A Framework for the Analysis and Optimization of Encoding Latency for Multiview Video

Pablo Carballeira, Julián Cabrera, Antonio Ortega, Fernando Jaureguizar, and Narciso García

***Abstract*—We present a novel framework for the analysis and optimization of encoding latency for multiview video. First, we characterize the elements that have an influence in the encoding latency performance: 1) the multiview prediction structure and 2) the hardware encoder model. Then, we provide algorithms to find the encoding latency of any arbitrary multiview prediction structure. The proposed framework relies on the directed acyclic graph encoder latency (DAGEL) model, which provides an abstraction of the processing capacity of the encoder by considering an unbounded number of processors. Using graph theoretic algorithms, the DAGEL model allows us to compute the encoding latency of a given prediction structure, and determine the contribution of the prediction dependencies to it. As an example of DAGEL application, we propose an algorithm to reduce the encoding latency of a given multiview prediction structure up to a target value. In our approach, a minimum number of frame dependencies are pruned, until the latency target value is achieved, thus minimizing the degradation of the rate-distortion performance due to the removal of the prediction dependencies. Finally, we analyze the latency performance of the DAGEL derived prediction structures in multiview encoders with limited processing capacity.**

***Index Terms*—Free viewpoint video, low latency, multiview coding, prediction structures, three-dimensional video (3DV), video-conference.**

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 [2]. 3DV offers a 3D depth impression of the observed scene, while FVV allows an interactive selection of the viewpoint and direction within a certain operating viewing range. To achieve those functionalities a data format richer than a single 2D video signal is needed. The spectrum of data formats that can enable those functionalities goes from purely image-based data formats such as multiview video (multiple views of the same scene) to data formats based on computer graphics such as 3D meshes and their corresponding textures [3]. A widely adopted approach is the one that includes multiview video and depth sequences as additional scene geometry information, allowing the possibility of generating additional views on virtual camera positions [4]–[6]. One common feature of these data formats is the presence of multiview video. Given that the size of this multiview video grows linearly with the number of views, while the available bandwidth is generally limited, an efficient compression scheme for multiview video is needed.

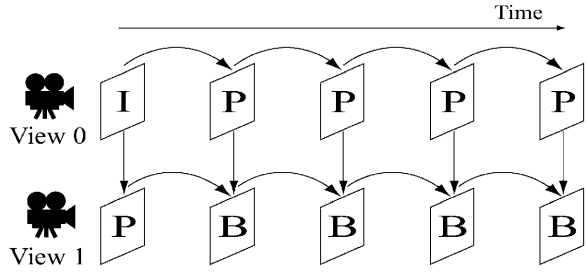

Fig. 1. Example of a multiview prediction structure for two cameras. Horizontal arrows correspond to temporal prediction and vertical arrows to interview prediction.

Multiview video coding (MVC) [7] is an extension of the H.264/MPEG-4 Advanced Video Coding (AVC) [8] standard that provides efficient coding of such type of multiview video. Besides, as depth signals can be represented as monochromatic video signals, MVC has been also commonly used to compress depth [5]. As an extension of AVC, MVC makes use of the set of AVC coding tools. The key additional feature of the MVC design, which increases the coding efficiency specifically for multiview video, is a new prediction relationship among frames of different views that exploits the interview redundancy among cameras. This prediction relationship is known as *interview prediction*. The concept of interview prediction is that motion estimation does not use only temporal references, but also includes interview references. Fig. 1 shows a sample prediction structure in which temporal and interview references are used.

MVC allows a wide range of applications and scenarios [9]. Here, we address real-time applications where strict constraints on the end-to-end delay are imposed, such as live broadcasting or immersive video-conferencing [10]. In video-conferencing scenarios, the one-way delay between both ends of the conversation is known as *communication latency*. Typical recommendations on maximum communication latency generally state that there is none or little impact below 150 ms,

while a serious impact may be observed above 400 ms [11]. Each element (encoder, transmitter, receiver, or decoder) adds some delay and contributes to the communication latency. The delay that the encoder adds to the system is known as *encoding latency* [12].

The encoding latency of a multiview encoder depends mainly on two different but interrelated factors:

- The multiview prediction structure: temporal and inter-view prediction relationships among frames establish coding order dependencies that play a major role in the encoding delay for a given frame.
- The hardware architecture and implementation of the encoder: specific architectural features of multiview encoders (e.g., number of processors, use of threads,...) influence the time that is needed to encode a given frame, and therefore, they affect the performance of the system in terms of latency.

For both single view and multiview encoders a main design variable to control the encoding latency is the prediction structure. In the single view case, encoding latency estimation is relatively simple and it can be reduced by removing long backward temporal dependencies. On the contrary, for multiview encoders, the computation of the encoding latency is much more complex: on the one hand, it requires to handle the richer nature of the multiview prediction structure that includes interview predictions; on the other hand, the fact that the encoder has to manage the encoding of several frames at the same time (frames from several views) makes the hardware platform characteristics play a significant role in the final latency value. Therefore, achieving low latency encoding configurations requires a deeper analysis that should address jointly both prediction structure and hardware architecture. Nevertheless, to the best of our knowledge most research in this area has been focused exclusively on the encoder implementation. Thus, for example, [13]–[15] address the optimization of multiview hardware encoder architectures by reducing frame processing times through the use of parallelization on multicore processors.

Regarding the design of multiview prediction structures, several options have been investigated to obtain efficient prediction structures in terms of rate-distortion (RD) performance. For example, Merkle *et al.* [16] propose different efficient prediction structures. Based on that work, the Joint Video Team (JVT) adopted the prediction structures presented in [17] as the non-normative structure for the Joint Multiview Video Model (JMVM). We argue that the design of multiview prediction structures has been mostly focused on improving RD performance, ignoring important differences in the latency behavior of multiview encoders, which may be critical for delay constrained applications.

In this paper, we propose a general framework for the characterization of the encoding latency in multiview encoders that captures the influence of 1) the prediction structure and 2) the hardware encoder model. This framework allows a systematic analysis of the encoding latency. Firstly, we provide a characterization of the elements that influence the encoding latency performance of the encoder. Second, we propose a method to compute the value of the encoding latency for any arbitrary prediction structure, and to determine the contribution of each prediction dependency to the encoding latency. From these results, we are able to design multiview prediction structures that are efficient in terms of RD and encoding latency.

Our previous work [18] has shown that, for a given encoder hardware platform, an accurate characterization of its encoding latency requires the modeling of the behavior of its specific hardware characteristics. Thus, both analysis and optimization results of encoding latency cannot be general but are particular to that hardware choice. To avoid this limitation, the main focus of our framework is to propose a model that decouples as far as possible the influence of the prediction structure and the hardware architecture on the encoding latency. Stemming from it, the analysis and optimization can be focused on the multiview prediction structure, and later on extended to specific hardware encoder implementations.

The primary element of the proposed framework is an encoding latency model that assumes that the processing capacity of the encoder is essentially unbounded. We will refer to it as the directed acyclic graph encoding latency (DAGEL) model. It can be seen as a task scheduling model [19] (the encoding of a frame is the task unit) that we use not to compute the schedule length, but the encoding latency. We show that the encoding latency values obtained with the DAGEL model are accurate for multiview encoders with a finite number of processors greater than a required minimum, which we are able to identify. Otherwise, results provided by the DAGEL model represent a lower bound to the actual encoding latency of the encoder.

As an example of DAGEL application, we show how the DAGEL model can be used to reduce the encoding latency of a given multiview prediction structure to meet a target value, while preserving as much as possible its RD performance. In this approach, the objective is to prune the minimum number of frame dependencies (those that introduce a higher encoding delay in the original structure) until the latency target value is achieved. Therefore, the degradation of RD performance due to removal of prediction dependencies is limited. RD analyses show that the resulting structures achieve bitrate savings of up to 44%, or PSNR gains of up to 2.49 dB, compared to other commonly used prediction structures of the same encoding latency value.

Finally, we demonstrate that those selected prediction structures, which have a minimum encoding latency (computed with the DAGEL model) as compared to other pruning options, still produce a minimum encoding latency in other models of hardware platforms that do not meet the minimum requirements on the number of processors. In our experiments, for some specific prediction structures, we are able to reduce the number of processors down to a 61% (average value) of the minimum number of processors mentioned before, while the prediction structures designed with the DAGEL model still have a minimum encoding latency value. For cases with an even lower number of processors, we analyze the deviation of the designed prediction structures with respect to the optimal ones in terms of encoding latency.

This paper is organized as follows. In Section II, we present the framework for encoding latency analysis. In Section III, the features of the hardware encoder model are shown. In Section IV, we present the DAGEL model. In Section V, we

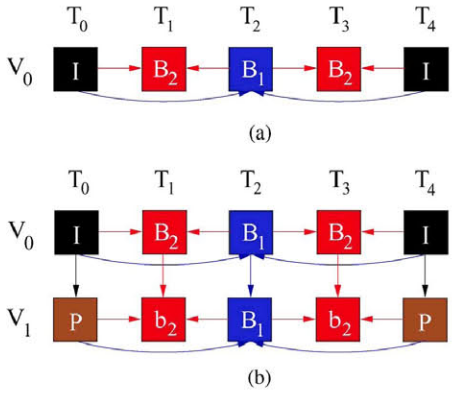

Fig. 2. Prediction structures for single view and multiview video. (a) Single view prediction structure based on a hierarchical temporal prediction scheme. GOP size of four frames. (b) Multiview prediction structure. Additional inter-view prediction is added to the temporal only prediction structure in (a).

show how the DAGEL model allows us to design multiview prediction structures with low latency and a limited penalty on the RD performance. In Section VI, we present the conclusions.

