Texture synthesis for model-based coding

Young Wook Sohn, In Kwon Kim, and *Rae-Hong Park
(Dept. of EE, Sogang Univ., C. P. O. Box 1142, Seoul 100-611, Korea)

Abstract: Model-based coding is one of several approaches to very low bit rate image coding and it can be used in many applications such as image creation and virtual reality. However, its analysis and synthesis processes remain difficult, especially in the sense that the resulting synthesized image reveals some degradation in detailed facial components such as furrows around eyes and mouth. To solve the problem, a large number of methods have been proposed and the texture update method is one of them. In this paper, we investigate texture synthesis for model-based coding. In the update process of the proposed texture synthesis algorithm, texture information is stored in a memory and the decoder reuses it. With this method, the transmission bit rate for texture data can be reduced compared with the conventional method updating texture periodically.

Introduction

Recently, new approaches to low bit rate coding with improved image quality have been proposed, and knowledge-based coding is one of them. In contrast to waveform coding, knowledge-based coding analyzes an image in a structural way, so the large compression ratio can be obtained with its application areas naturally different from those of conventional waveform coding methods. Model-based coding is one of them and it describes the object motion in terms of the parameters of the assumed model. With this concept, semantic coding rules are specified and various applications such as very low bit rate videophone, image creation, and virtual conference are possible [1].

Many papers have shown considerable results in motion analysis of the input video sequences. But because model-based coding is still at its infancy, some topics remain unsolved, for example, initialization process, improving image quality, background processing, and so on. For updating texture information, a tradeoff between image quality and bit rate exists, on which the paper is focused.

In this paper, we observed some repeated actions of facial expressions and described a storage and indexing algorithm for texture information. Transmission schemes were also suggested and the expected bit rate and image quality were discussed.

Model-based coding

Model-based coding is a kind of knowledge-based coding which focuses on an efficient representation of a person's head. A common 3-D wireframe model of a person's head is shared on both the encoder and decoder.

For initialization, each characteristic point of the model is adjusted to an input facial image and the texture information of the first frame is transmitted. After initialization, the encoder analyzes the motion of an input facial image and encodes it according to the action rule of the shared

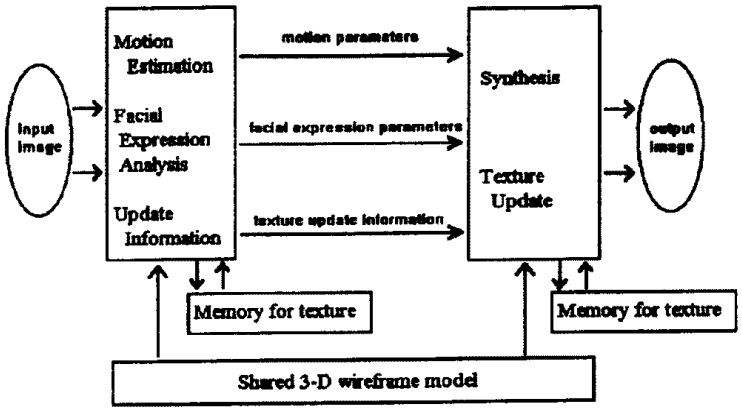


Fig. 1. System block diagram

Table 1. AU examples [4]

No.	AU Name
1	Inner Brow Raiser
2	Outer Brow Raiser
5	Upper Lid Raiser
10	Upper Lip Raiser
16	Lower Lip Depressor
20	Lip Stretcher
26	Jaw Drop
41	Lid Drop
43	Eyes Closed
45	Blink

