate face video streaming system, which incorporates the prior knowledge about faces into a traditional waveform based technique. We observe that facial motions are highly repetitive so that most of face images can be reconstructed from multiple reference frames through texture synthesis technique. Furthermore, different face regions (eye region, mouth region, and the rest of the face) tolerate different level of error because of different perceptual importance.

Therefore, similar to long term memory prediction technique, we maintain a database of reference frames at run time. But our system differs from the long term memory prediction method in that we select frames into the database in an optimal way. A new frame is selected into the database only when it contains very different face appearance from those which are already in the database. In this way, we minimize the size of the database while covering a wider rage of face variations. To take advantage of the perceptual differences among different face regions, we divide a face video frame into the face and background, and further decompose the face into multiple layers including the head, eye, and mouth. A different set of reference frames are maintained from each face region. Different layers are then coded with different quality based on their perceptual importance. The face layers are assigned with more bits and are coded with higher quality. Among the face components, eyes and mouth are considered the most important and they are coded with the highest quality setting. Our coder is more robust than the traditional model-based coder because it does not construct an explicit face model.

2. RELATED WORK

Following the work of MPEG-4 face animation standard, there has been a lot of research work on model-based image coding such as [7, 9, 1, 11]. A challenging task is to automatically construct the realistic 3D face model for animation and estimate the parameters of facial motion, which still remains an open problem today. To achieve better photo-realism, some model-based methods such as [3] use 2D image based model. The facial motion is modeled from extensive image samples by subspace model. In such approach, the precise alignment of images are crucial to avoid blurry

[*] University of Illinois at Urbana Champaign. zhenwen@ifp.uiuc.edu.
[†] Microsoft Research, zliu@microsoft.com
[‡] Microsoft Research, mcohen@microsoft.com
[§] Microsoft Research, jinl@microsoft.com
[¶] University of Washington, kzheng@cs.washington.edu
[‖] University of Illinois at Urbana Champaign. huang@ifp.uiuc.edu

in synthesis.

On the other hand, traditional waveform based coding techniques such as H.261/H.263/H.264 and MPEG-1/2/4 [6] encode the video signal as a waveform without specific knowledge of the semantic content of the frames. Compared to the model-based methods, they are more robust and practical. However, it is challenging to achieve good quality under low bit-rate. Moreover, sophisticated technology can also be very computational expensive.

Recently Eisert et al. [4] tried to use model-based coder to aid waveform based coding. The synthesized frame was used as one of the reference frames for prediction. In case the model fails, the synthesized frame can be used by block-based motion-compensated prediction, and the followup waveform coder only needs to encode the residue. However, it still requires highly accurate face modeling and face motion analysis to make the model-based prediction module useful.

In this paper, we propose an approach that uses very limited knowledge of the face to improve the coding efficiency. We only decompose a face video into specific layers such as the head, eyes, and mouth. The decomposition is based on approximate face position, which can be estimated easily and robustly by the existing face detection technique. We encode each layer with a scheme similar to the multi-reference frame coder in H.264, however, we update our reference frame database in a optimal way. New frames are selected into the reference frame database only when it is very different from the frames which are already in the database. In addition we encode different layer with different quality setting based on their perceptual importance. We use a management scheme to select good references, and constrain the search within each category so that the coding can be done in real-time.

3. LOW BIT-RATE FACE VIDEO STREAMING

3.1. System overview

For our face video compression system, we assume the following: (1) the camera is static; (2) there is only one talking face, which is the focus of attention; and (3) the conversation lasts several minutes. We consider the background to be less important, and update it infrequently.

The architecture of our system is illustrated in figure 1. The input video frame is first processed to locate the face and its component. After the face position is found, we further decompose the face video into four layers: (1) the head (without eyes and mouth), (2) eyes, (3) mouth, and (4) background. We only encode the background at initial frame, and update it very infrequently. The head, eyes and mouth layers are then coded using multi-reference frame prediction with different quality control. We assume different perceptual importance of different layers. From the most important to the least important, the four layers are ranked as follows: mouth, eyes, head and background. The more important a layer is, the better the quality should be. We use SAD (sum of absolute difference) between the reconstructed frame and original frame as quality measure. Thus an important layer such as the mouth is encoded with the smallest reconstructed SAD, while the least important background layer is updated very infrequently.

