A FEEDBACK SYSTEM FOR GRAPHICS VIDEO CODING AND NETWORKING

Dongmei Wang[1] *and Russell M. Mersereau*[2] *

[1]Agere Systems
StarCore Technology Center
Atlanta, Georgia 30328 USA
[2]Georgia Institute of Technology
Center for Signal and Image Processing
Atlanta, GA 30332-0250 USA

ABSTRACT

This paper presents feedback design of a desktop visual communications system. The system consists of video coding using a 3-D graphics model and video transmission over the Internet. To reduce the overall video quality degradation in visual communications caused by coding and networking errors, we jointly researched the compression and transmission of video signals. In building the 3-D graphics model-based coding structure, instead of using feedforward idea based on pixel intensity, we develop a three-level signal representation and an analysis-by-synthesis feedback framework. In prototyping the video over IP in desktop conferencing, we adapt the video encoders coding rates based on feedback of the network states and receivers. We contribute to the analysis, modeling, and transmission of multimedia signals for desktop visual communications.

1. INTRODUCTION

A multimedia communications system includes two parts– data compression and data transmission [1]. Data compression targets the best tradeoff between coding quality and bit rate. Data transmission includes the network protocols and their performance evaluation. Among the quality of service performance parameters, delay and packet loss of video data are two major concerns in real world visual communications.

In this work, we develop a very low bitrate video coding and transmission system using feedback design strategy. For video compression, we build a 3-D graphics model-based coding framework. We develop a three-level signal representation that connects the video intensity signal, 2-D facial shape, and a 3-D head-and-shoulder graphics model. We model 3-D nonrigid motion in the 3-D graphics domain to assist motion analysis in input video. For video transmission, we prototype a desktop Internet videoconferencing system and control the transmission rate based on the information fed back from the current network session and participating end users.

2. BACKGROUND

A 3-D graphics model-based coder requires the use of object-based encoding and decoding methods that are completely different from conventional block-based video coding standards such as MPEG1, MPEG2, and H.263. The encoder performs human facial feature extraction, texture mapping, head-motion estimation, facial deformation extraction, and parameter coding. The decoder extracts the parameter set (texture, shape, and motion), synthesizes video using 3-D graphics models, and evaluates the synthesized video quality, as shown in Figure 1. The geometrical

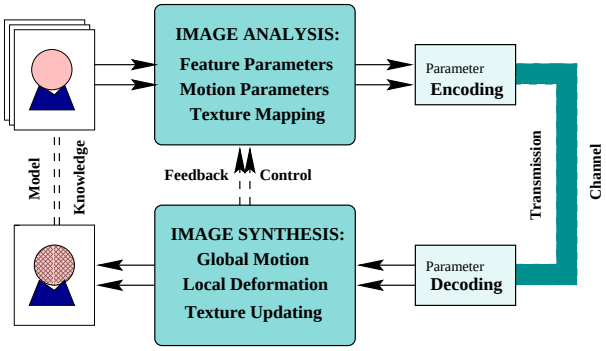

Figure 1. Model-based Coding System

modeling of head-and-shoulder include volume, surface, and polygonal representations. We chose a polygonal surface representation because of its rendering efficiency. To avoid the cracks in the rendered image due to non-planar facets when the post-transformation model is projected back onto the display plane, we adopt a modified version of the 3-D graphics head-and-shoulder model CANDIDE [2]. CANDIDE forms a triangular mesh of the head using 160 vertices. With triangular facets, all three vertices must always lie on a plane, even after transformation and numerical roundoff. In texture mapping, we manually adjust the generic 3-D head-and-shoulder graphics model to fit to a natural image of a neutral facial expression (Figure 5(b),(c)).

Unlike conventional video coders, a 3-D graphics model-based coder extracts 3-D object structure and motion information rather than 2-D displacement vectors of image intensity pixels, which further reduces the temporal redundancy in the video. In a head-and-shoulder scene, global motion accounts for head motion and local nonrigid motion accounts for deformations corresponding to facial expressions. Previous work on 3-D motion estimation of head-and-shoulder scenes includes (1) rigid head motion extraction with no facial expression [3]; (2) a roughly stationary head motion with changing expressions [4]; and (3) a model that estimates both global and local motion [5]. This last model is based on spatial-temporal derivatives of the first-level pixel information, and uses linear movement of 3-D vertices to approximate the nonlinear behavior of muscle stretching and contraction.