II. Framework for Encoding Latency Analysis

The analysis of encoding latency for a multiview encoder is more complex than for a single view encoder due to the use of interview prediction and the possibility that multiple processors will be used for multiview encoding. This complexity highlights the need for a systematic model for the latency analysis. In order to develop this systematic analysis we make two major assumptions; 1) a frame is the basic encoding unit (i.e., the encoding of a frame cannot be split into several processes) and 2) the encoding of a new frame cannot start until all its reference frames have been completely encoded.

A. Encoding Latency for Single View Video

Let us consider a single-view video encoder. The encoding latency can be defined as the maximum delay (over all frames in the sequence) between the capture of a frame and the instant when that frame is completely coded. That is,

$$Lat = \max_{j=0,\ldots,M-1}(t_{\mathrm{cod}_j} - t_{\mathrm{capt}_j}) \tag{1}$$

where M is the number of frames, t_{cod_j} is the time instant when frame x_j is completely coded and t_{capt_j} is the capture time of x_j. For every frame in the sequence, t_{cod_j} can be computed as

$$t_{\mathrm{cod}_j} = t_{\mathrm{start}_j} + \Delta t_{\mathrm{proc}_j} \tag{2}$$

where t_{start_j} is the instant when the encoding process of x_j starts and $\Delta t_{\mathrm{proc}_j}$ is the corresponding processing time for this frame. The same concepts can be defined considering only frames within a single group of pictures (GOP) of the video sequence. We name *GOP latency* the maximum encoding delay among frames in a single GOP, i.e., the maximum value of (1) only considering frames of that GOP. GOP latency can be different for different GOPs.

Let us consider as an example the prediction structure for a single view encoder in Fig. 2(a). Fig. 3 shows the encoding

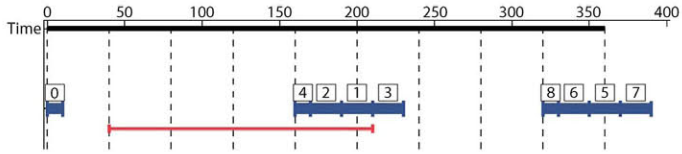

Fig. 3. Encoding chronogram for the prediction structure in Fig. 2(a). Blue bars show the processing time for each of the frames. The label numbers represent the order in which the corresponding frame was captured. The red bar at the bottom shows the encoding latency of the whole sequence.

chronogram for two consecutive GOPs of that prediction structure. The vertical dotted lines mark the capture times of the frames, t_{capt_j}. The beginning of each blue bar corresponds to t_{start_j} whereas its end corresponds to t_{cod_j}. It can be seen that the encoding latency is the time elapsed between t_{capt_1} and t_{cod_1}, and therefore $Lat = t_{\mathrm{cod}_1} - t_{\mathrm{capt}_1}$ in (1).

B. Encoding Latency for Multiview Prediction Structures

The previous ideas can be easily extended to a multiview scheme considering all the frames of the whole set of views. We assume that all the frames of all views have to be transmitted to a receiver and that all will be displayed (or alternatively the receiver can choose any arbitrary view for display among those received). This means that we need to consider the worst-case latency among all frames, i.e., the maximum time elapsed between a frame (from any view) being available at the encoder and its encoded version being ready for transmission. The encoding latency for a multiview encoder can be defined formally as

$$Lat = \max_{\substack{i=0,\ldots,N-1\\ j=0,\ldots,M-1}} \left(t_{\mathrm{cod}_j^i} - t_{\mathrm{capt}_j^i}\right) \tag{3}$$

where N is the number of views, M the number of frames per view, $t_{\mathrm{cod}_j^i}$ is the instant when x_j^i (frame j of view i) is completely coded and $t_{\mathrm{capt}_j^i}$ is the capture time of x_j^i. In the multiview case, $t_{\mathrm{cod}_j^i}$ is computed as

$$t_{\mathrm{cod}_j^i} = t_{\mathrm{start}_j^i} + \Delta t_{\mathrm{proc}_j^i} \tag{4}$$

where $t_{\mathrm{start}_j^i}$ is the instant when the encoding process of x_j^i starts and $\Delta t_{\mathrm{proc}_j^i}$ is the corresponding processing time for this frame. We can also define the *GOP latency* with the same criterion of the single-view case.

We also need to define another relevant time instant in the encoding chronogram of a frame, $t_{\mathrm{ready}_j^i}$, as the instant when x_j^i is ready to be encoded, i.e., all its reference frames have been completely coded. It is computed as

$$t_{\mathrm{ready}_j^i} = \max\left(t_{\mathrm{capt}_j^i}, \max_{l \in L(i,j)}(t_{\mathrm{cod}_l})\right) \tag{5}$$

where $L(i,j)$ is the set of reference frames for x_j^i.

Fig. 2(b) shows a multiview prediction structure for two views. The temporal prediction relationships are the same as in Fig. 2(a) with an additional interview prediction for the second view V_1. Fig. 4 shows the encoding chronogram for that multiview prediction structure where two independent

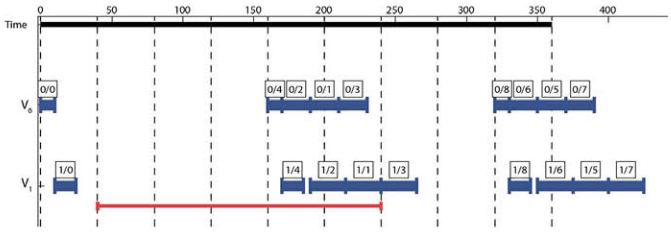

Fig. 4. Encoding chronogram for the prediction structure in Fig. 2(b). Displayed on the vertical axis, two different processors are used to encode each of the two views. The processing time of each frame is labeled with the view number and its capture order, i.e., V_i/T_j.

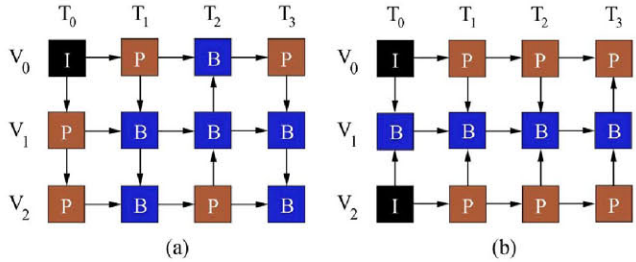

Fig. 5. Examples of arbitrary multiview prediction structures. (a) Prediction structure 1. (b) Prediction structure 2.

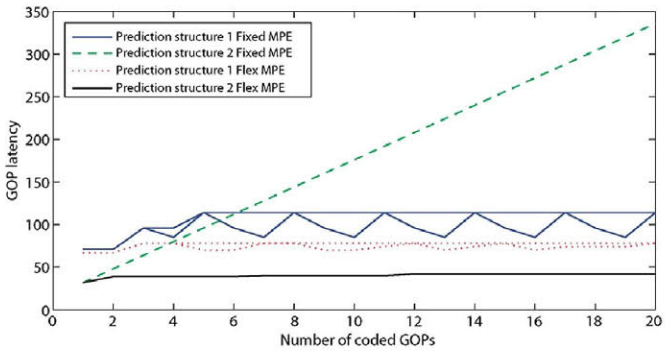

Fig. 6. Comparative analysis of the GOP encoding latency for the multiview prediction structures in Fig. 5. The overall encoding latency is represented as the upped-bound of the graphics.

processors are used on the multiview encoder. Each of them is assigned to encode frames corresponding to only one of the views. In this case, the encoding latency is the time elapsed between $t_{\text{capt}_1^1}$ and $t_{\text{cod}_1^1}$, and therefore $Lat = t_{\text{cod}_1^1} - t_{\text{capt}_1^1}$ in (3). The encoding latency of the multiview case is higher than that of the single view case, as the encoding delay of the frames of V_1 increases due to interview prediction.

As an example of $t_{\text{ready}_j^i}$ in an encoding chronogram, if we consider frame x_3^1 in the prediction structure in Fig. 2(b), its reference frame set, $L(1,3)$, is composed by frames x_2^1, x_4^1 and x_3^0. Therefore, $t_{\text{ready}_3^1}$ is the time instant immediately after that all the frames in $L(1,3)$ have been already encoded. In the chronogram in Fig. 4, $t_{\text{ready}_3^1} = t_{\text{cod}_3^0}$, as x_3^0 is the latest of the frames of $L(1,3)$ to be encoded.

C. *Influence of the Multiview Encoder Hardware Architecture*

While (3), (4), and (5) only depend on the coding order relationships imposed by the prediction structure, and therefore they are valid for all hardware encoder architectures, the relationship between $t_{\text{start}_j^i}$ and $t_{\text{ready}_j^i}$, and the value of $\Delta t_{\text{proc}_j^i}$ depend on the specific hardware encoder architecture being used (e.g., number of processors, sequential or parallel processing, etc.).

In any case, for any hardware encoder architecture, if we assume that a given frame cannot be encoded before its reference frames have been coded, then

$$t_{\text{start}_j^i} \geq t_{\text{ready}_j^i}. \tag{6}$$

Thus, encoding of x_j^i cannot start until all frames in $L(i,j)$ have been coded, but the start of the encoding of x_j^i may be delayed if there are not processing resources available at $t_{\text{ready}_j^i}$.