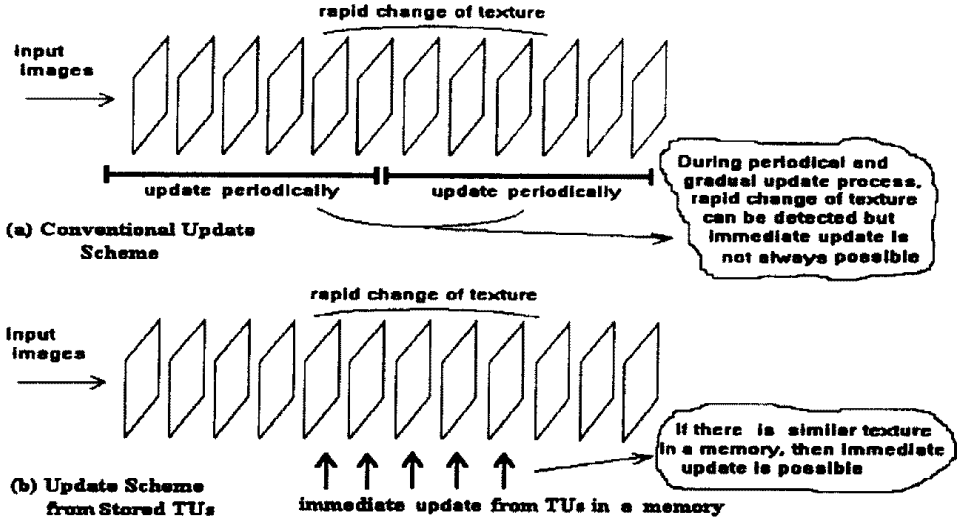


Fig. 2. Occurrence of rapid change of texture

model. One of the most common action rules of the shared model is the facial action coding system (FACS), which describes an expression of a human face with a set of basic actions of facial components. Most of model-based coding systems analyze and synthesize a person's facial image based on the FACS [2][3]. Some examples of action units (AUs) are listed in Table 1 [4].

In motion tracking, global and local motions are analyzed. Global motion includes head rotation and translation whereas local motion represents the change of facial expression. To find global or local motion, typical methods such as block matching algorithm, optical flow computation, or contour matching method can be used. Li *et al.* [2] and Choi *et al.* [4] developed direct methods to find global and local parameters directly from 2-D motion estimation based on the optical flow concept. Such algorithms are closely related to the shared wireframe model, in the sense that their analysis process specifies the parameters of the model.

The decoder synthesizes a facial image with transmitted parameters. Texture information is filled on the wireframe model using a texture mapping method. To improve the reconstruction quality, texture update is necessary. Some approaches to texture update such as transmitting the difference texture information, or transmitting texture base and parameters have been presented.

Although the texture update method guarantees high quality of the reconstructed image, the bit rate increases because a large number of pixels must be transmitted. The increase of the bit rate is proportional to the amount of updated pixels and the update period. The hybrid form of the discrete cosine transform/differential pulse code modulation (DCT/DPCM) can be used to reduce the bit rate.

Choi *et al.* showed a synthesis example of some furrows with previously transmitted texture information, which also reduced the bit rate.

Usage of previous texture information

We employed a memory system for storing various texture information in both the encoder and decoder and used it for synthesizing the facial image. The block diagram of the proposed system is shown in Fig. 1, in which the memory part for storing texture information is added to the conventional system [2][4]. The usage of the previous texture information requires synthesizing process such as assembling texture information of some facial parts and updating luminance.

Motion analysis of the proposed system is similar to that of the conventional 3-D model-based coding system. Both the encoder and decoder share the same memory for storing texture units (TUs). The system reuses the stored TUs as many times as possible. The proposed method provides the following advantages;

(1) As the previous texture information can be reused, the amount of bits for texture update can be reduced.

(2) If the rapid change of texture information exists, the conventional update method may miss such changes during the transmission period. The reusage of the previous texture information can reflect the change as fast as possible in the decoder part because similar texture information may be stored in a memory. This concept is

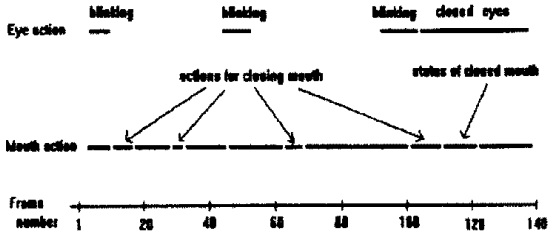


Fig. 3. Repeated actions in the 'Miss America' sequence

Table 2. Examples of expected TUs

TU number	contents
TU#1	blink left eye
TU#2	blink right eye
TU#3	open mouth
TU#4	close mouth
TU#5	speech of '[a]'
TU#6	speech of '[e]'
TU#7	speech of '[i]'
TU#8	speech of '[o]'
TU#9	speech of '[u]'

shown in Fig. 2.

In our system, we observed that most of actions in facial images are concerned with eyes and
mouth, and statistical knowledge for some frequent actions are used. Also we tried to increase the hit ratio of stored TUs, where the hit ratio is defined by

$$hit\ ratio = \frac{number\ of\ texture\ reusages}{number\ of\ update\ processes}.$$

Fig. 3 shows how typical actions appear in the 'Miss America' sequence. For eyes, the most frequent action is blinking process. Mouth has more various shapes than eyes and has some repeated actions such as closing. This means that if we can find general frequent actions for facial images, the previous texture information can be used efficiently in synthesizing the facial image.