The coding of each layer is similar to a traditional multi-reference frame coder, such as H.264, but we allow a larger number of reference frames and the database is updated selectively. For the layers which do not have a lot of variations such as the background or the face skin portion, the required number of reference frames is small. For face regions with larger variations such as the mouth, a larger number of reference frames is needed. But fortunately, the mouth region is much smaller than the entire frame. Therefore, the amount of memory required is still manageable.

For the coding of each layer, we first select the best matches from the corresponding reference database using fast matching. Based on the matching quality, we decide which information to be sent. If the best match is good enough, we may use the best match directly as reconstruction. If the best match is marginally good, we will use the best match as a prediction, and further transmits the residue error. If the best match is still a poor match, it will be abandoned, and we will send the current layer of the current frame with DCT transform and entropy coding, similar to the intra-coding in H.264. We also add the current layer of the current frame as a new reference in the database.

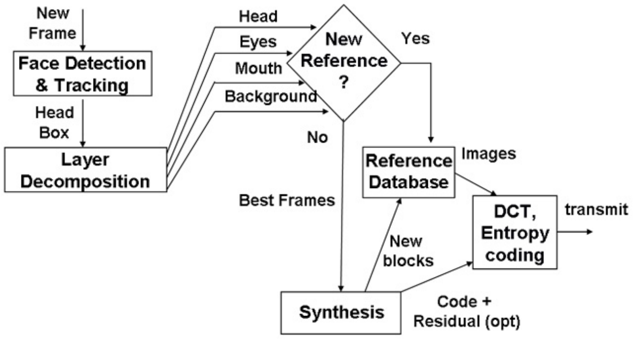

Fig. 1: *System architecture.*

3.2. Layer decomposition based on face detection

We decompose the face video into multiple layers, where each layer is encoded independently. We use the face detection algorithm in [8] to locate a rectangular box containing face. For real-time detection, the detector scans only in the neighborhood of the face location in the previous frame. After the face box is found, we locate eyes and mouth area based on their relative ratio in the face box. The precise facial feature locations are not crucial in our system, thus making the face detection more robust than that required by the model-based coding.

3.3. Multi-references prediction

To code each individual layer of the face video, we use multi-reference frame block-based motion-compensated prediction. The approach is similar to the multi-reference frame coder of H.264, except that we use a much larger number of reference frames and we update the database selectively. In the reference database, each reference frame is a reconstructed face image patch. The reference frame database is then searched for the best matching head, eye, and mouth during the encoding process.

We further decompose each layer of the current frame into block of size 16×16. Each block is then searched in the reference frame database for its best match. We use the SAD value as the criterion of the match, and the best matching block has the smallest SAD value with regard to the current block. We index the best match by the reference frame number and the offset in the reference frame. To find match in real-time, we use the following strategies: (1) multi-resolution matching, where 3-level pyramid is used; and (2) search considering temporal and spatial smoothness. Temporal smoothness means adjacent frames are close. Therefore, the references of the previous frame are used as the references of the current frame initially, and updated later when new image blocks are encountered. On the other hand, the spatial smoothness of a frame implies that good matches of many blocks in the frame probably come from the same reference frame, thus we can limit the number of the references used for one frame.

After the best match is found, it is evaluated according to its SAD value. If the match is good enough, we simply use the match as reconstruction, and just entropy encode the match index, which includes the reference frame number and the offset in the reference frame. If the match is marginally good, we use the match as the prediction, and further encodes the residual error between the current block and the match block. The residual error is DCT transformed and entropy encoded, just as an inter-coded block in traditional video coder. In addition, the match index is also sent to the decoder. If the match is poor, the current block is coded without reference to the match block. It is directly DCT transformed and entropy encoded, just as an intra-coded block in traditional video coder, and no match index needs to be sent. For a face frame, if a large number of blocks are poorly matched, we may optionally force the whole face frame to be intra-coded, and sent without reference. The details will be described in the following section.