For video transmission over the Internet, the TCP/IP protocol stack is used. TCP (Transmission Control Protocol) is slow and reliable. UDP (User Datagram Protocol) provides faster transmission, but its packets may get lost during routing, may arrive too late, or may be dropped due to buffer overflow. In visual communications, where real-time is a key concern, UDP is often used. To reduce

*D.Wang performed the work at Georgia Institute of Technology. Thanks to Intel Corporation for support.

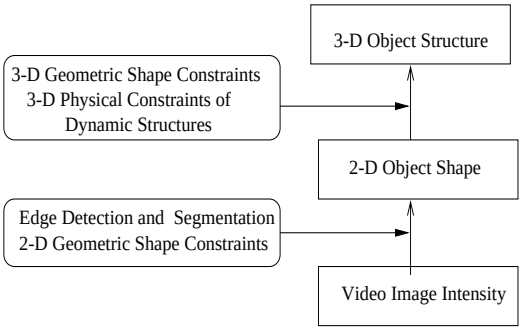

Figure 2. Three-level Signal Representation

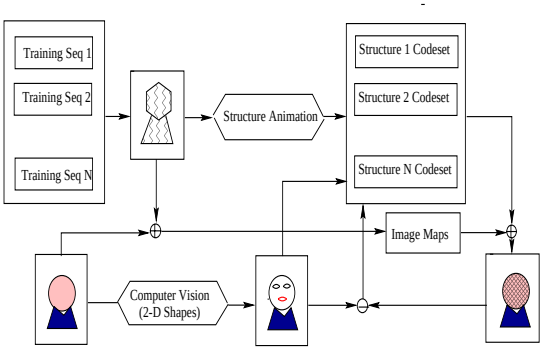

Figure 3. Feedback System Design

the quality degradation when using UDP, we propose an application content-level protocol that feeds the information of the network states and receiver capability back to the encoder to control coding rate. How do we define the network states and the receiver capability? In a multicast session, receivers may have different rendering speeds. We categorize the network and receiver states as "loaded" if the client detects packet loss caused by low receiving power and network congestion; "adequate" if the end user decodes and displays video properly; or "need-more-packets" if the client's decoder halts from time to time to wait for more packets to arrive. If too many receivers feed back network state information at the same time, then implosion could occur. To avoid implosion, the server first selects a sample set of receivers for the state feedback information, and then for each client selected, the client delays a random time interval before sending the state information to the server. In the application, once the server probes the current multicast session and obtains the current network state information, it adjusts the encoded video stream with one appropriate bit rate that fits most of the clients.

3. FEEDBACK DESIGN

In tracking global head motion and facial deformation, feedforward system design, i.e. extracting 3-D information from the intensity image directly using pure computer vision techniques, inherits a potential problem – depth information is lost during video capture. We propose a feedback system design that analyzes 3-D motion through synthesis. For this purpose, we develop a three-level signal representation that relates the input video to the output synthesized graphics. The three levels of the representation are the pixel-level intensity signal, the 2-D projection-plane shape description, and the 3-D graphics model vertex values in the canonical graphics model space (Figure 2). We use a finite set of typical human facial sequences as a training set for the human facial structures and obtain a codebook of 3-D facial structure parameters based on facial action coding system and facial motion synthesis system. For a new input facial sequence, we apply a texture map and choose a set of structure parameters that best describe the current facial structure and produce a close resemblance to the input video. This set of parameters is then used for coding.

3.1. System Analysis

The directly available information in a 3-D MBC input are the pixel intensities captured by video camera. Other available information is *a priori* information captured by the 3-D facial graphics model. In a feedforward system, video intensity change is assumed to come from the movement of the person in the foreground. Based on this assumption and properties of the intensity signal, we segment the foreground object using image processing techniques based on temporal motion and spatial edge information. From these intensity signals, we also extract the 2-D facial shape changes using high-level computer vision techniques. The 2-D shape changes reflect the projected 3-D motion.

For a 3-D graphics model, *a priori* knowledge about the human facial structure is available because the 3-D graphics model CANDIDE is based on a real person's face. Using computer animation techniques, we model 3-D facial movements. Then by applying the texture map, we obtain synthesized video.