For selecting TUs, two approaches are possible. The first method stores the most probable actions described above. The second method counts the hit number of each TU in a memory and replace the least frequent TU by the new one. Some examples for TUs are shown in Table 2, where the most concerned TUs are about eyes and mouth.

In the initial processes, pixelwise different texture information derived from several frames is sent to the decoder. For the next facial image, the encoder need not transmit all texture information if there are similar TUs in a memory. Otherwise, the encoder sends pixelwise texture information and the decoder appends it to the memory. This process is based on the assumption that the luminance of the texture does not change. So when the luminance changes, the decoder should update the luminance of texture. The analysis of texture luminance is described later.

Storage and indexing of TUs in a memory

As mentioned earlier, the TU is defined as

$$TU_n = T(AU_{n_1} + AU_{n_2} + \ldots + AU_{n_m})$$

where TU_n denotes the nth TU consisting of m AUs and $T(\cdot)$ represents texture information.

The number of AUs related to a TU should be small because a large number of combination can cause incorrect selection of the TU number and intensity.

We presumed that ten intensity values for each TU are enough and they are calculated from the intensities of the related AUs. In our work, we set the TU intensity to the mean value of related AU intensities. The TU intensity is then calculated as

$$intensity\ of\ TU_n = \frac{c}{m_n}(A_1 + A_2 + \ldots + A_{m_n})$$

where A_i denotes the ith AU intensity. m_n and c represent the number of the AUs related to the TU_n and a constant, respectively.

When the related AUs appear, we store its texture information in a memory. This leads to the structure as shown in Fig. 4. In real implementation, above scheme results in some empty TU memory as shown in Fig. 5. This is due to the rapid change of texture before the texture information is stored in the memory. If empty memory content is selected later, we use the nearest non-empty texture information. In restoring TUs from a memory, we find TUs only when there are related AUs that have been transmitted from the encoder. If AUs related to a certain TU exist, then we calculate the intensity of the TU from the intensities of the related AUs. Otherwise, we synthesize and update the texture information using the conventional method.

Transmission of texture information

As mentioned before, we assumed that the most frequently repeated action is eye blinking and mouth closing. Transmission schemes of texture information are as follows;

(a) Full texture information of the first frame of the sequence is transmitted to the decoder.

(b) For the predefined TU_n content based on AUs, we store and transmit the texture information with m_n intensity values.

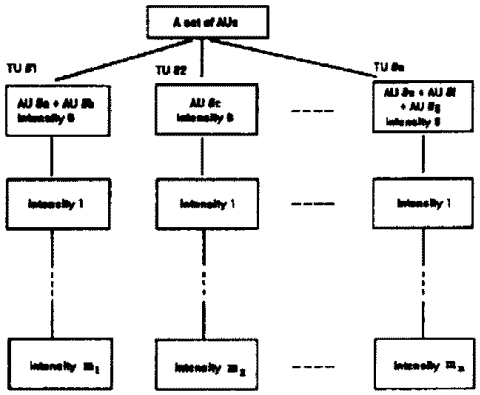


Fig. 4. Structure of TUs from input AUs

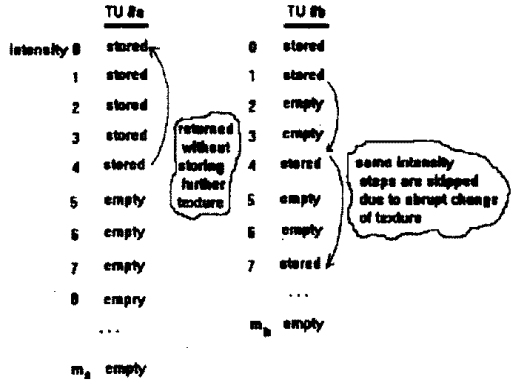


Fig. 5. Empty TUs

(c) After step (b), we transmit only AU numbers, AU intensities, and luminance parameters if no change of global motion occurs. To analyze the change of luminance, we assumed a luminance model as

$$I' = aI + b \tag{1}$$

where I' and I represent the desired and current luminance values, respectively. To find optimum constant values of a and b, an squared error function defined by

$$error = \sum[\, I_o(x,y) - (a I_m(x,y) + b)\,]^2 \tag{2}$$

is minimized, where I_o represents the intensity of an original input image and I_m denotes the stored image in a memory. Taking partial derivatives and setting the result to zero, we get [5]

$$a_{opt} = \frac{Cov_{I_o I_m}}{\sigma_m^2} \tag{3}$$

$$b_{opt} = \mu_{I_o} - a_{opt} \cdot \mu_{I_m} \tag{4}$$

where $Cov_{I_o I_m}$ represents the covariance of I_o and I_m, and σ_m^2 denotes the variance of the intensity of the previous texture. μ_o and μ_m signify the mean intensity values of the current and previous texture, respectively.

