A PSEUDO OBJECT-ORIENTED VERY LOW BIT-RATE VIDEO CODING SYSTEM WITH CACHE VQ FOR DETAIL COMPENSATION

Chung-Wei Ku, Liang-Gee Chen, You-Ming Chiu, and Yung-Pin Lee

DSP/IC Design Lab., Department of Electrical Engineering,
National Taiwan University, Taipei, Taiwan, R.O.C.
email: {william,lgchen}@video.ee.ntu.edu.tw
URL: http://video.ee.ntu.edu.tw/~william

ABSTRACT

In this paper, a pseudo object-oriented video coding system is proposed and implemented. In order to increase the coding efficiency, cache VQ algorithm is suggested to further compress those areas where motion estimation fails. According to our primary simulation results, the visual quality of long-timed sequences is still acceptable even for bit-rates below 10 Kbps. In addition to the high compression ratio for very low bit-rates, content-based applications are also expected since the proposed system utilizes segmented motion field; furthermore, the occurrences of prediction errors generally locate at emotionally important parts, e.g. eyes and mouth, etc. All the coded components are not only useful for compression but also meaningful for video recognition.

1. INTRODUCTION

As early as May 1991 the Moving Picture Expert Group (MPEG) raised the issue of audio-visual standard targeted at the bit-rate of 4.8-64 Kbps. These efforts was approved in July 1993 with the MPEG-4 nickname and the title "Very Low Bit-Rate Coding of Moving Pictures and Associated Audio". Basically larger compression ratio is expected in order to meet the modern modem standard V.34 in which the bandwidth is defined as 28.8 Kbps. Currently the standard about visual telephone is covered by ITU-T H.324, where the video coding is defined in H.263. Besides, the goal of MPEG-4 moves to the issues about content-based applications or functionalities. As a result, variable kinds of approaches are all developing very fast. For example, the model-based approaches [1] or analysis-synthesis approaches [2]. In this paper, a pseudo object-oriented very low bit-rate video coding system is proposed. The goal of the proposed system is to combine GSTN communication, personal computers, and audio-visual multimedia service. Compression of video for very low bit-rates is expected; in addition, content-based functionalities are another possible applications.

2. MOTION ESTIMATION AND CODING OF THE MOTION FIELD

2.1. Optical Flow based Motion Estimation

Similar to most video coding system, the proposed system removes the temporal redundancy in the video sequences via motion estimation. In order to code the motion vectors more efficiently, we would like to generate a motion field with less spatial variation and good prediction. Besides, the generated motion vectors had better be meanful about the real movements. For the above reason, we propose a modified cost function which is shown as follows:

$$\varepsilon = \int\int \; ((E_x u + E_y v - E_t)^2 + \alpha(u_x^2 + u_y^2 + v_x^2 + v_y^2) + \beta(u_{xy}^2 + v_{xy}^2))dxdy.$$

(u, v) is the motion vector and E_x is the partial deviation of the spatial-temporal function of a pixel along the x axis. Similar explanation is for E_y. The additional third term indicates the penalty on convoluted contours. Another problem is that the answers may be trapped in local minimum due to gradient descent approach. To reduce the probability of this situation, a pyramid approach is suggested. The higher level of the pyramid is composed of the subsampled pixels at the lower level. Motion estimation is executed from the top level to the bottom level. The scale of adjustment at higher level is larger than that of lower ones to achieve a coarse answer; in addition, more iterations are executed at lower level to obtain the precise vector. As a result, most local minima are avoided and the possibility of finding global minimum is increased. In addition, the post processing of edges and stationary regions are

0-7803-3258-X/96/$5.00 © 1996 IEEE

also suggested. All the above is named modified optical flow algorithm (MOFA) [6] and an example is given in Figure 1.

