WAM 12.2

An Efficient Streaming and Decoding Architecture for Stored FGS Video

Yi-Shin Tung, Ja-Ling Wu, Senior Member IEEE, Po-Kang Hsiao, and Kai-Li Huang

Communication and Multimedia Laboratory,
Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan

ABSTRACT

FGS (Fine Granularity Scalability) is the latest video-coding tool provided in the amendment 4 of MPEG-4 standard. In this paper, we address a corresponding pair of efficient streaming schedule and pipeline decoding architecture to speed up the decoding process. The design may be applied to the applications of streaming stored FGS video and will benefit other FGS related applications.

INTRODUCTION

FGS [8~10] (Fine Granularity Scalability) is the latest scalable video codec provided in the amendment 4 of MPEG-4 standard [1,2,4]. By bitplane coding [6,7] of DCT residues, it can support finest data cut-off in the enhancement layer. Besides, it also supports two extra tools, frequency weighting and selective enhancement, which enlarge the possibility of coded picture quality enhancement of the encoding process. FGS seems to be a suitable technique for providing video on-demand service [3] on a QoS quaranteed network, such as xDSL. A modern settop box usually provides VOD service while Internet access or phone service is also activated at the same time (c.f. Fig.1). That means bandwidth is shared and the available quantity for VOD transmission is varying with time. FGS has the best ability to make use of remainder bitrate (c.f. Fig.2) [5] to improve video quality at any instance. However, several problems have to be solved in its implementation. First, the speed of the bitplane decoding is limited, because repeatedly accessing of frame buffer is required. Second, large amount of enhancement data of referable frames (like I or P frames) may enlarge the size of the decoding buffer. Third, additional frame buffer duplication is also introduced for maintaining base-layer prediction and decoding enhancement video simultaneously. All of these will limit FGS's applications and increase the coressponding hardware cost.

THE PROPOSED DECODING PIPELINE

For VOD services, most of the videos can be processed and stored in advanced. And a short delay of several frames may be tolerable. Thus, we address a corresponding pair of efficient streaming schedule and pipeline decoding architecture in this paper. The design may be applied when streaming stored FGS video in face of the aforementioned problems.

First, to speed up the bitplane decoding, a multiple-way decoding algorithm is suggested to start enhancement data decoding after queuing the receiving enhancement data of one frame (c.f. Fig.3). By this way, multiple frame buffer accesses can be eliminated. It helps to reduce memory traffic a lot, while only one-frame delay comes along. Next, we suggested an efficient streaming schedule by delivering the demanded of enhancement data in displayed order, such that the decoding buffer needed for storing compressed enhancement data can be reduced. Finally, we modified the decoding architecture by introducing the delay of several frames period. In the refinement schedule, the enhancement-layer decoding will reuse the available frame buffers in the base-layer decoding. Thus, the modified architecture will eliminate the behavior of frame buffer duplication. Thus, the traffic of cache processes is reduced. Combining the proposed techniques, it requires only three frame buffers and two frames of decoding delay when no backward prediction frame existed in the base-layer streams (c.f. Fig.4). Otherwise, (3 + #B) frame buffers and (2 + #B) frame of delay are required (c.f. Fig.5), where #B denotes the maximum number of consecutive B frames between neighboring referable frames.

CONCLUSION

Our work shows a feasible study for the people who are interested in FGS related applications. For hardware designers, the proposed pipeline architecture (c.f. Fig.6) may provide a suitable reference for chip design; for software programmers, how to achieve complexity balance at any time instance will be the major issue; and for streaming providers, traffic balance or multicast may be their focuses.

REFERENCES

[1] ISO/IEC IS 14496, Information Technology – Generic Coding of Audio-Visual Objects – Part 1: System, Part 2: Visual, Part 3: Audio, Part 6: DMIF, International Organization for Standardization, 1998.

[2] ISO/IEC 14496 AM 1, Part 1: Systems, Part 2: Visual, Part 3: Audio, Part 6: DMIF, International Organization for Standardization, 2000.

[3] ISO/IEC 14496-2:1999/FPDAM 4, Amendment 4: Streaming Video Profile, N3518, July 2000.

[4] MPEG Requirements Group, "MPEG-4 Applications Document", Document ISO/IEC JTC1/SC29/WG11/N2724, Seoul MPEG meeting, March 1999.

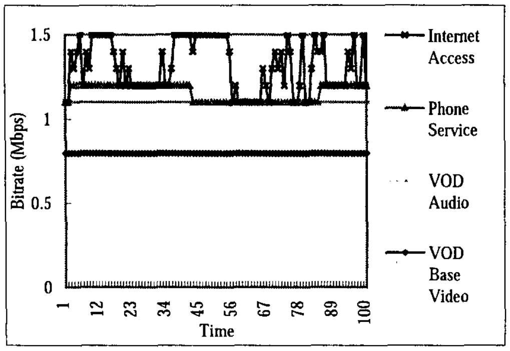

Figure 1: Examples of Traffic Bandwidth Sharing Among Several Applications: There are three services activated simultaneously over the 1.5Mbps ADSL line. They are: (a) VOD (b) Phone Service, and (c) Internet Access. The remainder of bitrate is changing with time instance.