If the encoder knows the corresponding TU information, then only the transmission of a_{opt} and b_{opt} is needed to update the luminance of texture.

(d) If there is no suitable TU stored in the decoder memory, for example, new action of mouth occurs, we transmit the texture information and append it to the decoder memory.

Texture Synthesis

Final synthesis processes are categorized as follows;

(a) Only indexed texture + luminance change
(b) Only updated texture + luminance change
(c) Indexed and updated texture + luminance change

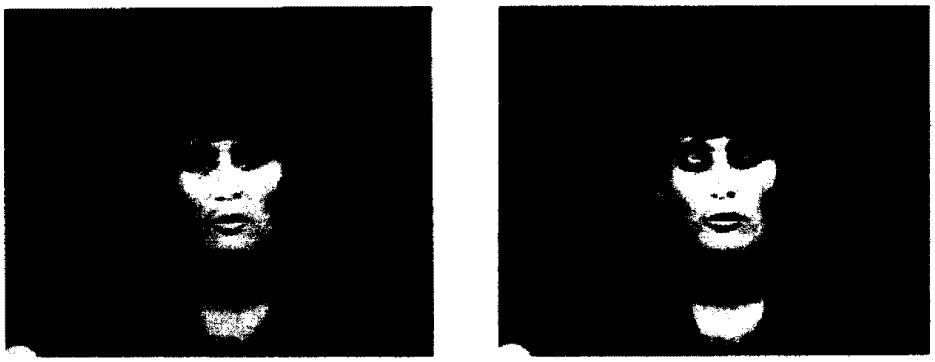
Most of synthesis processes will have the form of (c).

Results and Discussions

We used the base texture information in Fig. 6(a) as the first frame. After some extractions of eyes and mouth in other frames, texture information is combined with stored TUs.

Figs. 7 and 8 show synthesized images without and with luminance update, respectively, when parameters of another input image (Fig. 6(b)) are transmitted. Fig. 7(a) shows the synthesis result, in which luminance update is not assumed and only indices of TUs are used. Fig. 7(b) shows the difference between the original image and synthesis result, where the difference value is multiplied by ten for better comparison. On the other hand, with the luminance optimization, Fig. 8(a) shows the synthesis result. Fig. 8(b) shows the difference between the original image and the synthesis result, which shows improved quality compared with Fig. 7(b). The peak to signal to noise ratio (PSNR) values of Figs. 7(b) and 8(b) are 35.5 dB and 36.8 dB, respectively.

But in our experiments, some luminance optimization failure occurred, due to the inaccurate extraction of texture in the AU analysis. There are some limits on description of a person's face with AUs, resulting from the inaccurate



(a) First frame (b) Input frame

Fig. 6. 'Miss America' test sequence

estimation of motion and some incorrectly determined AUs. Not only the motion of AUs but also the related change of tissues on a face change the texture information. Although the 3-D model contains information for such relations, just analyzing AUs and extracting texture are not sufficient to describe a person's face precisely [6][7]. Also if the update period is too long, luminance optimization also fails. Because the luminance optimization is not a perfect solution with respect to the change of global motion, full updated texture information is needed when global motion such as rotation occurs.

In calculating the bit rate, Choi *et al.* assumed six global motion parameters (three position and three angular velocities) and ten AUs, because ten AUs are sufficient to represent a facial expression [4]. For each AU, six bits and five bits are allocated to represent the AU number and AU intensity, respectively. If we set the frame rate to 10 Hz, the bit rate required for encoding motion parameters is [4]

(6 global parameters * 5 bits/parameter + 10 AUs * (6 + 5) bits/AU)/frame * 10 frames/s = 1.4 Kbits/s.

Choi *et al.* set the update interval to 5 ~ 10 times during 3 minutes. With their settings, the bit rate of their method is about 3.5 ~ 10.5 Kbits/s [4].

In our system, if we set the update period to 18 seconds (this value corresponds to 10 times during 3 minutes) and the wlireframe model is composed of about 400 triangle patches, then the mean bit rate for the worst case (no TUs are selected and the total updated information is transmitted) is

1.4 Kbits/s + ((2 luminance parameters / triangle patch * 400 triangle patches) * 5 bits / luminance parameter + (10,000 pixels * 5 bits / pixel)) / 18 s ≒ 4.4 Kbits/s

where we presumed that the number of pixels to update full triangle patches is about 10,000 and about 400 triangle patches are used in the wireframe model.