Encoding latency results are individualized for each specific hardware encoder architecture, i.e., the encoding latency of a given prediction structure in one specific encoder architecture is not necessarily the same in a different one. To illustrate this, we show an example of the evolution of GOP latency for two multiview prediction structures (Fig. 5) which require the same computational load in the encoder (same number of prediction links). Fig. 6 shows the evolution of the GOP latency and the overall encoding latency (upper-bound of the graphics) with the number of coded GOPs. This evolution was computed using (3)–(6) for two different hardware encoder models with the same number of processors: the Fixed multiprocessor encoder (MPE) model and the Flexible MPE model [18]. The main difference between those models is that in the Fixed MPE model the encoding of the frames of one view is assigned to only one processor and in the Flexible MPE frames of any views can be assigned to any of the processors.

In Fig. 6, for the first prediction structure, we identify two different states for both hardware models: a *transient state* that corresponds to the region where the overall encoding latency is increasing and a *steady state* in which the overall encoding latency reaches a final value while the GOP latency can oscillate with the number of coded GOPs. The existence of these two states evidences that, in a practical multiview encoder design, and for certain type of multiview prediction structures, an analysis of the GOP latency evolution is needed, since an analysis of only the first GOPs of the sequence could lead to erroneous results.

Fig. 6 also provides an example of how a given prediction structure can lead to a completely different latency behavior depending on the encoding hardware architecture. Finally, it can be seen that in the case of the second prediction structure and the Fixed MPE model, the encoding latency is not bounded and grows with each additional encoded GOP. On the contrary, the same prediction structure and the Flexible MPE model produce a bounded latency value. Intuitively, the Flexible model is more efficient in the assignment of frames to processors than the Fixed MPE model. While in the Fixed MPE case frames of one view are encoded only by one processor, in the Flexible MPE model idle processor time intervals are filled with frames of different views. This increases the encoding throughput, and reduces the delays in starting to encode as compared to the Fixed MPE case. In the Fixed MPE case, these delays, and the overall encoding latency, are increased with each encoded GOP, resulting in an unbounded encoding latency. The existence of this bound is especially important for videoconferencing encoders. In practice,

it requires that the frame processing times do not exceed certain limits that are specific for each encoder architecture. More details on those limits for these two encoder architecture models can be found in [18].

III. Hardware Encoder Model

We have shown in the previous section that the characteristics of the multiview encoder implementation have an influence on its encoding latency performance. Two different encoder implementations can have considerably different encoding latency performances for the same multiview prediction structure. We next discuss different characteristics of the encoding hardware and the model for frame processing times.

A. Hardware Encoder Features

We define an encoder architecture by the following features.

- Number of processors: Number of independent processors that form the core of the multiview encoder.
- Reference exchange among processors: The capability of the processors to exchange the data corresponding to encoded frames is an indispensable feature in any multiview encoder, to enable interview prediction.
- Single/multi task encoding within each processor: Capability of the processors to encode one or several frames at a given time instant.
- Sequential/parallel frame encoding: The encoding operations for a frame can be done sequentially or in a parallel way on multiple processors.
- Policies to control the frame-to-processor assignment: These assignments can be static; i.e., frames of a camera are always encoded on the same processor, or flexible; i.e., the frames of a given camera are assigned to any of the idle processors at a given time.

B. Frame Processing Time Model

Frame processing times are clearly dependent on the hardware architecture of the multiview encoder, i.e., the processing capacity of the encoder. In our encoding latency framework, these can be fix or variable depending on the single/multi task nature of the processors.

- Single task encoding: During the encoding time period of a given frame the processor is exclusively assigned to the encoding of that frame. Thus, the encoding time of a given frame is equal for different encoder occupancy conditions.
- Multi task encoding: For multiview encoders that allow encoding multiple frames on a given processor, by means of parallelization strategies such as threading, the encoding time of a given frame depends directly on the computational load conditions of the processor.

To establish a realistic scenario, we model the processing time of a given frame as a function of the number of reference frames used to encode that frame. Thus, the processing time $\Delta t_{\mathrm{proc}_j^i}$ for frame x_j^i is

$$\Delta t_{\mathrm{proc}_j^i} = \Delta t_{\mathrm{basic}} + n(i,j)\Delta t_{\mathrm{ref}} \tag{7}$$

where $\Delta t_{\mathrm{basic}}$ is the time dedicated to all the operations not related to motion estimation or compensation, $n(i,j)$ is the number of reference frames for frame x_j^i and Δt_{ref} is the incremental processing delay required for each reference frame.

For single tasks encoding processors, $\Delta t_{\mathrm{basic}}$ and Δt_{ref} can be modeled as constant values for every frame and any time instant. For multi task encoding processors, the values of $\Delta t_{\mathrm{basic}}$ and Δt_{ref} can be variable for different frames and time instants of the chronogram and have to be computed depending on the processor occupancy conditions.

IV. Directed Acyclic Graph Encoding Latency Model

In our framework, it is necessary to know the encoder architecture in order to determine the relationship between $t_{\mathrm{start}_j^i}$ and $t_{\mathrm{ready}_j^i}$ for every frame of the sequence. Given the many factors that affect the encoding latency performance as discussed in Section III, a systematic encoding latency analysis for any arbitrary encoder architecture is challenging. This makes it necessary to compute the encoding latency by simulation for specific encoder architectures [12], [18].

We propose here an encoding latency model that solves the ambiguity in (6) by assuming an unbounded processing capacity. Under this assumption, the encoding latency of the multiview prediction structure can be computed by finding the critical path on a directed acyclic graph extracted from the multiview prediction structure. This model is also valid for single view prediction structures, but in that case, the encoding latency can be intuitively estimated from the longest backward prediction. Nevertheless, the complexity added by multiview prediction structures, and the possibility of processing several views in parallel at the encoder, makes it necessary to use such a systematic model to compute the encoding latency in the multiview case. We name this model the directed acyclic graph encoding latency (DAGEL) model. Details of the model are given in the following subsections.

A. Encoder Architecture With Unlimited Processing Capacity

The relationship in (6) can be simplified if an abstraction of the encoder architecture is chosen, by assuming that the multiview encoder has a sufficiently high number of processors (ideally an infinite number). Then, for any feasible prediction structure it can be guaranteed that there will always be at least one idle processor at any time in which a frame of the multiview sequence is ready for encoding. Then, no policies on the assignment of frames to processors are needed and the frame processing times do not depend on the computational load conditions of the encoder. Formally,

$$t_{\mathrm{start}_j^i} = t_{\mathrm{ready}_j^i} \tag{8}$$

therefore, using (5)

$$t_{\mathrm{start}_j^i} = \max\left(t_{\mathrm{capt}_j^i}, \max_{l \in L(i,j)}(t_{\mathrm{cod}_l})\right) \tag{9}$$

and (4) becomes

$$t_{\mathrm{cod}_j^i} = \max\left(t_{\mathrm{capt}_j^i}, \max_{l \in L(i,j)}(t_{\mathrm{cod}_l})\right) + \Delta t_{\mathrm{proc}_j^i}. \tag{10}$$

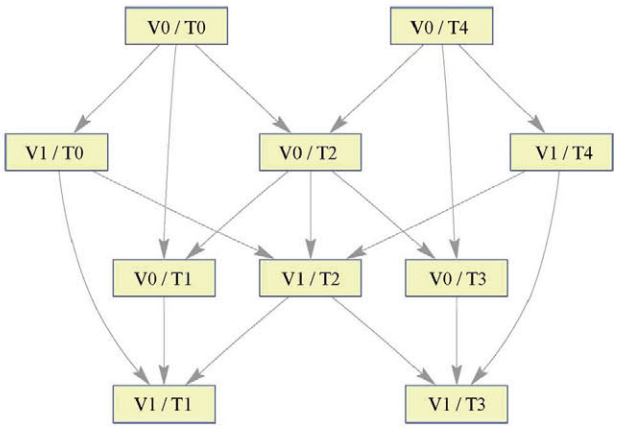

Fig. 7. DAG extracted from the prediction structure of Fig. 2(b). Nodes of the DAG represent frames while edges represent dependency relationships in the prediction structure. V_i/T_j represents frame j of view i as signaled in Fig. 2(b).

If this condition holds, the start of the encoding of a given GOP always begins when the first frame (in coding order) is ready to be coded. That means that a new GOP does not have to wait for other GOPs to be encoded to start its encoding process, so previous GOPs do not add any delay to the latency of the current GOP. Therefore, it is trivial to conclude that GOP latency does not grow with each encoded GOP, and therefore such a multiview encoder will always have a bounded encoding latency.

Also assume that the frame processing time does not depend on the frame content, but only on the number of prediction links. Then, the processing time per GOP is the same for every GOP (if they all have the same prediction structure). Thus, the GOP latency is the same for all the GOPs of the sequence and that value is also equal to the encoding latency of the whole sequence. This simplifies the encoding latency states in Section II-C, and therefore (3)–(5) only have to be computed for the first GOP to obtain the encoding latency value.

B. Definition of the DAGEL Model

Under the condition of (8) we can define the DAGEL model which allows us to systematically solve (3)–(5) for any arbitrary multiview prediction structure. The core of this model is a graph extracted from the multiview prediction structure, where the frames can be seen as nodes of the graph and the prediction dependencies as the edges. Due to the directed nature of the dependencies (one frame is predicted from the reference frame but not vice versa), the graph is directed. Each directed edge links a reference frame to the frame that is predicted from it. A path is a sequence of nodes linked by directed edges. Fig. 7 shows the directed graph derived from the prediction structure in Fig. 2(b).