3.2. System Design

In the training process, we collect human facial sequences representing several typical facial structures. Because we want to obtain facial structure deformation data that is as accurate as possible, we collect both front view and depth information of the person's facial structure. Then, we customize the 3-D model to the person by adjusting vertices in all three dimensions and obtain a texture map. We assume that only orthographic projection is employed in 3-D graphics rendering. To make the projected 3-D model animate in synchrony with the projected view of the face in the input video, we use a 2-D facial shape description as an intermediate layer to connect the image intensity signal and the 3-D graphics model. How do we obtain the initial 2-D shape templates? We manually identify the one-to-one mapping between the feature points on the 2-D shape contours and a subset of 3-D vertices on the 3-D graphics model (Figure 5(a)). In the subsequent video frames, we track the 2-D nonrigid shape changes using the deformable templates (Figure 5(d)). Next, we synthesize the 3-D graphics model in synchrony with the input video using 2-D deformable templates. However, there are more vertices on the 3-D graphics model than the number of 2-D facial shape feature points. How do we use these 2-D visual cues to generate the close-to-input 3-D synthesized video? We achieved this goal by designing a physically-based animation system.

A physically-based simulation system models the 3-D face as a flexible structure. We discretize the continuous flexible structure geometrically using a finite element method and simulate this temporarily through finite difference and iterative numerical techniques. The CANDIDE model is already a discretization of 3-D facial structure in the spatial domain. The continuous material masses are redistributed to the discrete points as lumped masses. We label the subset of 3-D model vertices that correspond to the 2-D shape feature points as control vertices. Then, to model the viscoelastic properties of facial tissue, we use biophysical units such as mass-spring-dampers to interconnect these control vertices and their neighboring vertices. When the 2-D shape templates deform in the input video, the control vertices on

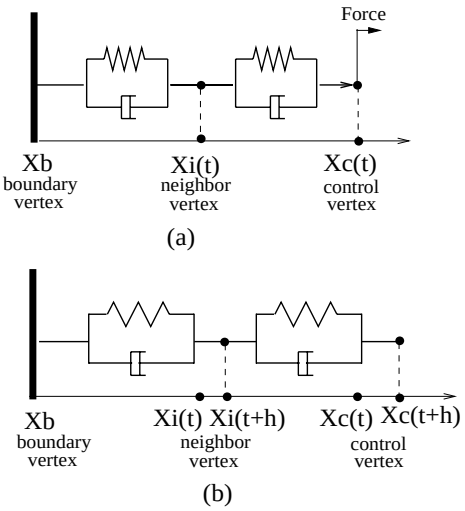

Figure 4. The nodal displacement of a spring-mass-damper system shown in one dimension: (a) the initial equilibrium state at time t: x_b is the boundary vertex; x_c is the control vertex; and x_i is the neighboring vertex of the control vertex. (b) the final equilibrium state at time $t+h$.

the 3-D model move accordingly. Based on physical laws, the neighboring vertices move also until the facial structure reaches a new equilibrium. The displacement is illustrated in a one-dimensional drawing of Figure 4 and the animation of 3-D graphics model is shown in Figure 5(e). For different persons in different training sequences, the same 2-D shape changes in the 2-D image plane cause small variations in the 3-D facial structure deformation. This effect is controlled by adjusting the spring-damper parameters and comparing the input video to the synthesized video (3-D structure plus texture map). Once we obtain the 3-D structure deformation parameters, we save them as a 3-D structure codebook for coding.

When a new video sequence becomes available, we extract the 2-D shape moving scales as visual cues. We use these visual cues to help retrieve the 3-D structure deformation information from the codebook. Then we apply the texture map to the 3-D structure to get the synthesized image (Figure 5(f)). The 3-D structure codeset that produces the closest texture resemblance to the input video is used for coding. This process is shown in Figure 3.

4. CODING FOR TRANSMISSION

Using 3-D MBC framework for visual communication, the video texture, 3-D graphics model parameters, structure motion data, and other synthesis parameters (including the biomechanical model parameters such as the spring-mass-damper coefficients, and time steps) need to be coded and transmitted.

For 3-D graphics model data, the highest transmission reliability should be assigned because if the receiver does not receive the customized 3-D graphics model information, decoding (rendering) will not be possible. Thus, TCP (or an equally reliable) protocol is used. The 3-D graphics model parameters include a list of the original model vertices and a list of the associated triangle topology information. Because the CANDIDE wireframe is relatively simple, we directly code the graphics model parameters. Although TCP is reliable, it is slow because of its three-way handshake. For other parameters, we use a UDP-type networking protocol to fulfill real-time transmission requirement in videoconferencing.