2.2. Arbitrarily Shaped Transform of the Segmented Motion Field

In order to encode these motion vectors efficiently, the spatial redundancy in the motion field is exploited. The motion field is subsampled then segmented into several homogeneous regions. After that, arbitrarily shaped transform (AST) is applied to these regions. Since the shape information is necessary for AST and chain code description costs much data amount, a rectangle approximation is suggested. According to the simulation results, the saving of data amount is obvious while the loss of performance is negligible [7]. The segmented and approximated motion field is then applied with AST in the subsampled pixel domain. Polynomial based kernels are selected in our system. In order to speed up the processing time, two heuristic rules are assumed to define the coefficient selection function (CSF). In our system, $f(x) = \sqrt{x}$ is selected as the CSF. Finally, for each region, two sets of the AST coefficients and one set of the knot points describing the shape are transmitted. Before being sent to channel, all the coded elements are variable length coded (VLC) by Huffman code to remove the statistical redundancy.

3. DETAIL COMPENSATION OF PREDICTION ERRORS

According to our experiment results, we found most of the errors occur in the position of eyes and mouth. The reason is that some detailed movement is not well compensated by the motion field; such as the open or close of eyes and mouth. However, the representation of fine emotion is especially important for conversation applications. In order to encode these part more efficiently and avoid the straightforward approach such as refresh frame, the idea of "motion off objects" is proposed. That is, for those not well compensated regions, which are generally eyes or mouth, cache vector quantization is applied instead of original motion compensated method.

3.1. Localization of Motion Off Objects

The first step is to find the appropriate position of those "motion off objects" (MFO's). In our system, the procedures of determining MFO is as follows:

1. A sliding window with size 8 by 8 is applied to the error frame in a way similar to two dimensional filter. The mean absolute error (MAE) of each sliding window is calculated.

2. If the MAE is too larger, in addition, 80% of the pixel errors in this windows are large enough, all the pixels in the window are marked as MFO regions.

As the sliding window operates to all the error frame, the distribution of MFO's are found. For example, according to the reconstructed error in Figure 2 (a), the locations of MFO are the white area in Figure 2 (b). We found the above scheme detects most of the motion off areas such as eyes and mouth. In fact, the proposed method to locate MFO is very similar to the morphological filter approach [4] or median filter approach [5].

3.2. Cache Vector Quantization Scheme

In order to encode MFO more efficiently, the temporal redundancy of the MFO's are exploited. That is, since most of the MFO's are about eyes and mouth of the same person, generally the image content about these parts are strongly correlated as the time passed. If some typical patterns about the eyes and mouth can be detected and memorized, the coding of later MFO's can be solved by sending the indices of the patterns rather than content of the image [3]. Basically, the above idea is an extension of cache vector quantization proposed in [8]. However, several problems still exist. The first problem is how to memorize these patterns for later pattern matching. Therefore, the idea of *image pattern prototype* (IPP) is adopted in our system. That is, several blocks with sizes 24 by 16 are extended from the MFO's. The idea is shown in Figure 3. The growing of the boundary is from the center of the MFO. For example, an IPP including the mouth area is displayed in Figure 4.

The second problem is the way to memorize appropriate patterns and apply the matching scheme. In our system, all the IPP's are saved in a cache-like codebook. That is, whenever the IPP of current frame is generated, they are compared with the IPP's in codebook. If the winner in the codebook is very similar to the input IPP, index of the winner IPP is transmitted; otherwise the input IPP is encoded by DCT similar to intra mode coding and this IPP is put into the cache codebook. The update strategy we used for cache replenishment is least-recently-used (LRU) because its performance is the best [8]. However, to find the best matched IPP from the whole codebook is time consuming and unreasonable because it is meaningless to compared a "mouth" IPP with an "eye" IPP. Therefore, the content in the codebook should be classified

in a reasonable sort. Observing the sequence about IPP, it is easy to find that the positions of each kind of the IPP's are strongly correlated because the position of mouth or eyes changes quite little in the sequences. According to the above idea, the position of the IPP is selected as the cache tag field in our system instead of some classification methods [3]. In other words, the position of the input IPP is compared with all the positions of the IPP's in codebook; only those IPP's near the input will become the candidate IPP's for later matching process. The matching criteria we used is mean absolute error (MAE).