Any dependency graph extracted from a feasible multiview prediction structure is necessarily a directed acyclic graph (DAG), i.e., a directed graph with no directed cycles [20]. If a directed cycle existed in the prediction structure, it would correspond to a situation in which a frame x_a is predicted from x_b, and x_b is predicted from x_a indirectly in a series of prediction steps. As this structure is not feasible, the graph is necessarily a DAG.

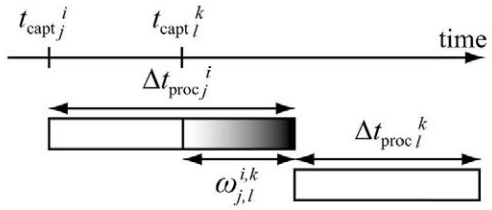

Fig. 8. Graphic significance of the cost value of the edges in the graph. The cost value $\omega_{j,l}^{i,k}$ is the delay introduced by the encoding process of frame x_j^i in the encoding process of frame x_l^k.

Each edge of the DAG has an associated cost value that indicates the delay added by a parent frame to the encoding process of its child frame. The cost value $\omega_{j,l}^{i,k}$ of the edge that links node x_j^i with node x_l^k is

$$\omega_{j,l}^{i,k} = \max\left(0, \left(t_{\text{capt}_j^i} + \Delta t_{\text{proc}_j^i}\right) - t_{\text{capt}_l^k}\right) \tag{11}$$

where $t_{\text{capt}_j^i}$ and $t_{\text{capt}_l^k}$ are the capture times of frames x_j^i and x_l^k, respectively, and $\Delta t_{\text{proc}_j^i}$ is the processing time of frame x_j^i.

Fig. 8 illustrates the computation of $\omega_{j,l}^{i,k}$ with a time chronogram in which the encoding process of parent frame x_j^i delays the encoding start of child frame x_l^k. As only positive delay values have a realistic meaning, $\omega_{j,l}^{i,k}$ is restricted to positive values.

The cost of a path is the sum of the costs of the edges that link the nodes in the path. Among the set of paths ending on the same node, we name *delay path* the one that has the highest cost value. The cost of the delay path indicates the elapsed time between the capture of that frame and the instant when all its reference frames have been encoded. If (8) holds, then (3), (4) and (5) can be systematically computed using the cost of the delay paths. For any frame of the multiview sequence

$$t_{\text{start}_j^i} = t_{\text{capt}_j^i} + p_{del_j^i} \tag{12}$$

where $p_{del_j^i}$ is the cost of the delay path of frame x_j^i. From (3), (4) and (12) it can be derived that the encoding latency value is

$$Lat = \max_{\substack{i=0,\ldots,N-1 \\ j=0,\ldots,M-1}} \left(p_{del_j^i} + \Delta t_{\text{proc}_j^i}\right). \tag{13}$$

Lat in (13) corresponds to the delay of the *critical path*.

The latency value obtained using the DAGEL model considers the effects of both the multiview prediction structure and the individual processing time of each frame if an unlimited processing capacity is available. Nevertheless, for each feasible prediction structure with a finite number of views, frame rate and frame processing times, it can be proved (see Section IV-D) that there exists a minimum number of processors, $K_{\min}$, that guarantees the availability of an idle processor for any frame ready to be encoded. Therefore, for a hardware encoder model with at least $K_{\min}$ processors, the latency value obtained with the DAGEL model corresponds to its real encoding latency.

For a multiview encoder with number of processors $K < K_{\min}$, the encoding latency value obtained with the DAGEL model is a lower bound to the encoding latency of the encoder. Intuitively, fewer processors results in delays to the encoding process for certain frames since it will be more likely that all

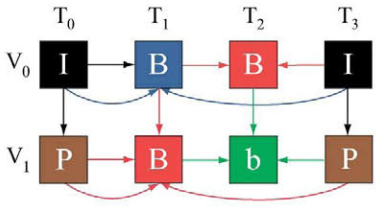

Fig. 9. GOP example.

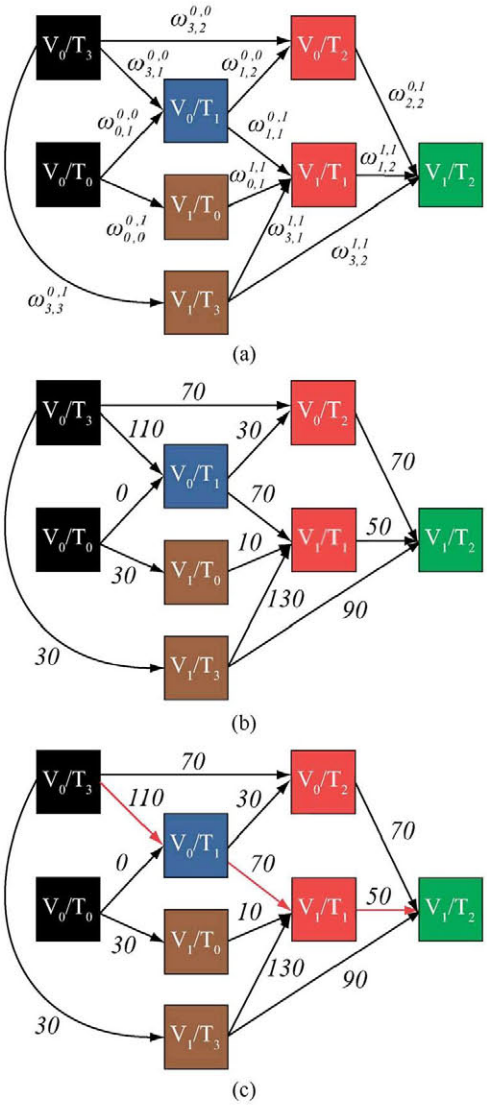

Fig. 10. Example of encoding latency analysis using the DAGEL model. (a) DAG extracted from the prediction structure in Fig. 9. (b) DAG with edge cost values. (c) DAG with edge cost values and critical path (red).

the processors are busy at certain times. Furthermore, if in this situation the processing times exceed certain limits, the encoding latency will not be bounded and will increment GOP by GOP, making the system not suitable for real-time encoding applications.

C. Example of Encoding Latency Computation Using the DAGEL Model

To illustrate the aforementioned concepts, we present an example of encoding latency analysis using the DAGEL model.

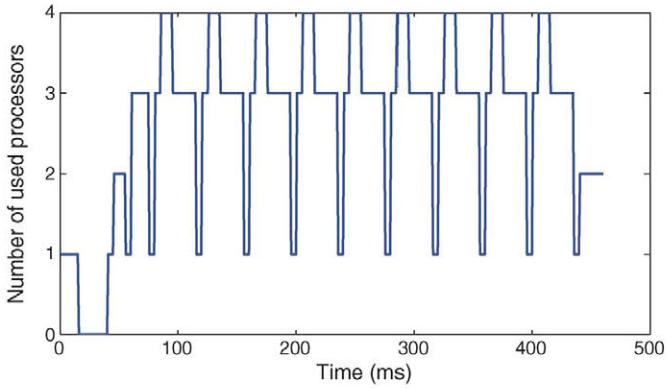

Fig. 11. Chronogram showing the number of processors needed for the encoding process of the prediction structure in Fig. 2(b).

For clarity reasons, we present an example of a simple prediction structure for two views (Fig. 9). From that prediction structure, the DAG that is shown in Fig. 10(a) is extracted.

The example time parameter values are the following: $\Delta t_{\text{basic}} = 30$ ms, $\Delta t_{\text{ref}} = 20$ ms and the elapsed time between the capture of two consecutive frames is $T_{\text{capt}} = 40$ ms. The values of the edge costs can be computed using (11). These values are shown in Fig. 10(b). For those edge cost values, the critical path of the graph is the one marked in red in Fig. 10(c). For the given time values, the encoding latency of the example prediction structure is

$$Lat = p_{del_2^1} + \Delta t_{\text{proc}_2^1} = 230 + 90 = 320 \text{ ms}. \quad (14)$$

This procedure can be programmed to obtain a systematic way to analyze the encoding latency of arbitrary multiview prediction structures.

D. Minimum Number of Processors in the DAGEL Model

In Section IV-A, we assumed an unlimited processing capacity in the multiview encoder that allows us to develop the DAGEL model. However, the minimum number of processors $K_{\min}$ that ensures that the condition in (8) holds can be computed by analyzing the number of frames that are encoded simultaneously. Given the DAGEL model and (4) and (12), the number of frames that are encoded simultaneously at any time interval can be easily computed. Let $N_{\max}$ be the maximum number of simultaneously encoded frames for a given prediction structure and frame processing time values. Thus, for any hardware platform with a number of processors $K \geq N_{\max}$, in which frames can be assigned to any processor, the condition in (8) holds. Therefore, the minimum number of processors that validates the DAGEL model, $K_{\min}$, is $N_{\max}$.

Fig. 11 shows the chronogram of processor usage for the example prediction structure in Fig. 2(b). The chronogram shows the results for the encoding of several GOPs. The maximum value of this chronogram ($N_{\max}$) is then $K_{\min}$ in that encoder example. Fig. 12 shows the evolution of $K_{\min}$ with the value of the parameters Δt_{basic} and Δt_{ref}, (frame processing time model presented in Section III-B), and the prediction structure in Fig. 2(b). As shown in the figure, higher frame processing times result in a higher value of $K_{\min}$.