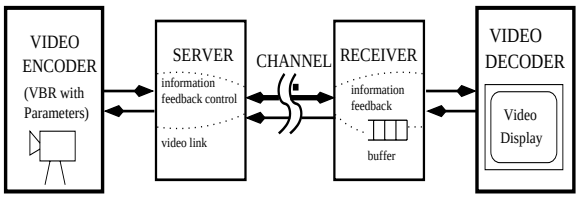

Figure 6. Desktop Video over IP System

Multiple factors impact decoded video quality. The UDP networking-related factors include packet loss, packet delay, and buffer overflow. In a 3-D MBC, losses due to buffer overflow are nontrivial because the time it takes to render an image can be significantly different for different CPUs. Thus, we develop an application content-level protocol that feeds the client receiving information back to the encoder to control the bit generation rate so that the receiver can continuously display the synthesized video (Figure 6). We mainly control the spatial scalability (number of vertices in the 3-D graphics model) rather than the temporal scalability (frame rate). The *Candide* model has 160 vertices. A second layer consisting of an additional set of vertices at the centroid of each triangle of the initial model can be added/dropped. Two sets of deformation synthesis parameters (animation scripts) – one for the original vertices, and the other for the additional vertices – would be placed into separate packets. This increases the robustness of the 3-D graphics rendering with respect to packet loss.

In coding the video texture, we choose a framework that causes less quality degradation for Internet transmission. That is, for a 3-D graphics model-based coder, because all of the subsequent frames are rendered on top of the first frame texture, we use a frame-based coder rather than a block-based coder to combat the blocking artifacts. For a frame-based wavelet decomposition of the facial texture, there are two types of code generation. One is to entropy encode and packetize each subband independently [6], and the other is to encode symbols across different subband using an interband tree-structured filterbank [7]. Due to packet loss in the Internet, we choose the latter approach. This is because if the most important low resolution subband in the wavelet decomposition is lost, we cannot reconstruct the basic image texture. But if we use an interband tree structure, we can put the tree branches into different packets to avoid coding an entire band of signals in the same packet.

5. CONCLUSION

In this work, we designed a feedback framework for visual communications. We have implemented both video coding and networking prototypes presented above. For video coding, our 3-D MBC system consists of the following components that support a feedback design:

- a 3-D computer graphics head-and-shoulder model and a three-level signal representation of the model;
- basic video object analysis using image processing and computer vision techniques;
- facial structure deformation synthesis using computer graphics and animation;
- facial motion analysis and coding through synthesis; and
- texture and structure deformation parameter coding.

Our contributions are a three-level signal model and a non-rigid facial motion synthesis system. The three-level signal model provides an interconnection of the video image at the encoder to the synthesized image at the decoder through 2-D shape feature points. This model is the foundation for