Although most of the MFO's are well enclosed in the IPP's, the exact position of MFO must be adjusted to match the input MFO instead of input IPP. For this purpose, the motion vectors of IPP are also adopted [3]. That is, the best matched IPP is the one whose MAE is the minimum with the IPP's best motion vector. Full search within a pre-defined range similar to block matching motion estimation is utilized to find the best motion vector for each IPP candidate. Another important issue is the position tag of each IPP in codebook must be updated if it is the "hit" one. In order to keep the correlation of locations with the input IPP, the position of the "hit" IPP is updated toward the input IPP with the displacement of the motion vector. In brief, the proposed cache VQ for MFO coding is: if the input IPP really "hits" one of the codeword IPP, index of the codeword and a motion vector related to this codeword are transmitted for data compression; otherwise the input is DCT coded in a way similar to intra mode coding.

4. SIMULATION RESULTS

The idea of cache VQ for MFO coding is verified by the simulation results. In order to determine the size of the IPP codebook, the simulation for several kinds of codebook is executed and the results for a long-timed sequence "*CMJ*" which is generated in our Lab for simulation are listed in Table 1. According to the simulation results, as the size of the codebook increases, the performance is improved because most of the typical IPP's can be found in the codebook gradually; however, as the size of the codebook is larger than 128, the performance can not be improved any more. Besides, the performance degrades a little because some "hit" IPP's are not good enough therefore the number of MFO's might increase in later sequences due to error propagation. In terms of bit-rate, the average bit-rate decreases as the size of codebook increases because lots of IPP's can be found in the cache codebook and index of "hit" IPP uses less data amount compared with DCT coding; however, the index costs more data as the size of codebook increases therefore the average bit-rate increases finally. According to the above phenomenon, the size of IPP codebook is selected as 128 in our system.

Since the IPP codebook is empty at the beginning, the hit ratio is not very high originally. However, as time passed, the cache is filled with those typical IPP's and the efficiency increases. In fact, the proposed system is simulated for several head and shoulders sequences. Some primary simulation results give 34 dB PSNR even at lower than 10 Kbps bit-rate for long-timed sequences. For short-timed sequences, since the utilization of cache codebook is not high enough, the average bit-rate will be between 10-20 Kbps with acceptable quality. Since the IPP's generally locates at the important parts of human features, more efforts are spent on investigating the application about content analysis currently. An example of IPP codebook content after coding several frames for sequence "*Miss America*" is shown in Figure 5.

5. CONCLUSION

A pseudo object-oriented very low bit-rate video coding system is proposed in this paper. The proposed system uses modified optical flow based motion estimation algorithm to remove the temporal redundancy; the spatial redundancy in the motion field are removed by the arbitrarily shaped transform and rectangle approximation. Finally, for those parts which are not well predicted by motion compensation, cache VQ is suggested to achieve further compression. Currently we are studying the advanced coding of MFO's and how to use the coded information for video recognition. From our simulation results, we believe the proposed method is potential for very low bit-rate video coding or even future content-based applications.

6. REFERENCES

[1] K. Aizawa, H. Harashima and T. Saito, "Model-based analysis-synthesis image coding (MBASIC) system for a person's face", *Signal Processing: Image Communication*, vol. 1, pp. 139-152, 1989.

[2] J. Ostermann, "Object-based analysis-synthesis oding based on the source model of moving rigid 3D objects", *Signal Processing: Image Communication*, vol. 6, pp. 143-161, 1994.

[3] M. Wollborn, "Prototype prediction for color update in object-based analysis-synthesis coding",

IEEE Trans. Circuit and System for Video Technology, vol. 4, pp. 236-245, Jun. 1994.

[4] W. Li, V. Bhaskaran, and M. Kung, "Very low bit-rate video coding with DFD segmentation", *Signal Processing: Image Communication*, vol. 7, no. 4-6, pp. 419-434, Nov. 1995.

[5] D. Qian, "A motion compensated subband coder for very low bit-rates", *Signal Processing: Image Communication*, vol. 7, no. 4-6, pp. 397-418, Nov. 1995.