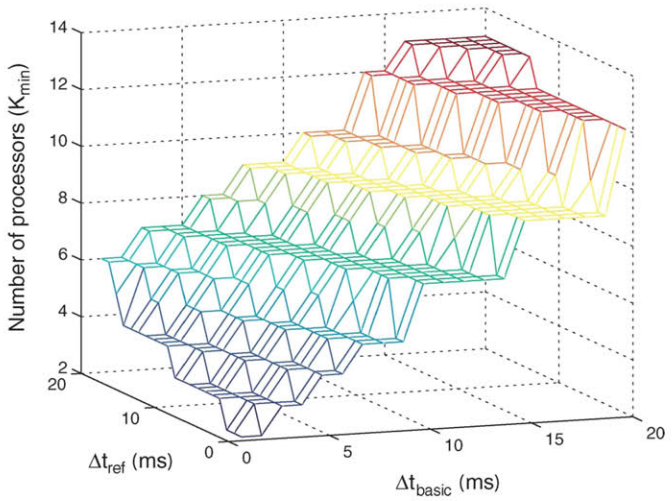

Fig. 12. Number of processors of the ideal encoder model for the prediction structure in Fig. 2(b) and different values of the frame time processing model.

This shows that for a given prediction structure and a set of frame processing times, we can obtain the minimum number of processors of the encoder so that the condition in (8) holds. Alternatively, for a given prediction structure and number of processors, we can obtain the maximum time that is available to be devoted to frame processing.

V. Application of the DAGEL Model: Encoding Latency Reduction for Multiview Prediction Structures

To demonstrate the capabilities of our DAGEL model, we show how it can be used to reduce the encoding latency of a given prediction structure down to a target encoding latency value. Using the DAGEL model, we are able to identify the prediction links that add a higher encoding delay in order to prune them. Assuming that the degradation of the RD performance is proportional to the number of pruned links, the objective is to prune the minimum number of frame dependencies until the latency target value is achieved. As a result of limiting the degradation of the RD performance for a target latency, we obtain new prediction structures that have a considerably better RD performance than commonly used multiview prediction structures with the same encoding latency.

The selection of those dependency links to be pruned is based on a comparison of encoding latency values of several prediction structures computed using the DAGEL model. Thus, this will be valid for multiview encoders that match the processing capacity requirements of the DAGEL model. However, there is no guarantee that those comparative encoding latency results will be achieved when fewer processors are used. Thus, we analyze the deviation of the comparative encoding results obtained with the DAGEL model when applied to multiview encoders with limited processing capacity. To sum up, we show that efficient prediction structures in terms of rate-distortion-latency can be designed using the DAGEL model, and that those are valid for real multiview encoders with limited processing capacity.

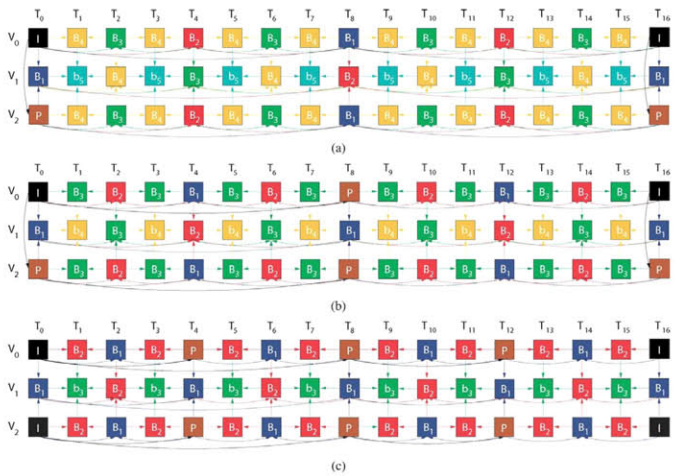

Fig. 13. Latency reduction by edge pruning in the DAG. (a) Prediction structure JMVM GOP16. (b) Prediction structure GOPx4. (c) Prediction structure GOPx10.

TABLE I
Time Parameter Values for the Frame Processing Time Model and Capture Period

Time parameter	Δt_{basic}	Δt_{ref}	Capture Period
Value (ms)	20	10	40 (25 fps)

TABLE II
Encoding Latency Values for JMVM Prediction Structures and Prediction Structures Obtained Using the DAGEL Model

	GOP16	GOP8	GOP4	GOPx4	GOPx10
Encoding latency (ms)	930	550	330	550	330

A. RD Analysis of Low Encoding Latency Multiview Prediction Structures

In order to assess the RD performance of the DAGEL based pruning approach, we have considered an initial prediction structure with three views, a GOP size of 16 frames and IBP prediction for the interview prediction scheme [17] [GOP16, Fig. 13(a)]. Then, we have iteratively pruned its associated DAG to reduce its encoding latency to the value of analogous JMVM structures with GOP sizes of 8 (GOP8) and 4 frames (GOP4). This has been done by an increasing number of cuts in the DAG, using an exhaustive search of the possible cut combinations. In this set of experiments the time parameter values of the frame processing time model (see Section III-B) have been estimated using the X264 software implementation of AVC in a general purpose PC: a QuadCore processor working at 2.40 GHz with 3.25 GB of RAM memory. The time parameter values are shown in Table I. The encoding latencies of these prediction structures are shown in Table II.

The structures obtained for four cuts (GOPx4) and ten cuts (GOPx10) are shown in Fig. 13(b) and (c), respectively. As shown in Table II, these structures have the same encoding latency value that JMVM structures GOP8 and GOP4, respectively. We have evaluated the RD performance of the different prediction structures using the JMVM software version 2.1 [27] and the MVC common conditions [28] for several multiview sequences with different characteristics (see Table III).

TABLE III
CHARACTERISTICS OF THE MULTIVIEW SEQUENCES USED FOR THE RD TESTS. MORE DETAILS FOR BALLROOM CAN BE FOUND IN [21], RACE1 AND FLAMENCO2 IN [22], NEWSPAPER IN [23], KENDO IN [24], BALLET IN [25]

Sequence	Spatial resolution	Frame rate	Moving cameras	Camera configuration	Array orientation	Camera distance
Ballroom	640x480	25 fps	No	Parallel	Horizontal	20 cm
Race1	640x480	30 fps	Yes	Parallel Convergent	Horizontal	20 cm
Flamenco2	640x480	30 fps	No	Parallel Cross	Vertical	20 cm
Newspaper	1024x768	30 fps	No	Parallel	Horizontal	5 cm
Kendo	1024x768	30 fps	Yes	Parallel	Horizontal	5 cm
Ballet	1024x768	15 fps	No	Arc	Horizontal	20 cm

The RD results are shown in Fig. 14, and Table IV shows the average RD differences [26] of GOPx4 and GOPx10 structures compared to GOP8 and GOP 4, respectively. GOPx4 shows an average PSNR gain of 0.25 dB and bitrate saving of 6.69% compared to GOP8, and GOPx10 shows an average PSNR gain of 0.90 dB and bitrate saving of 20% compared to GOP4. These results show that, for the tested sequences, the prediction structures obtained using the DAGEL model outperform the JMVM prediction structures with the same encoding latency value in terms of RD performance. This makes our structures more efficient to be used in applications with strict requirements in end-to-end delay than the commonly used JMVM structures.

TABLE IV
RD-BJONTEGAARD RESULTS [26] COMPARING PREDICTION STRUCTURES OBTAINED USING THE DAGEL MODEL AND JMVM PREDICTION STRUCTURES

	GOPx4/GOP8		GOPx10/GOP4	
	ΔPSNR(dB)	Δbitrate(%)	ΔPSNR(dB)	Δbitrate(%)
Ballroom	0.20	-5.33	0.71	-17.11
Race1	-0.04	1.05	0.29	-6.77
Flamenco2	0.01	-0.22	0.27	-5.44
Newspaper	0.70	-15.46	2.49	-44.48
Kendo	0.07	-1.83	0.47	-11.75
Ballet	0.54	-18.36	1.16	-34.46
Average	**0.25**	**-6.69**	**0.90**	**-20.00**

B. Comparison of DAGEL Results in Multiview Encoders With Limited Processing Capacity

1) Comparison Methodology: Let us consider a set of different prediction structures $\{S_i\}$ with $i = 0, \ldots, M-1$. Let their respective encoding latencies, computed with the DAGEL model, be $\{Lat_{\mathrm{DAGEL}}(S_i)\}$. Then, let us consider that

$$\exists j | Lat_{\mathrm{DAGEL}}(S_j) \leq Lat_{\mathrm{DAGEL}}(S_i),\ \forall i. \tag{15}$$

If we assume, without loss of generality, that $\{S_i\}$ have a comparable RD performance, the structure S_j would be the most amenable to be used in a video conferencing system, as its encoding latency is the lowest among the set.