3.4. Reference database management

The reference database is built progressively based on the already encoded face video. It is also synchronized between the encoder end and the decoder end. For a newly coded face video frame, if the match is good or marginally good for every block, the frame can be reconstructed using the existing reference database, and thus no new reference will be added. Otherwise, the reference database is updated as follows: (1) If the number of poorly matched blocks, excluding blocks along face boundary, exceeds 20% of the face area, we consider that the whole face to be poorly matched. We force the entire face to be intra-coded, and added as a reference to the multi-reference frame database; (2) If the poorly matched blocks are less than 5% of the total face area, or are all along face boundary, the poor match is mostly for the blocks between the face and the background. The poorly matched block is intra-encoded, however the frame is not added into the reference database; (3) Otherwise, we intra-encode the poorly matched block, inter-encode the marginally good matched block, and do not encode the good matched block. Moreover, the reconstructed frame is added into the reference database.

To prevent reference database from consuming too much memory and the search time from becoming too long, we prune the database as necessary. For each reference frame in the database, we keep a counter to record the last time it is used. If the counter exceeds a certain threshold, the reference frame is removed from the database. Currently, we set the threshold to be 150, meaning that a reference frame is removed from the database if it is not used in 150 frames.

4. EXPERIMENTAL RESULTS

In this section, preliminary experimental results are presented. We implement our face video codec on a PC with Pentium 4 2.0 GHz processor and 512 MB memory. The system uses a IEEE 1394 camera, which can capture frames of resolution 640×480 at up to 15 frames per second. Figure 2 shows a snapshot of an input video frame. The red boxes indicate the layer decomposition result from the face detection where the size of the face box (the largest rectangle) is about 200×230 (notice that this number changes from frame to frame). We crop out the face region and use the cropped video sequences to measure the performance of our system and to compare with the benchmark coder.

The reason that we do not use standard test sequence, such as Akiyo, in the standard video database is because our codec benefits significantly from longer face video sequence, which lasts several minutes and includes thousands of frames (1500 frames in the test video). We do not compare our face video coder with other model-based coders because no model-based coders are public available yet. In the preliminary implementation, the inter-code mode has not been implemented. Thus, a block is either encoded with match indices, or is intra-encoded. Our current system runs at $5 \sim 9$ frames per second (fps) depending on the size of the face in the video frame.

The benchmark coder is the JVT reference software JM 4.2 obtained from [2]. It is a joint effort between ISO MPEG and ITU H.26x after the successful H.264 development, and

Codec	Face video coder	JVT coder
Y PSNR (dB)	32.1	32.2
U PSNR (dB)	43.2	41.4
V PSNR (dB)	39.3	39.8
Bit-rate at 15Hz (Kbit/s)	19.1	19.8
Coding time of one frame (ms)	153	1370

Table 1: *Performance comparison of the face video coder to the JVT coder*

represents the state-of-the-art in low bitrate video coding. We use the average decoding peak signal-to-noise ratio (PSNR) of the face area to measure the compression performance between the face video coder and the JVT coder. Other comparison metrics include the coding bit-rate, and the encoding time of one frame. We compare the performance of the proposed face video coder with that of the JVT coder. Because our coder bears similarity to the long term memory prediction mode of H.264/JVT, the mode is enabled with parameter: $NumberReferenceFrames = 5$ and $AdditionalReferenceFrame = 19$.