[6] C.-W. Ku, Y.-M. Chiu, L.-G. Chen, and Y.-P. Lee, "Building a pseudo object-oriented very low bit-rate video coding system from a modified optical flow motion estimation algorithm", *to appear in ICASSP'96.*

[7] C.-W. Ku, Y.-M. Chiu, L.-G. Chen, and Y.-P. Lee, "The arbitrarily shaped transform of segmented motion field for a pseudo object-oriented very low bit-rate video coding system", *to appear in ISCAS'96.*

[8] C.-W. Ku, L.-G. Chen, and T.-D. Chiueh, "Cache vector quantisation algorithm in video compression", *IEE Electronic Letters*, vol. 29, No. 16, pp. 1423-1424, Aug. 1993.

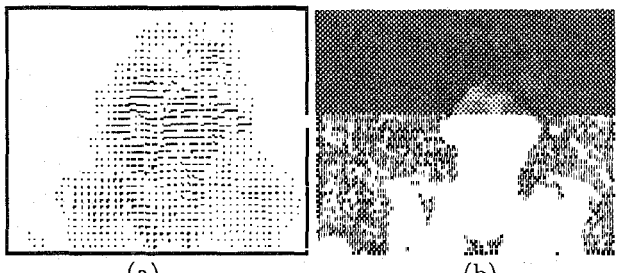

Figure 1: Results of MOFA for "Miss American": (a) motion field, (b) reconstructed frame.

Table 1: Results for different codebook sizes.

Size	PSNR	Bit-rate	Hit Ratio
32	33.332 dB	9.90 Kbps	69.9%
64	33.348 dB	8.33 Kbps	82.8%
128	33.382 dB	7.71 Kbps	85.7%
256	33.341 dB	7.78 Kbps	84.9%
512	33.341 dB	9.40 Kbps	85.0%

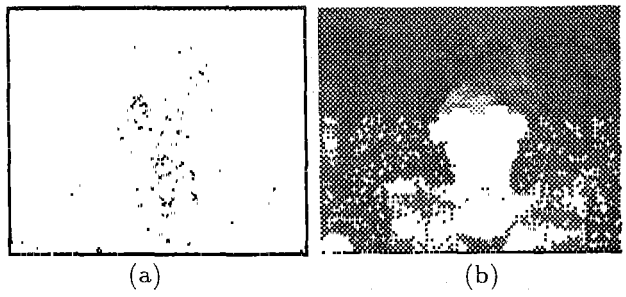

Figure 2: Localization of MFO: (a) reconstructed error, (b) locations of MFO.

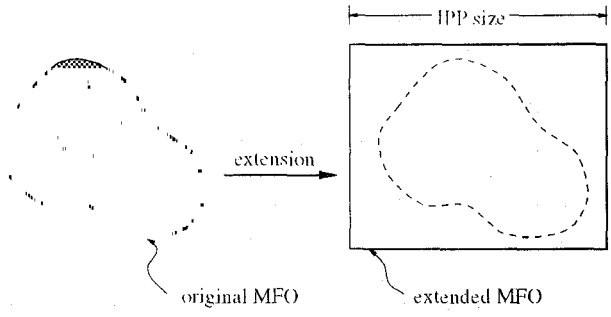

Figure 3: Extend MFO into image pattern prototype.

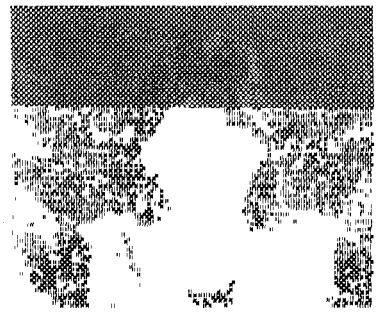

Figure 4: Location of the IPP including the mouth.

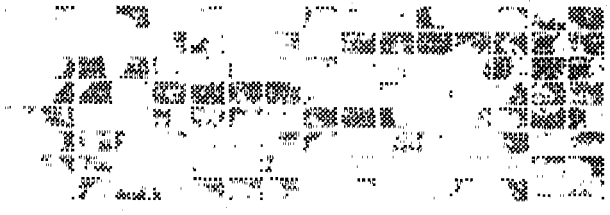

Figure 5: Content of the IPP codebook for "Miss America".