Systematic Analysis of the Decoding Delay on MVC Decoders

Pablo Carballeira, Julián Cabrera, Fernando Jaureguizar and Narciso García

Abstract — *We present a framework for the analysis of the decoding delay and communication latency in Multiview Video Coding. The application of this framework on MVC decoders allows minimizing the overall delay in immersive video-conference systems*[1].

Index Terms — Three-dimensional video, video-conference, multiview video coding, low latency.

I. INTRODUCTION

3D Video (3DV) and Free Viewpoint Video (FVV) are new types of visual media that expand the user's experience beyond what is offered by 2D video [1]. One common feature of the data formats [2] that can enable those functionalities is the presence of multiview video. The size of this multiview video grows linearly with the number of cameras and the available bandwidth of transmission systems is generally limited. Multiview Video Coding (MVC) [3] is an extension of the H.264/MPEG-4 Advanced Video Coding standard that provides efficient coding of such type of multiview video.

In bidirectional applications such as immersive videoconferencing [4], strict constraints on the delay of multiview video data transfer are imposed. The one-way delay between both ends of the conversation is known as *communication latency*, i.e. the delay between the time instant a video frame is captured and the time instant it is displayed at the receiver. Typical recommendations on maximum communication latency generally state that a serious impact may be observed above 400 ms [5].

Each element (encoder, transmitter, receiver, or decoder) contribute to the delay between the time instant a video frame is captured and the time instant it is decoded at the receiver: the *system delay*. To absorb that *system delay*, and to obtain a constant display frame rate, the receiver needs to introduce a certain delay from the moment it receives the first encoded frame until it is displayed. Whereas in broadcast or on-demand services, this *display delay* value may be over-dimensioned with little impact on the service, in bidirectional services it should be as minimum as possible to reduce the impact on the quality of experience. The minimum value of this delay depends directly on the value of system delay. An accurate computation of the decoding delay is essential to obtain this system delay. Whereas in single view decoders, the computation of the decoding delay can be easily estimated for a given hardware platform, in the case of multiview video coding, the presence of multiple views that need to be decoded simultaneously, and the complexity of multiview prediction structures increases the complexity of the decoding delay analysis.

We present a framework for the systematic analysis of the decoding delay on MVC decoders. The algorithm evaluates the decoding delay taking into account: (i) the multiview prediction structure and (ii) the hardware implementation of the decoder. Analogously to the encoding latency analysis [6], we use graph theory to compute this decoding delay. A graph is constructed from the multiview prediction structure and the maximum decoding delay is computed by solving this graph. The costs of the edges of the graph depend on the frame processing times and are computed iteratively depending on the concurrency of the decoding of several frames.

In Section 2, we present the analysis of the decoding delay and communication latency. In Section 3, we show some results and in section 4 we present the conclusions.

II. ALGORITHM FOR DECODING DELAY ANALYSIS

A. Analysis of communication latency in MVC

In this analysis, we make two major assumptions; (i) a frame is the basic decoding unit (i.e. the decoding of a frame cannot be split into several processes) and (ii) that the decoding of a new frame cannot start until its reference frames have been completely decoded. We also assume that all views have to be decoded, as all of them will be displayed or the receiver can choose any view for display among those received.

The communication latency of the system is given by the frame that has the maximum system delay according to:

$$Lat = \max_{\substack{i=0\ldots N-1 \\ j=0\ldots M-1}} \left(\delta_{\mathrm{cod}\,j}^{\ i} + \delta_{\mathrm{TX}\,j}^{\ i} + \delta_{\mathrm{dec}\,j}^{\ i} \right), \qquad (1)$$

where $\delta_{\mathrm{cod}\,j}^{\ i}$, $\delta_{\mathrm{TX}\,j}^{\ i}$ and $\delta_{\mathrm{dec}\,j}^{\ i}$ represent the components of the system delay (encoding, transmission and decoding delays) of frame x_j^i (frame j of view i), N is the number of views and M is the number of frames per view.

In [6], we present a model to solve $\delta_{\mathrm{cod}\,j}^{\ i}$ for any arbitrary

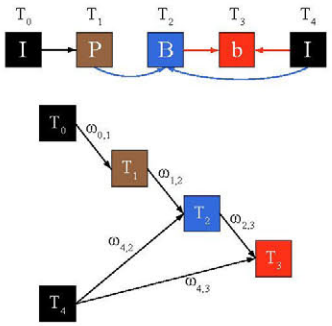

Fig. 1. Example prediction structure and its associated DAG. Nodes of the DAG represent frames while edges represent dependency relationships in the prediction structure.

MVC encoder using graph theory. $\delta_{\mathrm{TX}_j}^{\ i}$ can be evaluated over the transmission channel. To complete the analysis of the system delay, an algorithm to evaluate $\delta_{\mathrm{dec}_j}^{\ i}$ is also needed.