Then, let $\{Lat_{\mathrm{HW}}(S_i)\}$ be the encoding latency of the set of structures $\{S_i\}$ in a given hardware encoder model. If we had assumed, from the DAGEL results, that $Lat_{\mathrm{HW}}(S_j)$ is the minimum encoding latency of the set $\{Lat_{\mathrm{HW}}(S_i)\}$, we have the following relative prediction error:

$$err(\%) = 100 \left(\frac{Lat_{\mathrm{HW}}(S_j) - Lat_{\min}}{Lat_{\min}} \right) \tag{16}$$

where $Lat_{\min}$ is the minimum encoding latency of the set of prediction structures on the given hardware encoder

$$Lat_{\min} = \min \left(Lat_{\mathrm{HW}}(S_i) \right),\ \forall i. \tag{17}$$

To analyze the mismatching between the DAGEL model and more realistic hardware encoder models, we have considered the following approach: we start from an initial prediction structure with a given encoding latency and a lower target encoding latency value. Then, we use an algorithm, based on the DAGEL model, to remove the prediction links that introduce the highest encoding delays and to obtain a set of candidate prediction structures that have an encoding latency value lower than our target. Then, we evaluate the prediction error, in terms of encoding latency, of using the prediction structure with lowest latency in the DAGEL model in a realistic hardware encoder architecture. For the latency reduction algorithm we have used the *Tree-Search* algorithm (Section V-B2) and the Flexible MPE model [18] for the hardware encoder model. The Flexible MPE model considers a multiview encoder with N views and K processors, with the following characteristics.

- Each processor is **not** assigned to a single view: any frame from any of the N views can be encoded in any of the K processors.
- The processors encode their assigned frames sequentially.
- If, at a given time, several frames are ready to be encoded and K_f out of K processors are free:
  - — First, these frames are ordered by a frame priority value λ_j^i, that depends on the capture time of frame x_j^i and the capture times of the frames that are predicted from it.
  - — Then, the first K_f frames in the frame order list are encoded by assigning each one of them to one of the K_f free processors.

More details on how λ_j^i can be computed for any frame x_j^i can be found in [18].

2) Tree-Search Algorithm: Let us consider a target encoding latency Lat_T and an initial multiview prediction structure with n_L prediction links. To obtain a new prediction structure with encoding latency Lat_T, a number of links n_C have to be

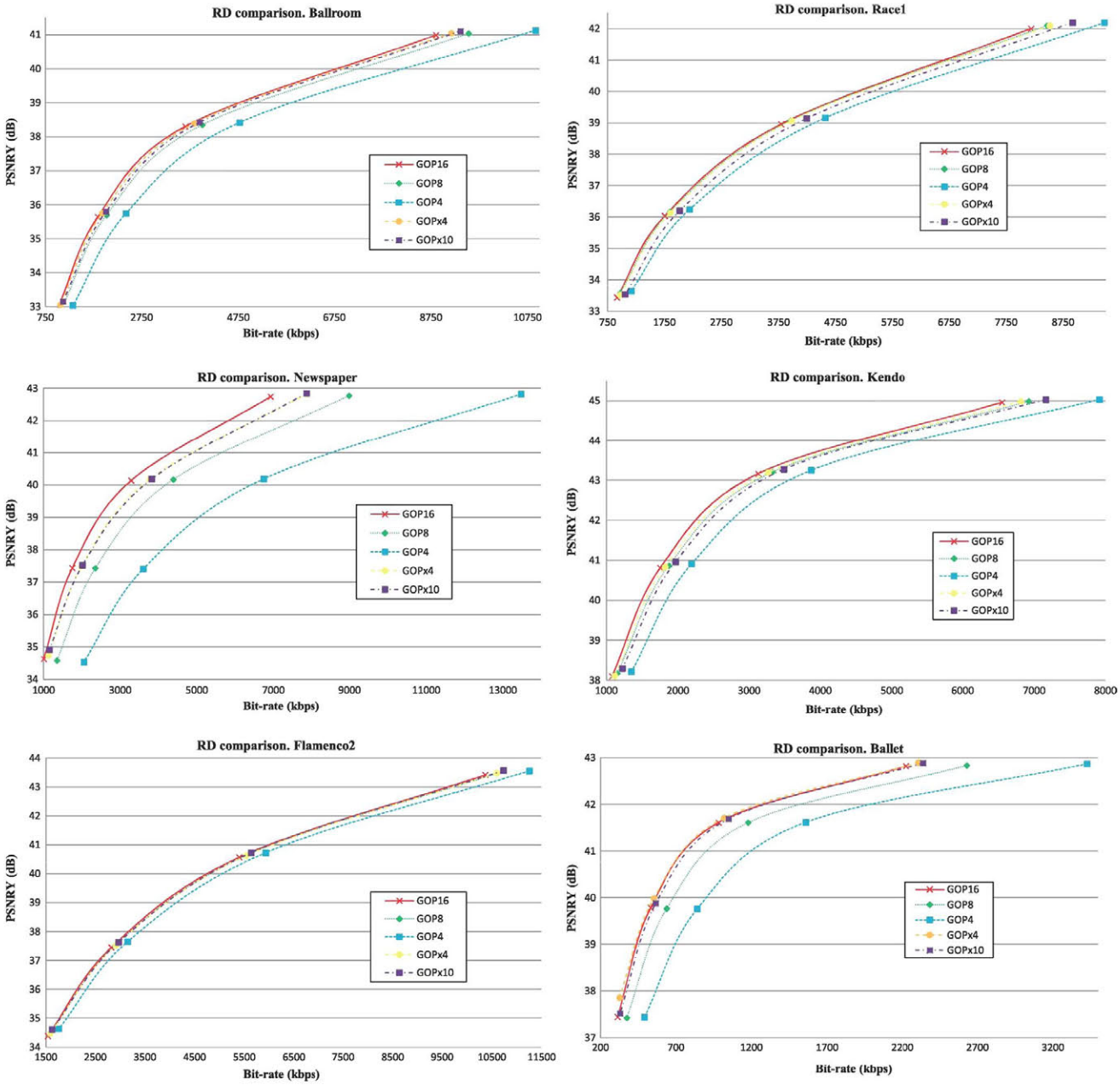

Fig. 14. RD comparison of JMVM prediction structures with a GOP size of 16, 8, and 4 frames and the pruned multiview prediction structures GOPx4 and GOPx10 (see Fig. 13). The PSNR values of the luminance signal averaged for all decoded views are plotted against bitrate values of the MVC coded streams. Results for the following multiview sequences: Ballroom, Race1, Newspaper, Kendo, Flamenco2, and Ballet.

pruned from the original structure. The exhaustive search of all cut combinations is a computationally intensive algorithm. The exhaustive search of all the possible cut combinations of n_C cuts over n_L links implies the evaluation of $C_{n_L}^{n_C}$ prediction structures:

$$C_{n_L}^{n_C} = \frac{n_L!}{n_C!(n_L - n_C)!}. \tag{18}$$

Since in multiview prediction structures n_L is usually high, if the number of cuts is high, it requires a high computational load for all the possible combinations. For example, for a JMVM structure of 3 views and a GOP size of 8 frames, $n_L = 36$. For $n_C = 3$, the number of prediction structures that have to be evaluated is 37820.

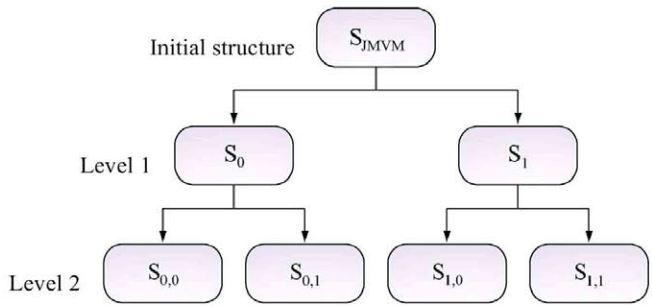

Fig. 15. Example of the Tree Search algorithm for $n_C = 2$ and $n_B = 2$. The states of the tree correspond to prediction structures obtained by cutting different prediction links in the parent prediction structure.