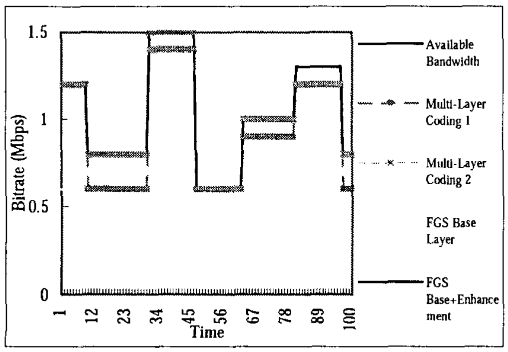

Figure 2: Bitrate Adaption for Multiple Layer Coding and FGS: Two configurations of multiple layer coding are assumed: (1) 600Kbps for the base stream and each additional stream of enhancement layer occupies 300Kbps. (2) 600Kbps for the base stream and each additional stream of enhancement layer occupies 200Kbps. FGS base layer is also set to 600 Kbps. Obviously,

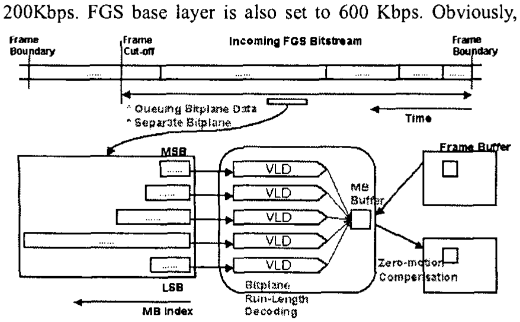

Figure 3: The Flowchart of Multiple-Way Bitplane Decoding: To speed up the bitplane decoding process, incoming FGS enhancement data are accumulated first and bitplane start-codes are found to separate data for each bitpalne VLD. Synchronization is achieved on MB basis. Zero-motion compensation is then used to sum up the resultant values.

[5] Girod, B., Farber, N., and Horn, U., "Scalable Codec Architectures for Internet Video-on-Demand", Telecom. Lab., Univ. of Erlangen-Nuremberg, Germany, 31th Asilomar Conference on Signals, Systems & Computers, 1998.

[6] Document ISO/IEC JTC1/SC29/WG11/M3989, "Bit-plane coding of DCT coefficients for FGS", Weiping Li, Optivision, Oct. 1998.

[7] Document ISO/IEC JTC1/SC29/WG11/M4204, "FGS Using Bit-Plane Coding of DCT Coefficients", Optivision, December 1998.

[8] Weiping Li, "Scalable Video Coding with Fine Granularity Scalability", ICCE 1999.

[9] Document ISO/IEC JTC1/SC29/WG11/M5122, "Results on Fine Granularity Scalability", Univ. of New South Wales, Oct. 1999.

[10] Weiping Li, Fan Ling, and Xuemin Chen, "Fine Granularity Scalability in MPEG-4 for Streaming Video", ISCAS 2000.

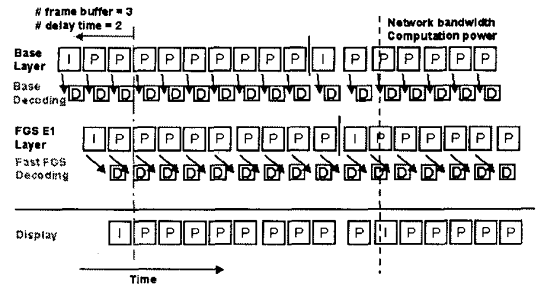

Figure 4: Streaming and Decoding Schedules for Nonbackward-Prediction Streams: Transmission of FGS data is delayed for one-frame interval to provide base-layer prediction

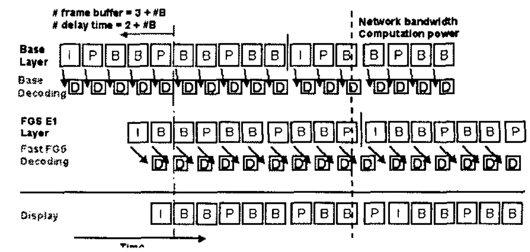

Figure 5: Streaming and Decoding Schedules for Backward-Prediction Streams: When B-frames exist in the base-layer stream, transmission of FGS enhancement is required

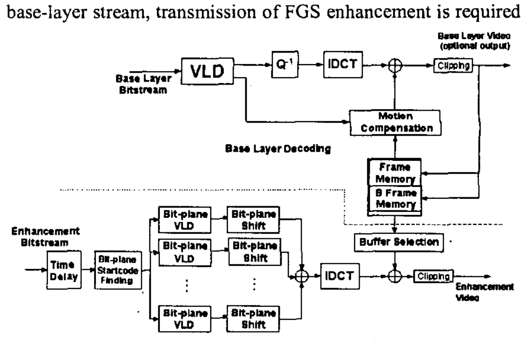

Figure 6: Decoding diagrams for FGS: In the proposed architecture, time delay, multiple-way bitplane decoding and frame sharing are cited.