Let $t_{\mathrm{RX}_j}^{\ i}$ be the time instant when encoded frame x_j^i is received at the decoder, and $t_{\mathrm{dec}_j}^{\ i}$ time instant when frame x_j^i is completely decoded. Then:

$$\delta_{\mathrm{dec}_j}^{\ i} = t_{\mathrm{dec}_j}^{\ i} - t_{\mathrm{RX}_j}^{\ i}. \qquad (2)$$

To solve (2), we use a similar approach than that one in [6]. A Direct Acyclic Graph (DAG) is constructed from the prediction structure. Figure 1 shows an example prediction structure for one view and its associated DAG. The costs of the edges of this graph are computed as follows:

$$w_{a,b} = \max\left(0, \left(t_{\mathrm{RX}_a} + \Delta t_{\mathrm{proc}_a}\right) - t_{\mathrm{RX}_b}\right), \qquad (3)$$

where $\Delta t_{\mathrm{proc}_a}$ is the processing time devoted to the decoding of a frame. Then, $\delta_{\mathrm{dec}_j}^{\ i}$ is computed by solving the paths with higher cost in the graph [6].

Generally, the processing capacity of the decoder is limited, and due to the inherent parallelization characteristics of MVC, several frames can be decoded concurrently in the decoder. Therefore, the value of $\Delta t_{\mathrm{proc}_j}^{\ i}$ varies along time depending on the processor occupancy conditions during the decoding process. To compute those processing times we use a hardware decoder model that is described next.

B. Computation of decoding processing times

For the hardware model of the decoder we assume a set of K processors with the following characteristics:

1. Each processor is **not** assigned to a single view: any frame from any view can be decoded in any processor.
2. Multi-task processors: The processors can decode several frames in parallel (e.g. multi-threading)

The frame processing time values on this decoder model are computed iteratively. On each step of this iteration, the DAG is solved and frame processing times and edge costs are updated depending on the processor occupancy conditions on the decoding chronogram. The frame processing times are updated in the case that a number of frames that have to be decoded at a given time instant is greater than the number of processors. For that update we assume that the computational load of a frame is equally distributed into the K processors. The iteration starts with a graph that assumes constant value for the frame processing times.

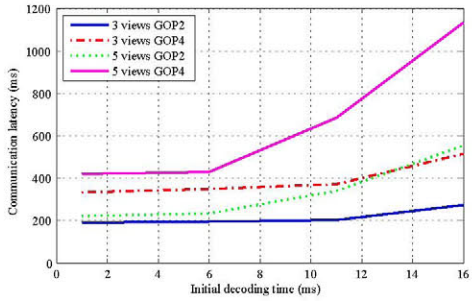

Fig. 2. Communication latency values for different prediction structures and values of the frame decoding time. Initial decoding times correspond to the processing time of a frame with one exclusively dedicated processor.

III. EXPERIMENTAL RESULTS

Figure 2 shows an example of the communication latency values for different JMVM prediction structures and different values of the frame processing times on the decoder. A MVC decoder with only one processor is used. For all experiments, a MVC encoder with a high processing capacity [6] was assumed. It can be seen that for a decoder with a limited processing capacity, the decoding processes of several views increases the minimum communication latency value for high frame processing time values.

IV. CONCLUSION

The analysis of the decoding delay on video decoders is key to establish the minimum display delay that can absorb the system delay. The algorithm we present here to evaluate the decoding delay can be implemented on MVC decoders to reduce the impact of the conversational delay of a videoconference system.

REFERENCES

[1] A. Smolic, K. Müller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegand, "3D video and free viewpoint video - technologies, applications and MPEG standards," in *Proc. IEEE International Conference on Multimedia and Expo*, July 2006, pp. 2161–2164.

[2] S. B. Kang, R. Szeliski, and P. Anandan, "The geometry-image representation tradeoff for rendering," in *Proc. International Conference on Image Processing*, vol. 2, September 2000, pp. 13–16.

[3] A. Vetro, P. Pandit, H. Kimata, A. Smolic, and Y. Wang, "Joint Draft 8.0 on Multiview Video Coding," *Doc. JVT-AB204*, Hannover, Germany, July 2008.

[4] I. Feldmann, O. Schreer, P. Kauff, R. Schäfer, Z. Fei, H. J. W. Belt, and O. D. Escoda, "Immersive multi-user 3D video communication," in *Proc. Int. Broadcast Conf.*, September 2009.

[5] G. Karlsson, "Asynchronous transfer of video," IEEE Communications Magazine, vol. 34, no. 8, pp. 118–126, August 1996.

[6] P. Carballeira, J. Cabrera, A. Ortega, F. Jaureguizar, and N. García, "A graph-based approach for latency modeling and optimization in multiview video encoding," in 3DTV-CON, 2011, May 2011, pp. 1–4.