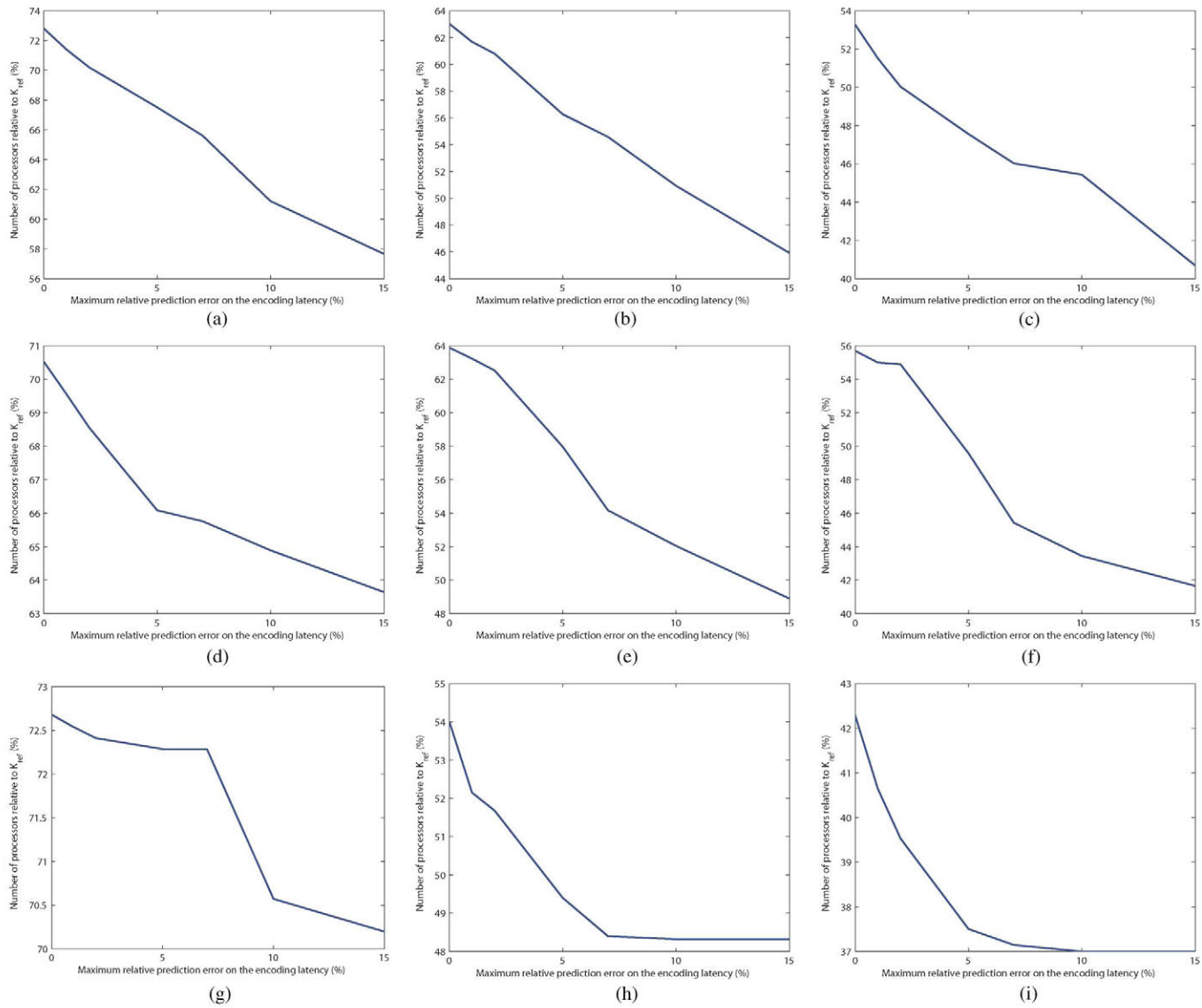

Fig. 16. Results of the comparison of the DAGEL model with the Flexible MPE model. For a given maximum relative prediction error on the encoding latency, the graphs show the minimum number of processors relative to K_{ref} for a multiview encoder within the Flexible MPE model. (a) 3 views GOP4; (b) 3 views GOP8; (c) 3 views GOP16; (d) 5 views GOP4; (e) 5 views GOP8; (f) 5 views GOP16; (g) 7 views GOP4; (h) 7 views GOP8; (i) 7 views GOP16.

Here, we propose a sub-optimum algorithm as a less computationally intensive alternative to the exhaustive search. We name this algorithm the *Tree-Search*. With the Tree-Search algorithm, for a target encoding latency Lat_T, we increase the number of link cuts in a decision-tree manner, where n_B is the number of branches for each node of the tree, creating sets of candidate prediction structures in each lower level of the tree until a prediction structure with a latency equal or lower to Lat_T is obtained. The resulting number of cuts of that selected prediction structure is n_C. Fig. 15 shows an example of the decision-tree for $n_C = 2$ and $n_B = 2$. In the following, we explain the Tree-Search algorithm with the help of this example.

Let S_{JMVM} (level 0 in the example) be the initial prediction structure. The states on the lower levels of the tree are generated by the following operations:

- For each parent prediction structure, its critical path is found using the DAGEL model.
- The encoding latency of the prediction structures that are obtained by cutting each of the links of that critical path is evaluated.
- As many as n_B prediction structures with the lowest encoding latency among the evaluated ones are selected to generate the children of the parent state in the immediately lower level. In the example, as $n_B = 2$, each parent structure generates two children structures.
- If any of the structures of the lowest level have a encoding latency equal or lower to Lat_T, the algorithm finishes. If not, another level of the tree is created.

Among the prediction structures on the lowest level, the one with the lowest encoding latency is selected.

3) Comparison With the Flexible MPE Model: Starting from an initial JMVM prediction structure, a pair of values $(\Delta t_{\text{basic}}, \Delta t_{\text{ref}})$ and a target encoding latency Lat_T, we apply the Tree-Search algorithm with a given number of branches n_B. As a result, we obtain a prediction structure S_{ref} and the

TABLE V
RESULTS OF THE COMPARISON OF THE DAGEL MODEL WITH THE FLEXIBLE MPE MODEL. MINIMUM REQUIREMENTS ON THE NUMBER OF PROCESSORS TO OBTAIN A ZERO RELATIVE PREDICTION ERROR ON ENCODING LATENCY FOR LOW-LATENCY MULTIVIEW PREDICTION STRUCTURES. RESULTS RELATIVE TO K_{ref} AND MULTIVIEW ENCODERS WITHIN THE FLEXIBLE MPE MODEL

	Prediction structures								
	3 views GOP4	3 views GOP8	3 views GOP16	5 views GOP4	5 views GOP8	5 views GOP16	7 views GOP4	7 views GOP8	7 views GOP16
Number of processors (%)	72.80	63.03	53.27	70.52	63.88	55.69	72.68	54.00	42.29

minimum number of processors K_{ref} that the DAGEL model assumes for that prediction structure.

Then, the prediction structures on the lowest level of the tree are compared in terms of encoding latency using the Flexible MPE model with a number of processors $K < K_{\text{ref}}$. Among the prediction structures in the lowest level of the tree, $Lat_{\min}$ is the lowest encoding latency value, and Lat_{ref} is the encoding latency of S_{ref} computed with the Flexible MPE model of K processors. The relative prediction error of using S_{ref} as the prediction structure with the lowest encoding latency value on that multiview encoder is computed using (16). This process is repeated for all the possible values of $K = 1, \ldots, K_{\text{ref}} - 1$.

For simplicity, in our experiments, we have used a fixed value of $n_C = 2$ in the Tree-Search algorithm. The experiment was repeated for different initial JMVM prediction structures with IBP prediction for the interview prediction scheme [17], and different values of $(\Delta t_{\text{basic}}, \Delta t_{\text{ref}})$. The initial prediction structures vary from one to seven views and a GOP size of 4, 8, and 16 frames. While the time period between frame capturing is $T_{\text{capt}} = 40$ ms, both frame processing time parameters Δt_{basic} and Δt_{ref} vary between 1 and 20 ms. We have empirically assessed that $n_B = 4$ guarantees that the results of the Tree-Search are the same as those of the exhaustive search for the prediction structures and time parameter values of the tests. We have used this value for the Tree-Search algorithm.

Fig. 16 presents the experimental results. Based on the analysis of the relative prediction error, we have computed the minimum number of processors of a multiview encoder within the Flexible MPE model, that guarantees a given maximum relative prediction error for the encoding latency. To allow a better comparison among graphs, results are presented relative to K_{ref}. This way, for an initial $K_{\text{ref}} = 4$, a value of 50% corresponds to a multiview encoder with two processors. For example, in Fig. 16(d), for a relative encoding error of 5% the minimum number of processors is 66%. That means that for that initial prediction structure, we can use the DAGEL model to design a new prediction structure with low encoding latency, and the number of processors can be reduced to a 66% of K_{ref} with a maximum relative prediction error on encoding latency of 5%. The results are specified for each initial JMVM prediction structure and averaged for all the evaluated values of $(\Delta t_{\text{basic}}, \Delta t_{\text{ref}})$.

Table V shows, for each of the initial prediction structures, the number of processors relative to K_{ref} that can be used in a multiview encoder while guaranteeing that there no prediction error on the comparative encoding latency results. That is, the limits on computational capacity for multiview encoders for which the conclusions derived from the DAGEL model are still valid. The results show the comparative results obtained by the DAGEL model are valid for encoder implementations with a number of processors between 42% and 72% of K_{ref}. That shows that, given a prediction structure with the lowest latency against other pruning options in the DAGEL model, we can use encoder configurations with a fewer number of processors than K_{ref} while that prediction structure still has the minimum encoding latency.

VI. CONCLUSION

We have presented a framework that allows a systematic analysis of encoding latency on multiview encoders. With the aim of decoupling the two main factors involved in the characterization of the encoding latency, the multiview prediction structure and the hardware encoder model, we have proposed the DAGEL model. This encoding latency model assumes that the processing capacity of the encoder is essentially unbounded, so that the latency only depends on the multiview prediction structure and the frame processing times. We have shown that the DAGEL model allows us to formalize any prediction structure as a direct acyclic graph. Therefore, by means of graph theoretic algorithms, its encoding latency can be computed as well as the contribution of the prediction dependencies to it. Moreover, we have proved that the encoding latency values obtained with the DAGEL model are accurate for multiview encoders with a finite number of processors greater than a required minimum, which we are able to identify. Otherwise, results provided by the DAGEL model represent a lower bound to the actual encoding latency of the encoder.

As an example of DAGEL application, we have used the DAGEL model to reduce the encoding latency of a given multiview prediction structure to meet a target value, while preserving the RD performance as much as possible. The objective of this approach has been to prune the minimum number of frame dependencies (those that add a higher encoding delay in the original structure) until the latency target value is achieved. Therefore, the degradation of the RD performance due to removal of the prediction dependencies has been limited. RD analyses have shown that the resulting structures achieve bitrate savings of up to 44% or PSNR gains of up to 2.49 dB compared to other commonly used prediction structures with the same encoding latency value.

Finally, we have demonstrated that those selected prediction structures which a have minimum encoding latency (computed with the DAGEL model) among other pruning options, still have a minimum encoding latency in other hardware architecture platforms that do not meet the minimum requirements on the number of processors. In our experiments, for some specific prediction structures and the Flexible MPE model, we have proven that encoder configurations with a number of processors of down to 42% of the processors assumed by the DAGEL model, could be used while the prediction structures designed with the DAGEL model still have a minimum encoding latency value.