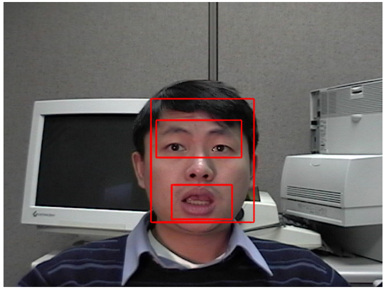

Fig. 2: *A face video frame. The red boxes illustrate head, eyes and mouth layer.*

Only one block size 8×8 is used in motion estimation and compensation to speed up the JVT coder. Moreover, only the first frame is encoded as I frame, and the rest frames are encoded as P frames. The comparison results between the face video coder and the JVT coder are shown in table 1. We observe that our face video coder achieves relatively the same PSNR performance compared to the JVT coder with one tenth of the computational complexity. Note that for the face video coder, the coding time include face detection and tracking time. On the average, the face video coder uses less than 20 references per frame. Figure 3 shows one of the reconstructed frame by JVT and the face video coder, respectively[1]. Even though the PSNR value of face video coder is slightly lower because it allows larger SAD value for less important facial area, the face video coder provides far superior visual quality. Compared to the JVT reconstruct frame, far more facial details are preserved, such as wrinkles in eyes and mouth area.

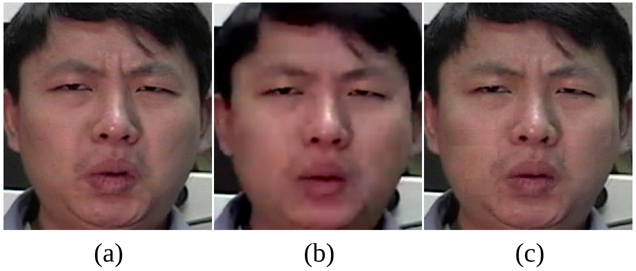

Fig. 3: *(a) Original frame; (b) and (c): Reconstructed frame 485 with (b) JVT coder (Y PSNR=32.3dB, 3144 bits); (c) the face video coder (Y PSNR=32.0dB, 2902 bits).*

5. CONCLUSION

In this paper, a novel low bit-rate face video streaming system is presented for face-to-face video teleconference. Our system improves the traditional long term memory prediction technique in that we choose new frames into the reference frame database in an optimal manner. By incorporating prior knowledge of face, we are able to significantly improve the efficiency of the traditional long term memory prediction approach without affecting the robustness. Experiment shows that our system is more computationally efficient and transmits video with better visual quality, compared with the existing systems with standard video codec. Future work includes: (1) improving the speed of the system by employing better temporal-spatial constraints; (2) optimizing intra-coding and entropy coding to further reduce bandwidth usage; and (3) reducing the blocking artifacts in the reconstructed video using the deblocking filter of an existing video coder, such as H.264.

6. REFERENCES

[1] K. Aizawa and T. S. Huang. Model-based image coding. In *Proc. IEEE*, pages 259–271, 1995.

[2] H. codec reference software. ftp://ftp.imtc-files.org/jvt-experts.

[3] E. Cosatto and H. P. Graf. Photo-realistic talking-heads from image samples. *IEEE Trans. on Multimedia*, 2(3):152–163, 2000.

[4] P. Eisert, T. Wiegand, and B. Girod. Model-aided coding: A new approach to incorporate facial animation into motion-compensated video coding. *IEEE Trans. on CSVT*, 10(3):344–358, 2000.

[5] ISO/IEC JTC1/SC29/WG11 N1902. *Text for CD 14496-2 Video*, November 1997.

[6] ITU-T Recommendation H.263 Version 2 (H.263+). *Video coding for low bitrate communication*, January 1998.

[7] H. Li and et al. 3-d motion estimation in model-based facial image coding. *IEEE Trans. on PAMI*, pages 545–555, 1993.

[8] S. Z. Li and et al. Real-time multi-view face detection, tracking, pose estimation, alignment, and recognition. In *IEEE CVPR Demo Summary*, 2001.

[9] J. Ostermann. Object-oriented analysis-synthesis coding (ooasc) based on the source model of moving flexible 3d objects. *IEEE Trans. on Image Processing*, 3:705–711, 1994.

[10] Polycom. http://www.polycom.com.

[11] H. Tao. *Non-Rigid Motion Modeling And Analysis In Video Sequence For Realistic Facial Animation*. PhD thesis, University of Illinois at Urbana-Champaign, ECE Department, 1998.

[1] Parts of the video clips are available at http://www.ifp.uiuc.edu/~zhenwen/icme04/