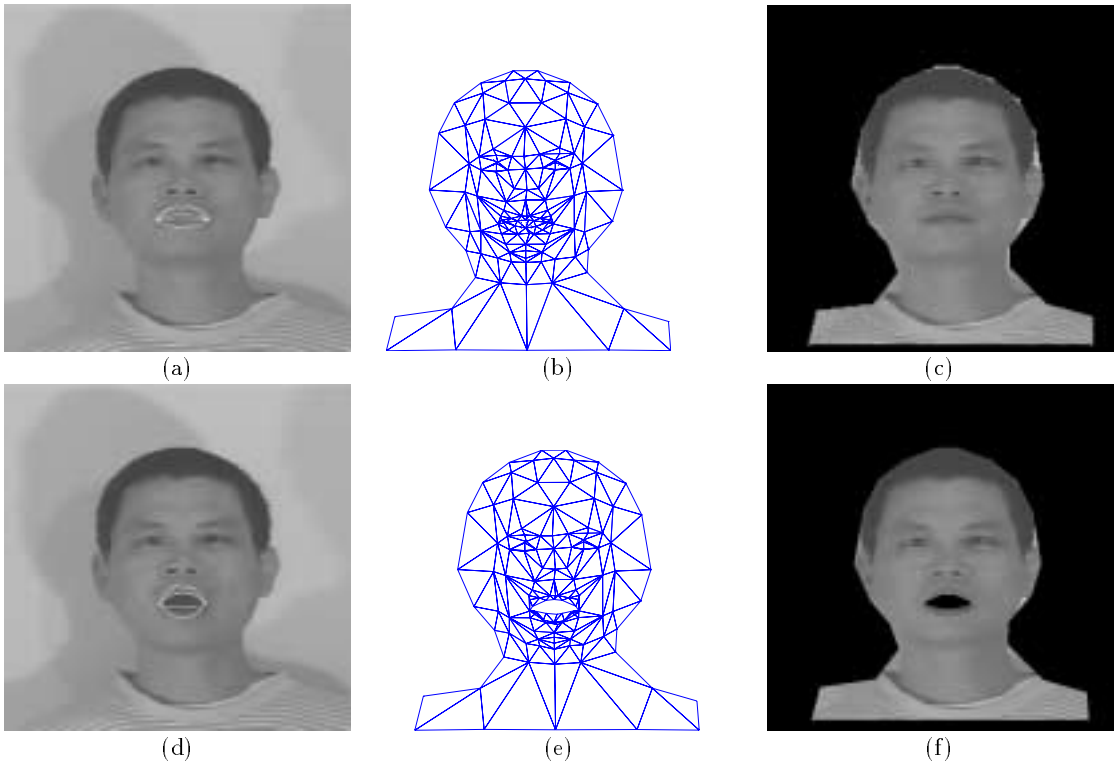

Figure 5. Nonrigid Facial Motion Tracking and Synthesis Using Deformable Template and 3-D Biomechanical Model: (a) placing mouth deformable template over frontal neutral face; (b) customizing CANDIDE model to fit to the person's neutral face; (c) texture mapping result; (d) tracking mouth movement using deformable template; (e) activating facial synthesis system to get 3-D geometry deformation; and (f) applying texture map to get the synthesis video image.

our 3-D analysis-by-synthesis feedback design, which better combats the ill-posed problem of 3-D motion estimation in the feedforward system because we have more control in facial animation. Through interaction in the training phase, we obtain a codebook of 3-D structure parameters for coding. Based on our finite element modeling, we need approximately 12.5 kbps to code the structure motion parameters for video displayed at 10 frames per second. The wavelet coding of QCIF size color texture (with neutral facial expression) requires less than 20 kpbs.

For video transmission over IP, to provide users at the receiving end with continuous video rather than a blank screen or jumping video, we feed back the network state and receiver rendering capability information to the server. During initialization, we use TCP to transmit the model vertices because if the receiver never receives the 3-D model information, it will never be able to synthesize any images. We use a frame-based tree-structured texture coding method that gives lower video quality degradation under the same packet loss rate. In order to deal with the possible packet loss resulting from the network congestion or receiver buffer overflow, we use 3-D models with different vertex resolutions for different levels of rendering accuracy. Our demonstration showed that with a limited number of receivers in one communication session, the networking feedback criteria works well. When in real Internet environment with potentially millions of users attending one session, how well the feedback system will work remains as an interesting topic for future investigation.

6. ACKNOWLEDGMENT

Thank to Dr. Tom Gardos and Mr. Sam Li for their help.

REFERENCES

[1] Franklin F. Kuo, Wolfgang Effelsberg, and J. J. Garcia-Luna-Aceves, *Multimedia Communications: Protocols and Applications*, Prentice Hall PTR, Upper Saddle River, NJ, 1998.

[2] M. Rydfalk, "Candide, a parameterized face," *Internal Report, Linkvping University*, Oct. 1987.

[3] A. Azarbayejani, B. Horowitz, and A. Pentland, "Recursive estimation of structure and motion using relative orientation constraints," in *Proc. Computer Vision and Pattern Recognition*, June 1993, pp. 294–299.

[4] D. Terzopoulos and K. Waters, "Analysis and synthesis of facial image sequences using physical and anatomical models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 569–579, June 1993.

[5] H. Li, P. Roivainen, and R. Forchheimer, "3-D motion estimation in model-based facial image coding," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 545–555, June 1993.

[6] Martin Vetterli, "Multidimentional subband coding: some theory and algorithms," *Signal Processing*, vol. 9, no. 2, pp. 97–112, Feb. 1984.

[7] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.