REFERENCES

[1] 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 *Proc. IEEE 3DTV Conf.: The True Vision—Capture, Transmission, Display of 3D Video (3DTV-CON), 2011*, May 2011, pp. 1–4.

[2] 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 Int. Conf. Multimedia Expo*, Jul. 2006, pp. 2161–2164.

[3] S. B. Kang, R. Szeliski, and P. Anandan, "The geometry-image representation tradeoff for rendering," in *Proc. Int. Conf. Image Process.*, Sep. 2000, vol. 2, pp. 13–16.

[4] M. Tanimoto, M. Tehrani, T. Fujii, and T. Yendo, "Free-viewpoint TV," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 67–76, Jan. 2011.

[5] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2007, pp. 201–204.

[6] "Call for Proposals on 3D Video Coding Technology," ISO/IEC JTC1/SC29/WG11. Geneva, Switzerland, Mar. 2011, *output doc. N12036*.

[7] A. Vetro, P. Pandit, H. Kimata, A. Smolic, and Y. Wang, "Joint draft 8.0 on multiview video coding," Jul. 2008, *Doc. JVT-AB204*, Hannover, Germany.

[8] ITU-T Rec. & ISO/IEC, "14496-10 AVC advanced video coding for generic audiovisual services," 2005.

[9] A. Vetro, T. Wiegand, and G. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard," *Proc. IEEE*, vol. 99, no. 4, pp. 626–642, Apr. 2011.

[10] 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.*, Sep. 2009.

[11] G. Karlsson, "Asynchronous transfer of video," *IEEE Commun. Mag.*, vol. 34, no. 8, pp. 118–126, Aug. 1996.

[12] P. Carballeira, J. Cabrera, A. Ortega, F. Jaureguizar, and N. García, "Comparative latency analysis for arbitrary multiview video coding prediction structures," in *Proc. IS&T/SPIE Int. Conf. Visual Commun. Image Process., VCIP 2009*, Jan. 2009, vol. 7257, pp. 72 570L-1–72 570L-12.

[13] Y. Yang, G. Jiang, M. Yu, and D. Zhu, "Parallel process of hyper-space-based multiview video compression," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2006, pp. 521–524.

[14] Y. Pang, L. Sun, S. Guo, and S. Yang, "Spatial and temporal data parallelization of multi-view video encoding algorithm," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, Oct. 2007, pp. 441–444.

[15] Y. Pang, L. Sun, J. Wen, F. Zhang, W. Hu, W. Feng, and S. Yang, "A framework for heuristic scheduling for parallel processing on multicore architecture: A case study with multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 11, pp. 1658–1666, Nov. 2009.

[16] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1461–1473, Nov. 2007.

[17] A. Vetro, P. Pandit, H. Kimata, A. Smolic, and Y. Wang, "Joint Multiview Video Model (JMVM) 7.0," *Doc. JVT-Z207*. Antalya, Turkey, Jan. 2008.

[18] P. Carballeira, J. Cabrera, A. Ortega, F. Jaureguizar, and N. García, "Latency analysis for a multi-processor multiview video encoder implementation," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf., APSIPA ASC '09*, Oct. 2009, pp. 367–372.

[19] Y.-K. Kwok and I. Ahmad, "Static scheduling algorithms for allocating directed task graphs to multiprocessors," *ACM Comput. Surveys*, vol. 31, no. 4, pp. 406–471, Dec. 1999.

[20] K. Thulasiraman and M. N. S. Swamy, *Graphs: Theory and Algorithms*. New York: Wiley, 1992.

[21] A. Vetro, M. McGuire, W. Matusik, A. Behrens, J. Lee, and H. Pfister, "Multiview video test sequences from MERL," *MPEG Contribution M12077*. Busan, Korea, Apr. 2005.

[22] A. Ishikawa, "KDDI test sequences for call for proposals on multi-view video coding," *MPEG Contribution M12402*. Poznan, Poland, Jul. 2005.

[23] Y. Ho, E. Lee, and C. Lee, "Multiview video test sequence and camera parameters," *MPEG Contribution M15419*. Archamps, France, Apr. 2008.

[24] M. Tanimoto, T. Fujii, M. P. Tehrani, M. Wildeboer, N. Fukushima, and H. Furihata, "Moving multiview camera test sequences for MPEG-FTV," *MPEG Contribution M16922*. Xi'an, China, Oct. 2009.

[25] S. B. Kang and L. Zitnick, "Projection Test and Results for Microsoft Research 3D Video (Breakdancing Frame)," [Online]. Available: http://research.microsoft.com/vision

[26] G. Bjontegaard, "Calculation of average PSNR differences between RD curves," in *ITU-T SG16/Q6, 13th VCEG Meeting, Doc. VCEG-M33*, Apr. 2001.

[27] P. Pandit and A. Vetro, "JMVM 2 software," *Doc. JVT-U208*. Hangzhou, China, Oct. 2006.

[28] Y. Su, A. Vetro, and A. Smolic, "Common Test Conditions for Multiview Video Coding," *Doc. JVT-T207 Klagenfurt*, Austria, Jul. 2006.

Pablo Carballeira received the Telecommunication Engineering degree (five years engineering program) from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 2007 and the Communications Technologies and Systems Master degree (two year M.S. program) from UPM in 2010. He is currently working towards the Ph.D. degree in Telecommunications at UPM.

Since 2007, he has been a member of the Grupo de Tratamiento de Imágenes (Image Processing Group), UPM. His research interests include image processing and video coding, focusing on multivew video coding (MVC) and 3D video coding.

Julián Cabrera received the Telecommunication Engineering degree and the Ph.D. degree in Telecommunications from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 1996 and 2003, respectively.

Since 1996, he has been a member of the Image Processing Group, UPM. He was a Ph.D. scholar of the Information Technology and Telecommunication Programs of the Spanish National Research Plan from 1996 until 2001. Since 2001, he has been a member of the faculty of the UPM, and since 2003 he has been an Associate Professor of Signal Theory and Communications. His professional interests include image and video coding, design and development of multimedia communications systems, focusing on Multivew Video Coding (MVC), 3D Video Coding and video transmission over variable rate channels. He has been actively involved in European projects (Acts, Telematics, IST) and national projects.

Antonio Ortega (S'91–M'95–SM'00–F'07) received the Telecommunications Engineering degree from the Universidad Politécnica de Madrid, Madrid, Spain, in 1989 and the Ph.D. degree in electrical engineering from Columbia University, New York, 1994, supported by a Fulbright scholarship.

In 1994, he joined the Electrical Engineering Department, University of Southern California (USC), where he is currently a Professor. He currently serves as an Associate Chair of EE-Systems and as was previously a Director of the Signal and Image Processing Institute at USC.

Prof. Ortega is a member of ACM and APSIPA. He has been Chair of the Image and Multidimensional Signal Processing (IMDSP) technical committee and a member of the Board of Governors of the IEEE Signal Processing Society (2002). He has been technical program Co-Chair of ICIP 2008, MMSP 1998 and ICME 2002. He has been Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE SIGNAL PROCESSING LETTERS, the *EURASIP Journal on Advances in Signal Processing*, and the *IEEE Signal Processing Magazine*. He is the inaugural Editor-in-Chief of the *APSIPA Transactions on Signal and Information Processing*, launched by APSIPA and Cambridge University Press in 2012. He received the NSF CAREER award, the 1997 IEEE Communications Society Leonard G. Abraham Prize Paper Award, the IEEE Signal Processing Society 1999 Magazine Award, the 2006 EURASIP Journal of Advances in Signal Processing Best Paper Award, and the ICIP 2011 best paper award.

His research interests are in the areas of multimedia compression, communications, and signal analysis. His recent work is focusing on distributed compression, multiview coding, error tolerant compression, wavelet-based signal analysis, information representation in wireless sensor networks and graph wavelets His work at USC has been or is being funded by agencies such as NSF, NASA, DOE, and companies such as HP, Samsung, Chevron, or Texas Instruments. Over 30 Ph.D. students have completed their Ph.D. theses under his supervision at USC and his work has led to over 300 publications in international conferences and journals, as well as several patents.

Fernando Jaureguizar received the Telecommunication Engineering degree (six-year engineering program) and the Ph.D. degree in telecommunications from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 1987 and 1994, respectively.

Since 1987, he has been a member of the Image Processing Group, UPM. In addition, since 1991 he has been a member of the faculty of the ETS Ingenieros de Telecomunicación at UPM, and since 1995 he has been an Associate Professor of Signal Theory and Communications, Department of Signals, Systems, and Communications. His professional interests include digital image processing, video coding, 3DTV, computer vision, and design and development of multimedia communications systems. He has been actively involved in European projects (Eureka, ACTS, and IST) and national projects in Spain.

Narciso García received the Ingeniero de Telecomunicación degree (five years engineering program) (Spanish National Graduation Award) and the Doctor Ingeniero de Telecomunicación degree (Ph.D. in communications) (Doctoral Graduation Award), both from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 1976 and 1983, respectively.

Since 1977, he has been a member of the faculty of the UPM where he is currently a Professor of signal theory and communications. He leads the Grupo de Tratamiento de Imágenes (Image Processing Group) of the UPM. He has been actively involved in Spanish and European research projects, serving also as evaluator, reviewer, auditor, and observer of several research and development programs of the European Union. He was a cowriter of the EBU proposal, base of the ITU standard for digital transmission of TV at 34–45 Mb/s (ITU-T J.81). His professional and research interests are in the areas of digital image and video compression and of computer vision.