For the best case when TUs are used and no update information need to be transmitted, then the bit rate of

1.4 Kbps + ((2 luminance parameters / triangle patch * 400 triangle patches) * 5 bits / luminance parameter) / 18 s ≒ 1.62 Kbits/s

is required.

Thus the range of the bit rate is from 1.62 Kbps to 4.4 Kbps when the update period is 18 seconds.

For more rapid change of texture, the update period must be shorter. But the decrease of the update period results in the dramatic increase of the bit rate. For example, if we set the update period to 3 seconds, the bit rate for the worst case is

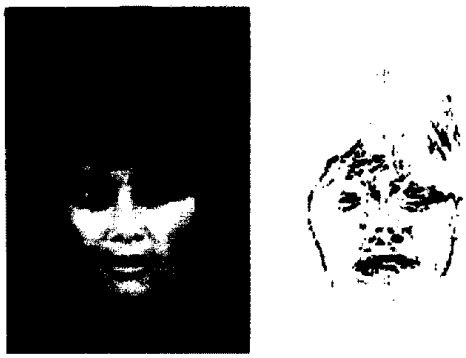
1.4 Kbits/s + ((2 luminance parameters / triangle patch * 400 triangle patches) * 5 bits / luminance parameter) + (10,000 pixels * 5 bits / pixel)) / 3 s ≒ 19.4 Kbits/s.

If only the stored TUs are required, which is the best case, then the bit rate of

1.4 Kbits/s + ((2 luminance parameters / triangle patch * 400 triangle patches) * 5 bits / luminance parameter) / 3 s ≒ 2.7 Kbps

is required.

The occurrence of the best case is not absolute, and the bit rate will range from 2.7 Kbps to 19.4 Kbits/s for the update period of 3 seconds. The average bit rate for a long sequence



(a) Result (b) Difference image

Fig. 7. Synthesized image without luminance update

(a) Result (b) Difference image

Fig. 8. Synthesized image with luminance update

depends on the content of TUs in a memory. Not only statistical knowledge of TU contents, but also the relations of TUs and AUs influence image quality and bit rate because some AUs have strong correlation with each other [4]. As mentioned earlier, we limited the maximum number of AUs related to the specific TU content to two.

Conclusions

We mainly discussed the texture update scheme and showed the reduction of the expected bit rate. Its value varies according to the TU content, and selecting the content is one of the important problems. We experimented with some basic actions such as eye blinking, mouth closing, and mouth opening. But in some sequences whose texture information was not stored in a memory previously, the full texture update information must be transmitted. The reconstructed image quality was improved with luminance update. Although we investigated this system for a real-time communication system, we believe that it can be used for a play-back system because it could find the most frequent action in the whole sequences. Further research will focus on development of algorithms for selecting TU content and for relating it with AUs.

References

[1] K. Aizawa and T. S. Huang, "Model-based image coding: Advanced techniques for very low bit-rate applications," *Proc. IEEE*, vol. 83, no. 2, pp. 259-271, Feb. 1995.

[2] H. Li, P. Rovinvainen, and R. Forchheimer, "3-D motion estimation in model-based facial image coding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-15, no. 6, pp. 545-555, June 1993.

[3] K. Aizawa, "Model-based image coding," in *Proc. SPIE Visual Comm. and Image Processing*, vol. 2308, pp. 1035-1049, Chicago, IL, Sep. 1994.

[4] C. S. Choi, K. Aizawa, H. Harashima, and T. Takebe, "Analysis and synthesis of facial image sequences in model-based image coding," *IEEE Trans. Circuits and Syst. for Video Technol.*, vol. CSVT-4, no. 3, pp. 257-275, June 1994.

[5] M. Wollborn, "Prototype prediction for colour update in object-based analysis-synthesis coding," *IEEE Trans. Circuits and Syst. for Video Technol.*, vol. CSVT-4, no. 3, pp. 236-245, June 1994.

[6] I. Essa and A. Pentland, "A vision system for observing and extracting facial action parameters," in *Proc. IEEE Conf. Comput. Vision and Patt. Recog.*, pp. 76-83, Seattle, WA, June 1994.

[7] D. Terzopoulos and K. Waters, "Analysis and synthesis of facial image sequences using physical and anatomical models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-15, no. 6, pp. 569-579, June 1993.